

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

FERRAMENTA DE AUXÍLIO A TRATAMENTOS
FISIOTERAPÊUTICOS COM O KINECT

FELIPE PILON

BLUMENAU
2013

2013/1-15

FELIPE PILON

FERRAMENTA DE AUXÍLIO A TRATAMENTOS

FISIOTERAPÊUTICOS COM O KINECT

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

Prof. Aurélio Hoppe, Mestre - Orientador

**BLUMENAU
2013**

2013/1-15

FERRAMENTA DE AUXÍLIO A TRATAMENTOS FISIOTERAPÊUTICOS COM O KINECT

Por

FELIPE PILON

Trabalho aprovado para obtenção dos créditos na disciplina de Trabalho de Conclusão de Curso II, pela banca examinadora formada por:

Presidente: _____
Prof. Aurélio Hoppe, Mestre – Orientador, FURB

Membro: _____
Prof. Dalton Solano dos Reis, Mestre – FURB

Membro: _____
Prof. Francisco Adell Péricas, Mestre – FURB

Blumenau, 8 de julho de 2013

Dedico este trabalho a todos os professores, familiares e amigos que, de maneira direta ou indireta, me ajudaram na elaboração do mesmo.

AGRADECIMENTOS

À Deus, pela capacidade recebida.

À minha família, que com esforço e sacrifício, me deram as condições necessárias para realizar mais este trabalho.

Aos meus amigos, pela ajuda recebida em todos os momentos que precisei.

Ao meu orientador, Aurélio Hoppe, que com paciência, competência e dedicação me incentivou e mostrou o caminho para a completa realização deste trabalho.

À minha empresa, BGA Gerencial Tecnologia, pelo incentivo e pelo tempo concedido durante o desenvolvimento deste.

Às alunas do Curso de Fisioterapia, Daniela Cristina Sabel e Giovana Vettorello, que compartilharam parte do conhecimento necessário para o correto desenvolvimento deste.

Tudo pode ser tirado de um homem, menos a última de suas liberdades – escolher de que maneira vai agir diante das circunstâncias do seu destino.

Viktor Emil Frankl

RESUMO

Este trabalho apresenta o desenvolvimento de uma ferramenta de auxílio a tratamentos fisioterápicos que faz o uso do sensor Kinect. A ferramenta permite ao fisioterapeuta criar exercícios utilizando peças posicionadas dinamicamente no espaço 3D de acordo com o paciente e o tipo de tratamento a ser feito. Ao executar um exercício cadastrado, o paciente visualiza na tela as peças do exercício e o esqueleto que representa o seu próprio corpo, reproduzindo em tempo real os seus movimentos. Os testes que foram realizados com pacientes reais comprovaram a eficácia no uso da ferramenta nos tratamentos de reabilitação dos mesmos. Para desenvolvimento da ferramenta utilizou-se o ambiente de desenvolvimento Visual Studio 2012, com a linguagem de programação C#, banco de dados SQL Server 2012 e utilizando as tecnologias *Microsoft Entity Framework* e *Windows Presentation Foundation*.

Palavras-chave: Kinect. Fisioterapia. Reabilitação.

ABSTRACT

This paper presents the development of a tool to aid the physiotherapy treatment that makes use of the Kinect sensor. The tool allows the therapist to create exercises using parts dynamically positioned in 3D space according to the patient and the type of treatment to be done. Registered to perform an exercise, the patient displayed on screen parts of the exercise and the skeleton that represents your own body, playing in real time their movements. The tests that were conducted with real patients demonstrated the effective use of the tool in rehabilitation treatments thereof. For tool development used the development environment Visual Studio 2012 with the C# programming language, database SQL Server 2012 and the technologies Microsoft Entity Framework and Windows Presentation Foundation.

Key-words: Kinect. Physiotherapy. Rehabilitation.

LISTA DE ILUSTRAÇÕES

Figura 1 - O sensor Kinect.....	15
Figura 2 - Juntas reconhecidas pelo Kinect	17
Quadro 1- Diferença(s) entre o método tradicional de fisioterapia e o método com o uso do Wii	21
Figura 3 - Tela do jogo Pong com duas raquetes habilitadas	23
Figura 4 - Movimento executado corretamente.....	25
Figura 5 - Diagrama de casos de uso da ferramenta.....	28
Figura 6 - Diagrama de Classes	29
Figura 7 - Diagrama de sequência da ferramenta	31
Figura 8 - Diagrama da base de dados.....	32
Quadro 2 - Gramática para reconhecimento dos comandos de voz	34
Quadro 3 - Implementação da interface IObjetoGrafico2D na classe Junta	35
Quadro 4 - Rotina que conecta ao sensor Kinect.....	36
Quadro 5 - Rotina para atualizar os esqueletos (parte 1).....	36
Quadro 6 - Rotina para atualizar os esqueletos (parte 2).....	37
Quadro 7 - Procedimento executado a cada novo frame captado pelo sensor Kinect.....	37
Quadro 8 - Rotina para solicitar o início da calibragem ao gerenciador de exercícios	38
Quadro 9 - Procedimento executado quando um comando de voz é reconhecido	39
Quadro 10 - Rotina para ordenar a lista de objetos gráficos	40
Quadro 11 - Procedimento para verificar a calibragem do esqueleto.....	41
Quadro 12 - Procedimento para iniciar o posicionamento das peças	41
Quadro 13 - Procedimento que inicia a execução de um exercício.....	42
Quadro 14 - Procedimento que carrega um exercício do banco de dados.....	43
Quadro 15 - Procedimento que retorna as peças a serem desenhadas	43
Quadro 16 - Executando a calibragem do esqueleto	45
Quadro 17 - Definindo as configurações do exercício	47
Quadro 18 - Posicionando as peças no espaço 3D	49
Quadro 19 - Carregando e executando um exercício	51
Quadro 20 - Data dos testes.....	54
Quadro 21 - Respostas dos pacientes ao questionário.....	55
Quadro 22 - Comparação com os trabalhos correlatos.....	56

Quadro 23 – Caso de uso calibrar esqueleto.....	62
Quadro 24 - Caso de uso cadastrar exercício	63
Quadro 25 - Caso de uso carregar exercício.....	64
Quadro 26 - Caso de uso executar exercício	64
Quadro 27 -Termo de consentimento livre e esclarecido (parte 1)	65
Quadro 28 - Termo de consentimento livre e esclarecido (parte 2)	66
Quadro 29 - Mini exame do estado mental (parte 1).....	67
Quadro 30 - Mini exame do estado mental (parte 2).....	68
Quadro 31 - Mini exame do estado mental (parte 3).....	69
Quadro 32 - Questionário (parte 1).....	70
Quadro 32 - Questionário (parte 2).....	71

LISTA DE SIGLAS

CIF – Classificação Internacional de Funcionalidade, Incapacidade e Saúde

CMOS – *Complimentary Metal-Oxide Semiconductor*

E3 – *Electronic Entertainment Expo*

GUI – *Graphic User Interfaces*

ICF – *International Classification of Functioning, Disability and Health*

MEEM – Mini Exame do Estado Mental

NUI – *Natural User Interfaces*

OMS – Organização Mundial da Saúde

RGB – *Red, Green and Blue*

RV – Realidade Virtual

SGBD – Sistema Gerenciador do Banco de Dados

TCLE – Termo de Consentimento Livre e Esclarecido

UML – *Unified Modeling Language*

WPF – *Windows Presentation Foundation*

XAML – *eXtensible Application Markup Language*

SUMÁRIO

1 INTRODUÇÃO.....	13
1.1 OBJETIVOS DO TRABALHO	14
1.2 ESTRUTURA DO TRABALHO	14
2 FUNDAMENTAÇÃO TEÓRICA	15
2.1 O KINECT.....	15
2.2 REABILITAÇÃO FISIOTERAPÊUTICA.....	17
2.2.1 Incapacidade motora	18
2.2.2 Reabilitação neurológica.....	18
2.3 TRABALHOS CORRELATOS	20
2.3.1 O impacto da tecnologia Nintendo Wii no tratamento fisioterapêutico e na satisfação de pacientes em uma clínica no Vale do Taquari	20
2.3.2 Desenvolvimento de um jogo eletrônico para reabilitação utilizando um sensor de som e movimento (Kinect)	21
2.3.3 Improving Motor Rehabilitation Process through a Natural Interaction Based System Using Kinect Sensor.....	24
3 DESENVOLVIMENTO DA FERRAMENTA	27
3.1 REQUISITOS DA FERRAMENTA A SER DESENVOLVIDA.....	27
3.2 ESPECIFICAÇÃO DA FERRAMENTA	28
3.2.1 Diagrama de casos de uso	28
3.2.2 Diagrama de classes	28
3.2.3 Diagrama de sequência	30
3.2.4 Base de dados	32
3.3 IMPLEMENTAÇÃO DA FERRAMENTA.....	32
3.3.1 Ferramentas utilizadas.....	33
3.3.2 Implementação da ferramenta	34
3.3.3 Operacionalidade da ferramenta	43
3.3.3.1 Calibrando o esqueleto	44
3.3.3.2 Criando um novo exercício.....	46
3.3.3.3 Executando um exercício.....	49
3.4 RESULTADOS E DISCUSSÃO	51
3.4.1 Avaliação da ferramenta	51

3.4.2	Versão inicial reprovada	52
3.4.3	Versão final	53
3.4.4	Amostragem e instrumentos de coleta de dados	53
3.4.5	Procedimentos para coleta de dados	54
3.4.6	Análise qualitativa dos resultados	54
3.4.7	Comparativo com os trabalhos correlatos	56
4	CONCLUSÕES	57
4.1	EXTENSÕES	58
	REFERÊNCIAS BIBLIOGRÁFICAS	60
	APÊNDICE A – Detalhamento dos casos de uso	62
	APÊNDICE B – Termos de consentimento livre e esclarecido	65
	APÊNDICE C – Mini exame do estado mental	67
	APÊNDICE D – Questionário	70

1 INTRODUÇÃO

O avanço tecnológico e o crescimento da indústria de computadores fez com que a tecnologia de Realidade Virtual (RV), que existe há mais de duas décadas, deixasse de ser viável apenas para as grandes empresas e instituições de pesquisa. Atualmente é fácil encontrarmos software e hardware de baixo custo baseados nessa tecnologia. Hoje pode-se afirmar que a tecnologia de RV oferece uma opção financeiramente acessível para a solução de diversos problemas (VALERIO NETTO; MACHADO; OLIVEIRA, 2002, p. 4).

Segundo Cabreira e Müling (2012, p. 2), com o incremento dos componentes gráficos nos anos 80 surgiram as primeiras *Graphic User Interfaces* (GUI), que são interfaces gráficas que remontavam metáforas do cotidiano do usuário, criando uma interação mais amigável. A partir daí, as interfaces gráficas se desenvolveram e tornaram-se a principal ponte de interação entre o humano e o computador.

Cabreira e Müling (2012, p. 2) ainda citam que os avanços tecnológicos, hoje, possibilitam criar novas formas de interação. Essas formas modificam drasticamente a forma de pensar das interfaces tradicionais e abrem espaço para discutir novos paradigmas, como o de *Natural User Interfaces* (NUI). Estas podem ser definidas como interfaces que não apresentam uma metáfora gráfica controlada mecanicamente, e sim, criam experiências nas quais o usuário tem a sensação de estar interligado com o sistema em questão com o uso de gestos ou linguagem sonora. Os autores dizem que NUI não é um conceito novo, mas que com o lançamento de sistemas digitais, como o Kinect, abrem-se caminhos para interfaces digitais cada vez de formas mais naturais.

Embora o Kinect tenha sido criado inicialmente para atuar em jogos no console Xbox 360 da Microsoft, percebeu-se a possibilidade de aplicar a tecnologia presente no sensor em outras situações. A Microsoft então lançou recentemente a versão do Kinect para Windows, incentivando o desenvolvimento de aplicações além de jogos tradicionais (CABREIRA; MÜLLING, 2012, p. 2).

Rocha, Defavari e Brandão (2012, p. 17) afirmam que “apesar de ser um campo de estudos recente, o uso do Kinect como ferramenta de auxílio no atendimento de pacientes, não só neurológicos, mas de diferentes áreas, representa um imenso leque de possibilidades”.

Bruckheimer (2011) afirma que o uso do conceito de Reabilitação Virtual através de tecnologias como o Kinect em tratamentos fisioterapêuticos introduz a possibilidade de motivar os pacientes, pois permite que seja estabelecido um contexto para a realização das atividades que extrapolam o tratamento. Bruckheimer diz que na fisioterapia tradicional o

paciente deve realizar uma sequência de atividades repetitivas e que, através de recursos computacionais, é possível utilizar um jogo, por exemplo, onde o paciente conseguiria realizar estes mesmos movimentos.

Como pôde ser visto, tratamentos fisioterápicos feitos no modo tradicional podem ser, muitas vezes, desgastantes para os pacientes. Também percebe-se que pesquisas relacionadas ao uso do Kinect na fisioterapia ainda são raras, pois se trata de uma tecnologia que foi recentemente disponibilizada para seu uso além dos jogos de entretenimento. Diante disto, desenvolveu uma ferramenta de auxílio a tratamentos fisioterapêuticos que utilize o Kinect para servir de motivação aos pacientes na execução dos exercícios durante tais tratamentos. Ao fim, espera-se ser possível fazer uma avaliação quanto ao uso do Kinect, observando sua aplicabilidade neste tipo de tratamento.

1.1 OBJETIVOS DO TRABALHO

Este trabalho tem, como primeiro objetivo, desenvolver uma ferramenta de auxílio a tratamentos de reabilitação fisioterapêutica utilizando o sensor Kinect, de forma que contribua como motivação aos pacientes na execução dos exercícios propostos pelos fisioterapeutas. A ferramenta deverá permitir aos fisioterapeutas cadastrarem exercícios de acordo com a necessidade de cada tratamento e paciente.

Como segundo objetivo, deseja-se avaliar se o uso da ferramenta juntamente com o sensor Kinect é viável nos tratamentos de reabilitação fisioterapêutica. Deverão ser avaliadas características como: satisfação do paciente, precisão da aplicação, facilidade no uso e o nível de contribuição ao tratamento.

1.2 ESTRUTURA DO TRABALHO

Este trabalho está dividido em quatro capítulos. O primeiro capítulo apresenta a motivação e os objetivos a serem atingidos com este trabalho. No segundo capítulo está descrita a fundamentação teórica utilizada para o desenvolvimento da ferramenta proposta, seguida de três trabalhos correlatos. Já no terceiro capítulo estão expostas as informações sobre o desenvolvimento da ferramenta desenvolvida e os dados obtidos nos testes com os pacientes. Por fim, no quarto capítulo são apresentadas as conclusões do trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

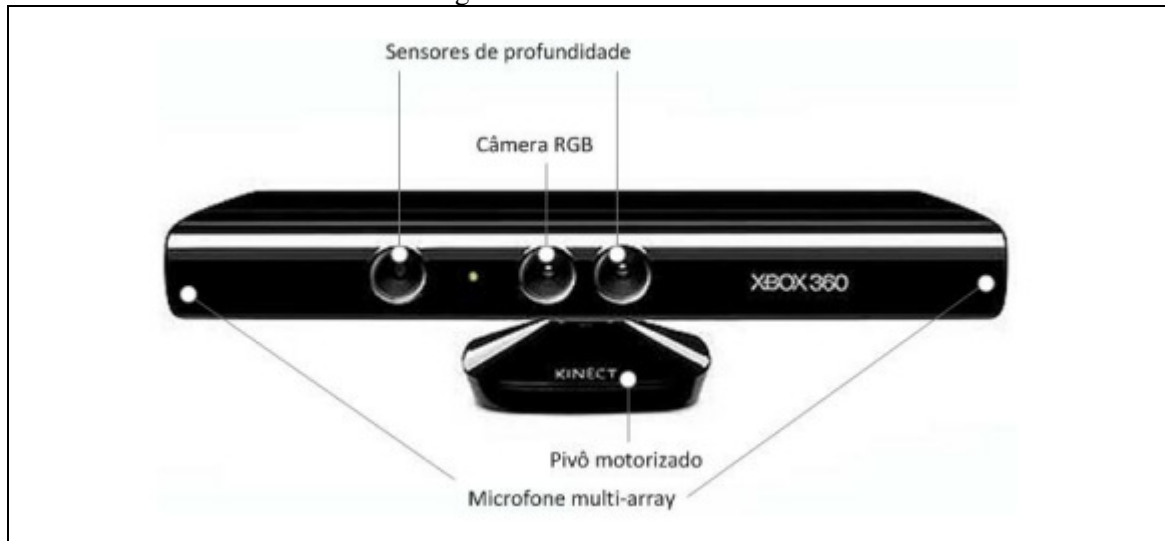
Na seção 2.1 deste capítulo serão demonstradas informações sobre o sensor Kinect. Na seção 2.2 será feita uma descrição de como são realizados os tratamentos de reabilitação fisioterapêutica tradicionais. Por fim, na seção 2.3, serão listados três trabalhos correlatos referentes ao uso de recursos computacionais em tratamentos de reabilitação.

2.1 O KINECT

O Kinect é um dispositivo capaz de capturar movimentos de objetos expostos a ele. Foi desenvolvido pela Microsoft e teve sua divulgação oficial anunciada em 2009 na *Electronic Entertainment Expo* (E3). Na época, o projeto, de codinome Projeto Natal, foi apresentado como uma tecnologia para jogos onde o usuário controlaria os objetos do jogo sem o uso dos tradicionais *joysticks*, mas sim, utilizaria movimentos corpóreos (LOWENSOHN, 2011).

O dispositivo é baseado num par de câmeras capazes de capturar imagens em 3D. O uso desta tecnologia focando a visão e interação natural foi proposta inicialmente por uma equipe de pesquisadores israelenses (possessores da patente do projeto), cujo início das pesquisas aponta para o ano de 2005 (ZALEVSKY, 2007).

Figura 1 - O sensor Kinect



Fonte: Sá (2011, p. 18).

O Kinect é um dispositivo que possui uma avançada combinação de hardware e software. A seguir é feita uma breve descrição sobre a funcionalidade da camada de hardware do mesmo, que pode ser observada na Figura 1. Segundo Sá (2011, p. 19), os principais componentes de hardware encontrados no Kinect são:

- a) câmera *Red, Green e Blue* (RGB): utilizada para reconhecimento facial e de outras características possíveis, através do reconhecimento das cores vermelho, verde e

azul. A câmera trabalha numa resolução de 640 x 480 pixels e a uma taxa de 30 *Frames Por Segundo* (FPS);

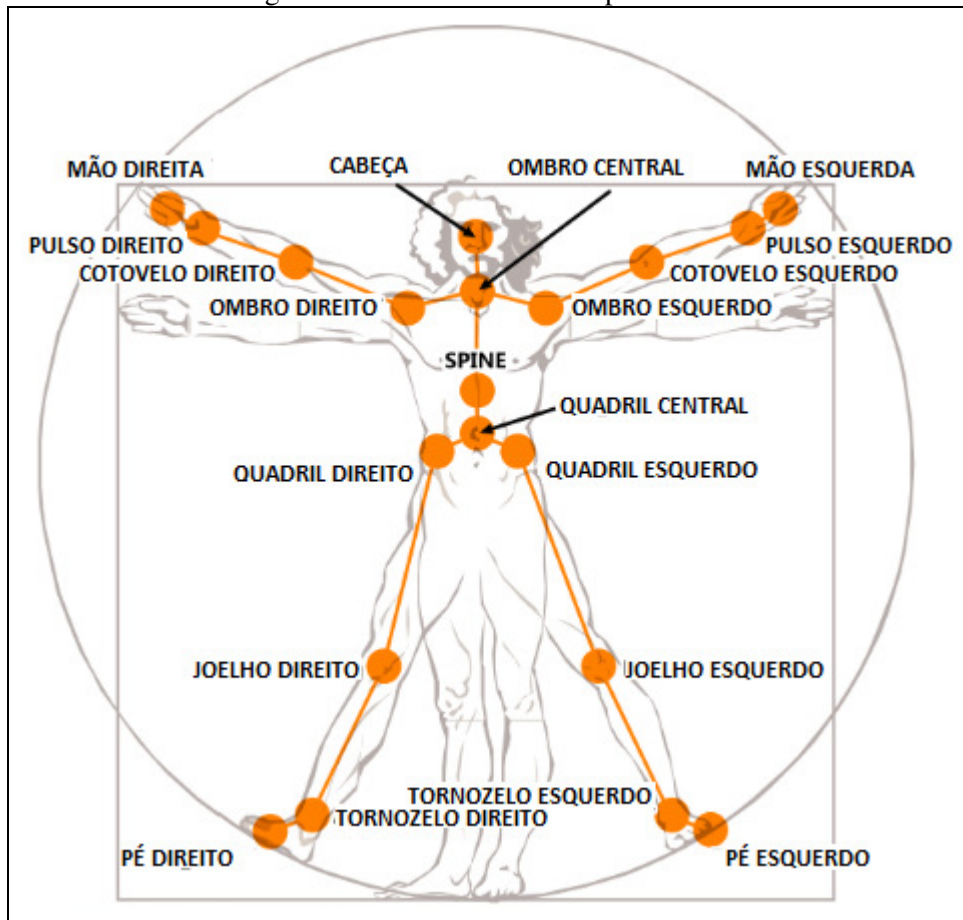
- b) sensores de profundidade: é a combinação do uso de um projetor de luz infravermelha e de um sensor *Complimentary Metal-Oxide Semiconductor* (CMOS), conseguindo assim, identificar a profundidade dos objetos do ambiente independente dos níveis de iluminação;
- c) conjunto de 4 microfones: a combinação do som captado pelos 4 microfones permite isolar as vozes do usuário de outros ruídos do ambiente, permitindo o uso dos comandos por voz de uma maior distância.

Apesar da excelente combinação de hardware existente no Kinect, o que mais se destaca é a camada de software. Em entrevista à revista *Scientific American* o diretor responsável pela incubação do projeto na Microsoft, Alex Kipman, afirmou que “cada simples movimento do corpo é uma entrada, assim você precisaria programar infinitas reações para tais ações, logo, não seria possível criar um programa que tenha todas essas ‘ações e reações’ previamente cadastradas” (KUCHINSKAS, 2010).

Seguindo este raciocínio, os desenvolvedores da Microsoft perceberam que seria necessário que o programa aprendesse como reagir classificando os gestos das pessoas no mundo real. Para tanto, foi feita a leitura de uma quantidade massiva de dados capturando movimentos em cenários da vida real. Com isso tornou-se possível mapear os dados de pessoas de diferentes idades, tipo de corpos, vestuário, etc., resumidamente, tornou-se possível saber a pose atual em que ela se encontra (KUCHINSKAS, 2010).

Com o Kinect podendo identificar o “esqueleto” de um corpo, viabilizou-se fazer com que o sistema aprenda a destacar as juntas e definir as distâncias entre elas. Sá (2011, p. 19) destaca que “O esqueleto fornecido pelo sensor do Kinect contém um conjunto de até vinte diferentes articulações identificadas pelo sensor, cada uma dessas articulações representa uma articulação correspondente no corpo humano”, como pode ser observado na Figura 2. Assim, sabendo quais juntas do corpo são adjacentes entre elas, é possível dizer qual a posição atual do membro correspondente, mesmo que este último esteja encoberto por outro membro ou pelo esqueleto de outros usuários.

Figura 2 - Juntas reconhecidas pelo Kinect



Fonte: Sá (2011, p. 18).

Um ponto importante a ser observado é o fato de que o sensor Kinect devolve não somente o esqueleto das pessoas, isso é fornecido de modo padrão pela biblioteca Kinect SDK pelo fato de que o seu foco inicial é o reconhecimento de pessoas visando à interação através de gestos e movimentos. O Kinect tem poder para fornecer toda uma malha de pontos do ambiente à sua frente, assim é possível fazer ferramentas para reconhecer qualquer objeto nesse ambiente (YAMADA et al., 2012).

Para quem deseja desenvolver aplicações com o uso do Kinect, a Microsoft, disponibiliza para download, desde 2011, a biblioteca Kinect SDK. A partir dela desenvolvedores do mundo todo podem usar os recursos oferecidos pelo Kinect no desenvolvimento de aplicações específicas para uma ou mais áreas do conhecimento (CABREIRA; MÜLLING, 2012, p. 2).

2.2 REABILITAÇÃO FISIOTERAPÊUTICA

Nesta seção, serão levantados alguns pontos importantes em relação a tratamentos de reabilitação fisioterapêutica. Inicialmente será exposta uma breve explicação do termo

incapacidade motora, e como é feita sua avaliação. A seguir será descrito como são realizados tratamentos de reabilitação usando mecanismos da neuroplasticidade.

2.2.1 Incapacidade motora

Em maio de 2001, a Organização Mundial da Saúde (OMS), aprovou *International Classification of Functioning, Disability and Health* (ICF). Que foi traduzido para o idioma português pelo Centro Colaborador da Organização Mundial da Saúde para a Família de Classificações Internacionais em Língua Portuguesa com o título de Classificação Internacional de Funcionalidade, Incapacidade e Saúde (CIF) (FARIAS; BUCHALLA, 2005, p. 189).

Segundo Farias e Buchalla (2005, p. 189), “a CIF descreve a funcionalidade e a incapacidade relacionadas às condições de saúde, identificando o que uma pessoa ‘pode ou não fazer na sua vida diária’”. Ainda segundo o autor, a CIF considera as funções dos órgãos ou sistemas e estruturas do corpo, assim como as limitações de atividades e da participação social no meio ambiente onde a pessoa vive.

Segundo o modelo proposto na CIF, o termo incapacidade resulta da interação entre a disfunção apresentada pelo indivíduo, limitação de suas atividades e a restrição na participação social, com os fatores ambientais que podem vir a atuar como facilitadores ou barreiras para o desempenho dessas atividades e participação nas mesmas (FARIAS; BUCHALLA, 2005, p. 189).

2.2.2 Reabilitação neurológica

Segundo Ribeiro (2005, p. 2), “as interações entre os estímulos ambientais e as respostas de um organismo determinam as propriedades comportamentais que lhe garantem adaptação a diferentes situações e individualidade comportamental”. O autor também descreve que o sistema nervoso central, possui uma rede neural com células que determinam a sensibilidades e as ações motoras, traduzindo-as em comportamento. Quando existem lesões, surge um desarranjo nesta rede e o processo nervoso reinicia seus processos de reorganização e regeneração.

Para Rocha, Defavari e Brandão (2012, p. 17), compreendendo que as patologias neurológicas podem provocar deficiência, incapacidades e desvantagens, tem-se que a reabilitação deve buscar impedir, reverter/corrigir, minimizar ou, pelo menos, proporcionar um mínimo de qualidade de vida durante o período de acometimento destes problemas. Os

autores definem que “reabilitar significa adquirir novamente uma habilidade tenha ela sido perdida ou reduzida”.

Fernandes e Santos (2012, p.1) definem que a reaprendizagem motora é definida como sendo uma mudança relativamente permanente no comportamento como resultado da prática. A aprendizagem motora é assumida em todos os níveis do sistema nervoso central, sendo que a necessidade ou desejo de interagir com o ambiente de forma adequada é um fator importante nas alterações que ocorrem a esse nível. As autoras destacam também que os movimentos repetitivos, o *feedback* e a motivação do paciente também são princípios básicos na reaprendizagem motora.

Quanto aos movimentos repetitivos, segunda as autoras Fernandes e Santos (2012, p. 5), estes têm grandes benefícios para o resultado funcional, pois elas descrevem que certos efeitos de estimulação cerebral só ocorrem se a área em questão for estimulada repetitivamente. Porém, as autoras alertam que, estas repetições devem ser reguladas a fim de assegurar que fatores como o cansaço ou dor sejam minimizados.

Fernandes e Santos (2012, p. 6) afirmam que até meados da década de 90 pensou-se que não havia possibilidade de reparação ou alteração do sistema nervoso central após uma lesão no mesmo, porém terapeutas descobriram que muitos pacientes melhoraram e aprenderam a mover-se novamente. Isso marcou a descoberta do uso da neuroplasticidade para reaprendizagem motora.

Ribeiro (2005, p. 2) descreve que “o principal propósito da neuroplasticidade é a capacidade que o sistema nervoso central possui em modificar algumas das suas propriedades morfológicas e funcionais em relação às alterações ambientais”. O autor complementa que a neuroplasticidade permite o desenvolvimento de alterações estruturais em resposta à experiência e como adaptação a condições mutantes e estímulos repetitivos.

Segundo Fernandes e Santos (2012, p. 8), a ativação cortical é alterada na aprendizagem de uma sequência repetida, introduzindo um padrão mais normal. Assim, uma tarefa de aprendizagem motora pode tornar-se um estímulo para a mudança da neuroplasticidade e pode corrigir padrões de atividades cerebrais e sustentar a função.

Enfim, Umphred (2004) descreve que a prática repetitiva é utilizada em uma tentativa de reestabelecer a capacidade do paciente em desempenhar atividades da vida diária, ou seja, de reaprender a fazer e corrigir incapacidades e limitações funcionais.

2.3 TRABALHOS CORRELATOS

Nesta seção serão apresentados três trabalhos correlatos. O primeiro trabalho é um estudo realizado para verificar os impactos no tratamento fisioterapêutico e na satisfação dos pacientes com a utilização do console Nintendo Wii (BRESCIANI; CONTO, 2012). O segundo trata-se de um trabalho de conclusão de curso onde foi criado um jogo eletrônico para reabilitação fisioterapêutica utilizando o dispositivo Kinect (FERRAZ; YAMASHITA, 2012). Já o terceiro trabalho descreve uma ferramenta que permita interação corporal objetivando melhorar a execução de exercícios pelos pacientes em tratamentos de reabilitação (FREITAS et al., 2012).

2.3.1 O impacto da tecnologia Nintendo Wii no tratamento fisioterapêutico e na satisfação de pacientes em uma clínica no Vale do Taquari

O trabalho realizado por Bresciani e Conto (2012, p. 81) consiste em uma pesquisa qualitativa e exploratória feita em uma clínica na cidade de Lajeado, no Estado do Rio Grande do Sul, onde foi analisada a satisfação dos pacientes que fizeram uso do videogame Nintendo Wii durante os tratamentos fisioterapêuticos.

Segundo Bresciani e Conto (2012, p. 82), “o objetivo principal foi verificar os impactos no tratamento fisioterapêutico e na satisfação dos pacientes com a utilização do software Nintendo Wii para treinamento do controle motor e aumento de estímulo.” Para realização da pesquisa foi utilizada uma amostragem não probabilística, de seis pacientes, tendo três deles de 19 a 25 anos, um que possui de 26 a 30 anos, um na faixa de 31 a 40 anos e outro com mais de 50 anos. Eles foram submetidos a um questionário estruturado, entrevistas individuais e observação local.

Para elaboração do questionário foi usada a escalado tipo Likert, onde os itens são avaliados um com intensidade maior que o outro, podendo o paciente alternar entre cinco opções. Segundo os autores, “essa escala é ideal para o questionário qualitativo de avaliação da satisfação, pois como há um número reduzido de respondentes, é possível comparar o nível de satisfação entre eles com maior relevância” (BRESCIANI; CONTO, 2012, p. 82).

Os pacientes foram submetidos a sessões semanais de fisioterapia que duravam entre 45 e 60 minutos, nas quais ficavam em contato com o Nintendo Wii por cerca de 70% do tempo. Antes disso, apenas dois dos pacientes haviam tido contato ou tinham ouvido falar do Wii, porém, todos já haviam feito tratamentos fisioterapêuticos da maneira tradicional, isto,

ao ponto de vista os autores, facilitou a comparação entre o método de reabilitação tradicional e o método que utiliza o Wii (BRESCIANI; CONTO, 2012, p. 81).

Logo após os pacientes terem o primeiro contato com o Wii, os mesmos foram questionados sobre a diferença sentida entre os dois métodos citados anteriormente, as respostas podem ser vistas no Quadro 1.

Quadro 1- Diferença(s) entre o método tradicional de fisioterapia e o método com o uso do Wii

“Mais disposição para fazer os exercícios e parece que há uma recuperação mais rápida”
“Benefícios: correção da postura e do equilíbrio, melhora dos movimentos, recuperação mais rápida, além da motivação”
“Mais divertido, não se vê o tempo passar”
“É melhor, reabilitação melhor”
“Tenho uma resposta melhor do que estou fazendo, como consequência, melhorando meu desempenho”
“Com o uso do videogame há maior interação e acompanhamento da recuperação”

Fonte: Bresciani e Conto (2012, p. 88).

Neste primeiro questionário os autores afirmam que os pacientes sentiram-se motivados com a técnica, sendo que todos classificaram como ótima a experiência obtida. Com base em outros questionamentos feitos, foi possível identificar que um dos pontos mais motivadores é o fato de poder acompanhar o exercício em tempo real na TV, como foi destacado por um deles quando disse que “ajuda em um melhor *feedback* visual, facilitando”. Outros benefícios também foram relatados, dentre eles a falta de percepção da dor e do tempo gasto, também somaram como resultados positivos (BRESCIANI; CONTO, 2012, p. 90).

Apesar dos resultados positivos, os autores alertam que o videogame só se torna uma ferramenta no tratamento fisioterapêutico quando realizado sob a supervisão de um profissional fisioterapeuta. Seu uso sem este último torna o videogame somente uma ferramenta de lazer (BRESCIANI; CONTO, 2012, p. 92).

Tendo a pesquisa finalizada, Bresciani e Conto (2012, p. 14) chegaram à conclusão de que “[...] fica evidente a satisfação do paciente ao executar as atividades. [...] ele está realizando o mesmo tipo de exercício que é aplicado no método tradicional de fisioterapia, mas de forma distraída, não percebendo o cansaço e concluindo toda a atividade”.

2.3.2 Desenvolvimento de um jogo eletrônico para reabilitação utilizando um sensor de som e movimento (Kinect)

O objetivo do trabalho é projetar e implementar um jogo utilizando o sensor de som e movimento Kinect para ser utilizado em reabilitação (FERRAZ; YAMASHITA, 2012). A motivação que levou aos autores a realizar o trabalho foi o fato de muitas pessoas que passam

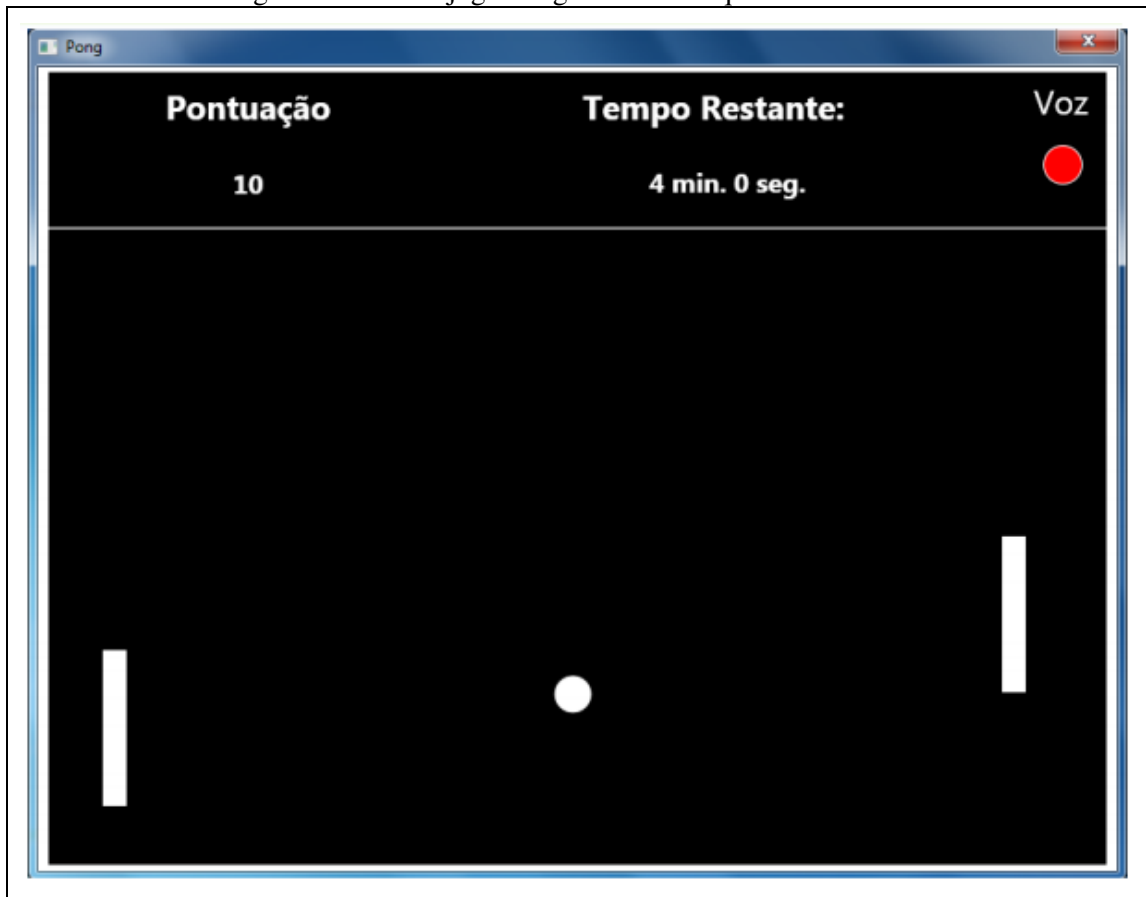
por algum tipo de reabilitação precisarem fazer diversas sessões de fisioterapia e isto pode se tornar um fator de desmotivação ao paciente. E a criação de um software para o Kinect pode, além de aumentar a motivação do paciente e a eficácia da reabilitação, facilitar o trabalho tanto do paciente como do fisioterapeuta.

Visando tornar o uso da ferramenta mais confortável, motivadora e eficiente, foi proposto um jogo simples e de fácil compreensão: o Pong. A ideia foi fazer com que o paciente use até duas juntas do corpo, reconhecidas pelo Kinect, para controlar ambas as raquetes. O objetivo do jogo é não permitir que a bola chegue às laterais da tela. O jogo permite que parâmetros como intervalo de tempo e velocidade da bola sejam ajustados (FERRAZ; YAMASHITA, 2012).

Para implementar o projeto, Ferraz e Yamashida (2012), utilizaram o programa Visual Studio 2010 juntamente com o Kinect *for* Windows SDK 1.5.

O jogo também permite que o usuário jogue apenas com uma das mãos, ou seja, no jogo ele controlará apenas uma raquete. Existe um sistema de pontuação que é mostrado na tela, a pontuação é determinada pelos acertos no centro da raquete vezes 30, mais os acertos na outra parte da raquete vezes 10, menos as bolas perdidas vezes 5. Na tela também é mostrado o tempo restante, quando este acabar o jogo para. Na Figura 3 pode-se observar o jogo em execução.

Figura 3 - Tela do jogo Pong com duas raquetes habilitadas



Fonte: Ferraz e Yamashida (2012).

O jogo Pong desenvolvido por Ferraz e Yamashida (2012) permite salvar os resultados de uma partida em arquivos texto ao final de um jogo. Outra opção disponível é habilitar o controle da ferramenta por comandos de voz para controlar um jogo em execução, por exemplo, os comandos de voz “*Stop*” e “*New*”, que servem para parar e começar um novo jogo, respectivamente.

Ao final do projeto, Ferraz e Yamashida (2012) concluem que foi possível desenvolver um software para reabilitação utilizando o Kinect. A ferramenta desenvolvida pode ser instalada facilmente podendo ser usada na própria casa do paciente já que ele não necessitará nada mais que um PC com o sistema operacional Windows 7 ou superior, do programa Kinect for Windows SDK v1.5 e do próprio sensor Kinect. Os comandos de voz também ajudaram permitindo controlar a ferramenta sem precisar sair da posição. Além disso, o software conseguiu, de maneira eficaz, captar o movimento de até duas juntas e associá-las ao movimento das raquetes. Por fim, a ferramenta demonstrou níveis de dificuldade adequados a uma sessão de reabilitação, com os parâmetros ajustáveis permitindo que o software se adapte a cada paciente.

Quanto a possíveis melhorias, Ferraz e Yamashida (2012) propõe criar um jogo para dois jogadores, incluir novas juntas nas opções, gravar a execução de um jogo e futuramente, caso o Kinect ganhe um pacote de idiomas em português, seja feita a tradução dos comandos de voz.

2.3.3 *Improving Motor Rehabilitation Process through a Natural Interaction Based System Using Kinect Sensor*

Este trabalho foi desenvolvido por uma equipe de pesquisadores da Universidade Federal de Pernambuco (FREITAS et al., 2012). O estudo feito propõe o desenvolvimento de uma interface que permita interação corporal objetivando melhorar a execução de exercícios pelos pacientes em tratamentos de reabilitação, seja ela ortopédica ou cognitiva.

O protótipo proposto supervisiona a execução dos exercícios, provendo orientações e estímulos. Ele foca na orientação e correção dos movimentos em execução durante a terapia de reabilitação motora. Isso permitiria que uma pessoa fizesse o tratamento sozinho, onde não teria o contato direto com o fisioterapeuta (FREITAS et al., 2012).

Através do esqueleto provido pelo Kinect é possível executar uma análise efetuando o reconhecimento e a correção dos movimentos. Para que o sistema consiga avaliar os exercícios que o paciente fará, é preciso reconhecer primeiramente um movimento e se o mesmo está sendo feito de maneira correta. Como se trata de um protótipo, o foco inicial é usar o sistema na avaliação de movimentos efetuados a partir do ombro, onde o paciente movimenta o braço afastando-o do resto do corpo, criando um ângulo entre o tronco e o braço no plano frontal (FREITAS et al., 2012).

Para calcular o ângulo atual do movimento do braço são utilizados os dois pares de pontos cotovelo-ombro e quadril-ombro. Assim, aplicando o cálculo do cosseno obtêm-se o ângulo. Da mesma maneira faz-se o cálculo para encontrar o ângulo existente entre o braço e o antebraço do paciente. Para esse, são utilizados os dois pares de pontos cotovelo-ombro e cotovelo-mão (FREITAS et al., 2012).

Também foi preciso garantir que o movimento está sendo executado dentro do plano estabelecido. Para isso também é verificado o ângulo entre os pontos do braço e do tórax do paciente, assim garante-se a correta análise do exercício independente da orientação do corpo do usuário em relação ao Kinect (FREITAS et al., 2012).

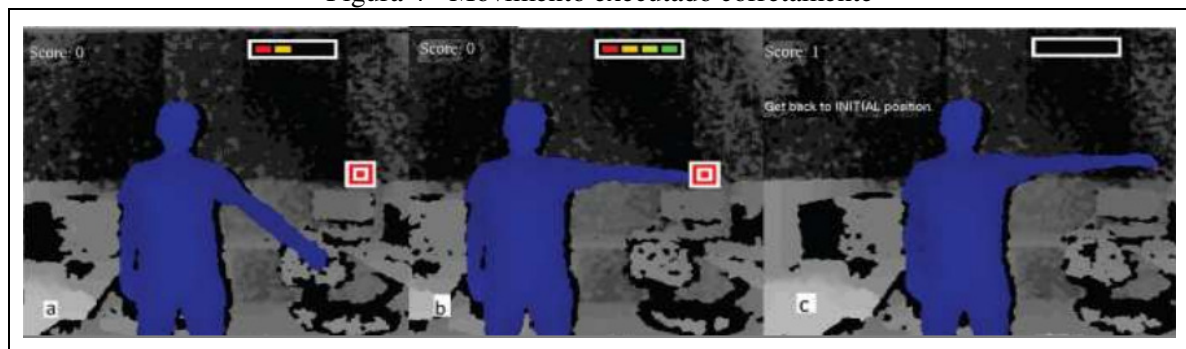
Captado todos os ângulos é possível realizar a avaliação dos movimentos. Estes, se realizados de maneira correta, somam pontos positivos para o paciente. Segundo Freitas et al.

(2012, p. 1), neste trabalho, um movimento é considerado como executado corretamente quando:

- a) o ângulo formado na abdução do ombro é maior ou igual a 90°;
- b) o ângulo do cotovelo é maior ou igual a 160°;
- c) o ângulo formado entre o braço e o tronco no plano frontal está entre 80° e 100°;
- d) a altura dos ombros direito e esquerdo são semelhantes, para fornecer uma equivalência no peso aplicado por ambos sob o tronco;
- e) o ângulo de abdução atual do braço é maior que antes;
- f) volta-se o braço para a posição de repouso.

Para reduzir as chances de o usuário fazer um movimento errado foi implementado uma interface que oferece um *feedback* imediato. Esta interface exibe: a pontuação atual do paciente, uma mensagem de felicitações a cada cinco movimentos corretos, uma mensagem indicando que o paciente deve voltar à posição de relaxamento, uma marcação na parte do corpo que está fora do plano correto e um alvo indicando à que posição o braço deve chegar. Além disso, é exibida uma barra de estado que vai sendo carregada à medida que o movimento é completado. Na Figura 4 é demonstrado como a interface exibe um exercício efetuado de maneira correta (FREITAS et al., 2012).

Figura 4 - Movimento executado corretamente



Fonte: Freitas et al. (2012).

Para avaliar a aplicabilidade do protótipo foram feitos testes com três tipos diferentes de pessoas: três profissionais de fisioterapia, a fim de criar um parecer técnico sobre a ferramenta. Quatro pacientes adultos, para validar a utilidade e o nível de interação e três pacientes idosos, devido ao fato destes comporem grande parte da população que necessita de tratamentos de reabilitação motora (FREITAS et al., 2012).

Na avaliação feita pelas 10 pessoas submetidas aos testes, os principais pontos avaliados negativamente foram o tamanho reduzido da interface, a clareza das informações e

a falta de estímulo. De modo positivo foi citada a motivação e a satisfação dos usuários, além da facilidade no uso (FREITAS et al., 2012).

Com os testes efetuados, verificou-se que o Kinect possui boa sensibilidade referente aos movimentos feitos. Quanto ao protótipo proposto, foram submetidas a ele 50 repetições do exercício feitas de maneira correta e 60 praticadas incorretamente. Em todos os movimentos corretos a pontuação foi marcada, obtendo 100% de êxito em todos os casos, inclusive em testes com o paciente sentado e em diferentes ângulos em relação ao sensor. Com o nível de acerto apresentado, os autores concluíram que o uso do Kinect em tratamentos deste tipo é aplicável (FREITAS et al., 2012).

3 DESENVOLVIMENTO DA FERRAMENTA

Neste capítulo, na seção 3.1, serão expostos os requisitos da ferramenta a ser desenvolvida. Já na seção 3.2, estarão descritos os dados referentes a etapa de especificação da ferramenta. Na seção 3.3 serão demonstrados os dados sobre a implementação da ferramenta. E por fim, a seção 3.4, trará os resultados obtidos seguidos de uma discussão referente aos mesmos.

3.1 REQUISITOS DA FERRAMENTA A SER DESENVOLVIDA

Os requisitos da ferramenta foram levantados juntamente com acadêmicos da área de fisioterapia. A ferramenta para auxílio a tratamentos fisioterápicos com o Kinect deverá:

- a) captar os movimentos dos usuários (Requisito funcional - RF);
- b) exibir os esqueletos na tela (RF);
- c) calibrar o esqueleto exibido de acordo com a estatura do usuário (RF);
- d) permitir cadastrar um exercício (RF);
- e) permitir cadastrar níveis de dificuldade para um exercício (RF);
- f) permitir salvar os exercícios cadastrados (RF);
- g) permitir executar um exercício cadastrado (RF);
- h) permitir mudar o nível de dificuldade de um exercício durante sua execução (RF);
- i) permitir o controle da ferramenta por comandos de voz (RF);
- j) ser desenvolvido com base na biblioteca Kinect SDK 1.6 (Requisito não-funcional - RNF);
- k) ser desenvolvido para utilização no sistema operacional Windows 7 ou superior (RNF);
- l) ser desenvolvido sob a plataforma .NET Framework 4.5 (RNF);
- m) ser desenvolvido utilizando a linguagem de programação C# (RNF);
- n) utilizar a tecnologia *Windows Presentation Foundation* (WPF) para construção das interfaces com o usuário (RNF);
- o) utilizar o banco de dados SQL Server 2012 (RNF);
- p) permitir que um exercício contenha no máximo dez peças (Regra de Negócio - RN);
- q) permitir que um exercício contenha no máximo cinco estágios (RN);
- r) captar os movimentos de no máximo três usuários (RN).

3.2 ESPECIFICAÇÃO DA FERRAMENTA

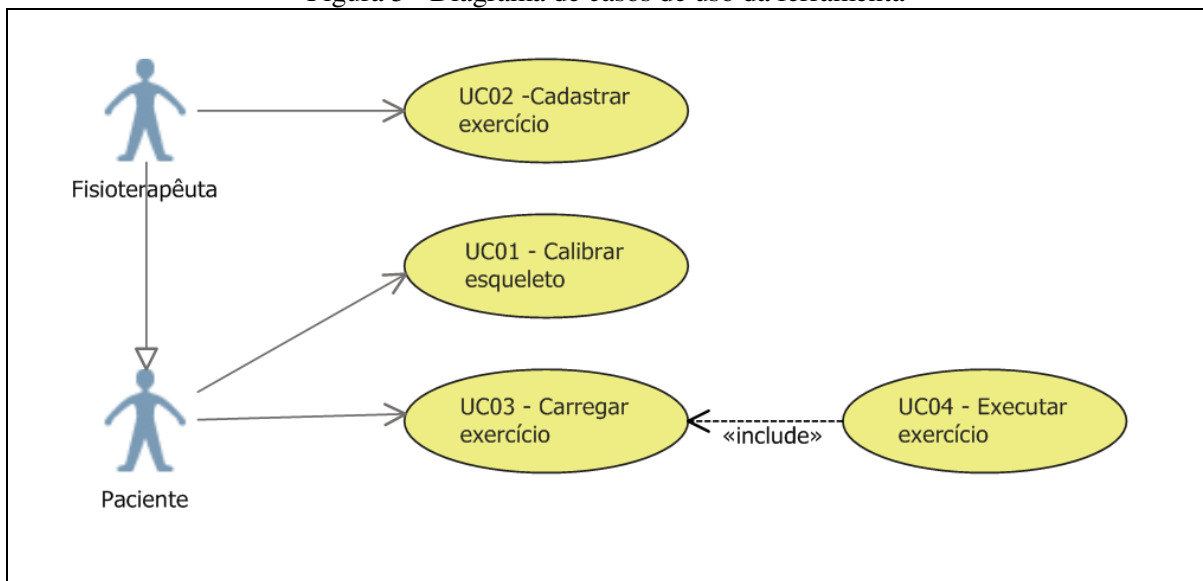
A especificação do trabalho foi feita usando a *Unified Modeling Language* (UML), para especificação dos diagramas de casos de uso, de sequência e de classes, todos foram desenvolvidos usando o ambiente de desenvolvimento Visual Studio 2012.

3.2.1 Diagrama de casos de uso

Nessa seção será apresentado o diagrama de casos de uso (Figura 5), onde são demonstradas as funções exercidas pelos usuários da ferramenta.

A Figura 5 demonstra o diagrama de casos de uso da ferramenta. No caso apresentado o paciente irá carregar e executar os exercícios que serão cadastrados pelo fisioterapeuta. Este último, por sua vez, poderá realizar o cadastramento dos exercícios além de poder executar as funções de um paciente. Por fim, o paciente e, conseqüentemente, o fisioterapeuta terão acesso ao caso de uso calibrar esqueleto.

Figura 5 - Diagrama de casos de uso da ferramenta

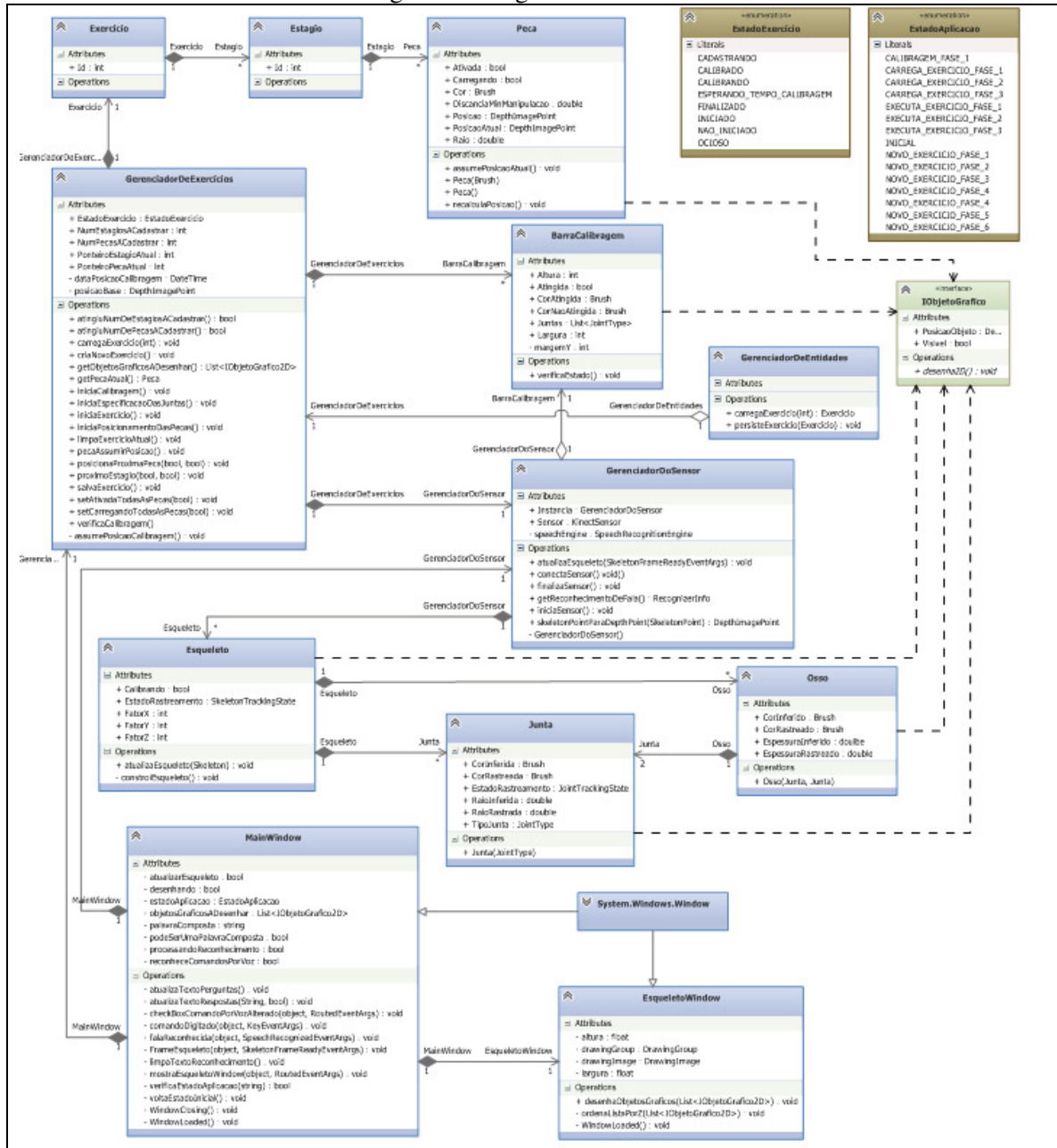


Os cenários principais, alternativos e exceções de cada caso de uso demonstrado na Figura 5 podem ser vistos no APÊNDICE A.

3.2.2 Diagrama de classes

Na Figura 6, encontra-se o diagrama de classes da ferramenta de auxílio a tratamentos fisioterapêuticos, onde pode-se observar os relacionamentos e a estrutura de cada classe. O diagrama foi construído utilizando a ferramenta Visual Studio 2012. Na sequência será feita uma breve descrição da utilidade de cada classe.

Figura 6 - Diagrama de Classes



O sistema possui duas classes de interface gráfica com o usuário, a primeira é a MainWindow, onde será exibido o estado do sistema e as opções disponíveis, também por onde será feita a entrada dos comandos que irão controlar a ferramenta. A segunda classe de interface gráfica com o usuário é a EsqueletoWindow, que foi criada exclusivamente para desenhar os objetos gráficos da ferramenta, como os esqueletos e peças dos exercícios. Para que um objeto de alguma classe existente na ferramenta seja desenhado pela classe EsqueletoWindow, a classe desse objeto deverá implementar a interface IObjetoGrafico.

A classe que controla os exercícios e a calibragem dos esqueletos é a classe `GerenciadorDeExercicios`. Os métodos dessa classe serão chamados pela classe `MainWindow` de acordo com os comandos passados à ferramenta pelo usuário.

A classe `GerenciadorDoSensor` realizará a conexão ao sensor e irá ler as informações dos esqueletos rastreados pelo mesmo. Esta mesma classe manterá atualizada, com informações captadas pelo sensor, uma instancia da classe `Esqueleto` representando o esqueleto principal. Além disso, haverá uma lista de objetos, também da classe `Esqueleto`, representando os esqueletos secundários rastreados pelo sensor, quando existirem. A classe `GerenciadorDoSensor` será criada usando o padrão *singleton*, assim, sempre que um ou mais objetos dentro da ferramenta precisem saber a posição de determinada junta do esqueleto, eles obterão a informação de uma mesma fonte.

Já a classe `Esqueleto` foi criada para representar o corpo dos usuários dentro da aplicação, a classe contém objetos das classes `Ossos` e `Junta`. Todas as três classes implementam a interface `IObjetoGrafico`, para que o esqueleto possa ser exibido para o usuário.

Os exercícios são representados pelas classes `Exercicio`, `Estagio` e `Peca`. Uma instância da classe `Exercicio` conterá uma lista de objetos da classe `Estagio`, esta última representa um nível de dificuldade dentro de um exercício. Cada instancia da classe `Estagio` contém uma lista de objetos da classe `Peca`. Sendo que cada instancia de `Peca` armazenará sua posição no espaço 3D.

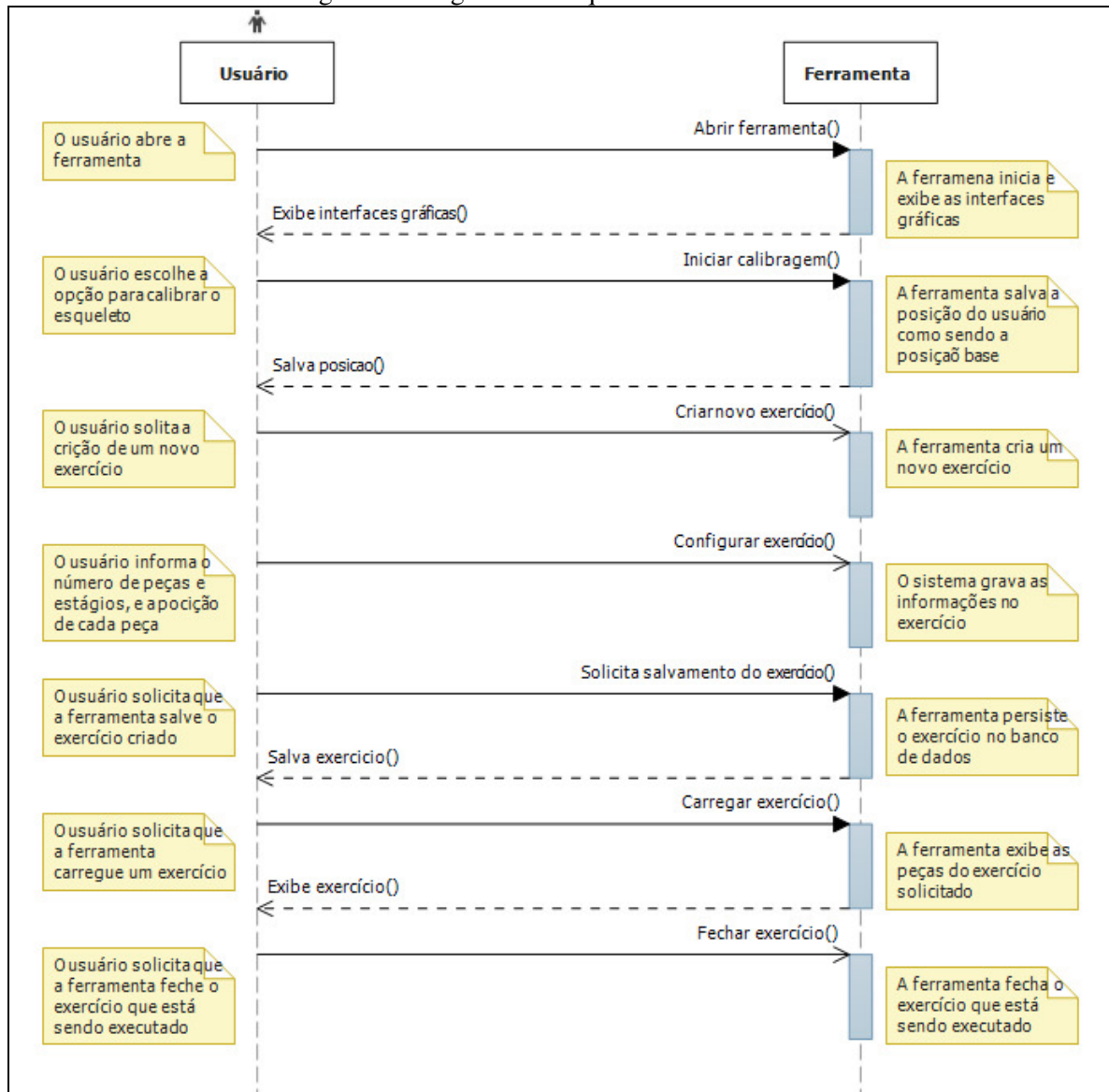
Para persistir e resgatar os objetos do banco de dados foi definida a classe `GerenciadorDeEntidades`, que usará entidades criadas a partir da tecnologia *Entity Framework* para transformar objetos das classes `Exercicio`, `Estagio` e `Peca` em objetos do banco de dados que poderão ser persistidos, podendo também depois serem resgatados pela classe `GerenciadorDeEntidades` em um processo inverso.

Para auxiliar na calibragem foi definida a classe `BarraCalibragem`, que implementará a interface `IObjetoGrafico` e será controlada por uma instancia da classe `GerenciadorDeExercicios`.

3.2.3 Diagrama de sequência

A seguir, na Figura 7, é exibido o diagrama de sequência da ferramenta, nele é demonstrando como ocorrerá a interação entre o usuário e a ferramenta.

Figura 7 - Diagrama de sequência da ferramenta



No diagrama de sequência exibido na Figura 7, pode-se observar como ocorre a troca de mensagem entre o usuário e a ferramenta. Inicialmente, o usuário irá abrir a ferramenta, a mesma vai carregar todas as janelas interfaces gráficas e os demais componentes. Numa outra troca de mensagem o usuário solicita que seja iniciada a calibragem do seu esqueleto. A ferramenta então desenha os objetos que auxiliam na calibragem e aguarda que o usuário posicione-se na distância correta para gravar a posição base.

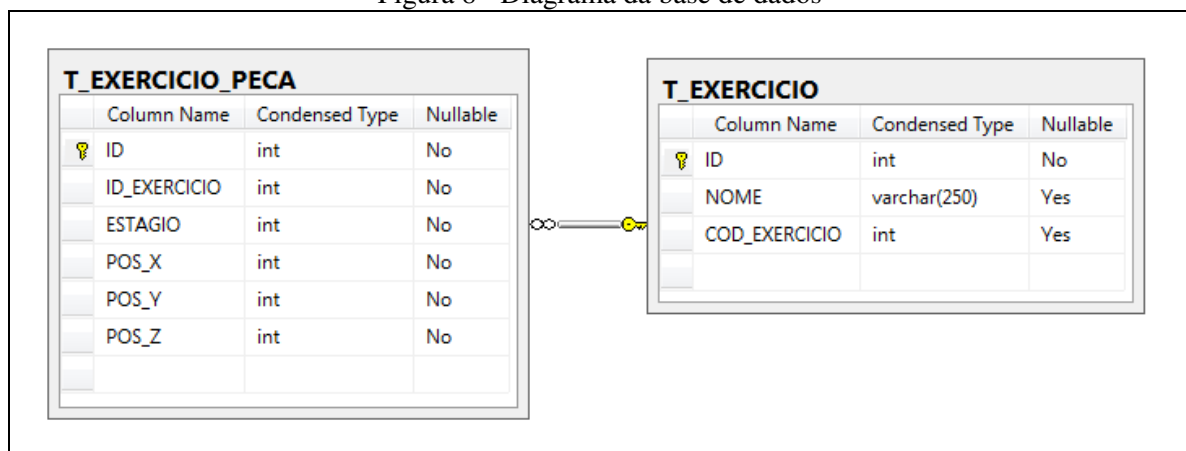
Depois de calibrado o usuário pode solicitar o cadastramento de um novo exercício, feito isto, o usuário deve fornecer os dados do exercício, como número de peças e estágios e a posição de cada peça. Ao fim da configuração o usuário deve solicitar que o exercício seja salvo. A ferramenta, por sua vez, salva o exercício no banco de dados.

Por fim, o usuário poderá solicitar que um exercício seja carregado, ao fazer isto, a ferramenta vai resgatar o exercício no banco de dados e exibi-lo na tela. E, no momento que achar necessário, o usuário poderá solicitar que o exercício seja fechado pela ferramenta.

3.2.4 Base de dados

A base de dados foi criada com base no diagrama apresentado na Figura 8. Nela podem ser vistos as tabelas, os tipos de dados das colunas e as chaves primárias e estrangeiras.

Figura 8 - Diagrama da base de dados



A partir da Figura 8 pode-se observar que existem duas tabelas, sendo que a tabela T_EXERCICIO possui três colunas: o ID para armazenar o código único para os exercícios, a coluna NOME que armazenará o nome dos exercícios e a coluna COD_EXERCICIO que conterà um código também único para os exercícios.

Já a tabela T_EXERCICIO_PECA contém 6 colunas: a coluna ID que possui um valor único para cada peça, a coluna ID_EXERCICIO que armazena o identificador único do exercício ao qual a peça pertence, a coluna ESTAGIO que indica em qual estágio a peça deve aparecer, e por fim, as colunas POS_X, POS_Y e POS_Z que armazenam as posições no espaço 3D de cada peça.

3.3 IMPLEMENTAÇÃO DA FERRAMENTA

Nesta seção serão apresentadas as técnicas e ferramentas utilizadas no desenvolvimento da ferramenta de auxílio a tratamentos fisioterapêuticos. Inicialmente irão ser apresentados detalhes do ambiente de desenvolvimento utilizado e ao final, informações sobre a ferramenta desenvolvida.

3.3.1 Ferramentas utilizadas

A linguagem de programação escolhida para desenvolvimento foi a C# da plataforma Microsoft .NET. A escolha deve-se ao fato de a linguagem ser de fácil entendimento, e por ela possuir uma grande comunidade de desenvolvedores, tornando-a uma das mais conceituadas no mercado com uma ampla documentação disponível na web.

O Sistema Gerenciador do Banco de Dados (SGBD), escolhido foi o SQL Server 2012 Express, o principal motivo que levou à escolha é o fato de também ser uma tecnologia da plataforma .NET, oferecendo facilidade na integração do banco de dados, quando comparadas com as demais tecnologias empregadas no desenvolvimento, principalmente na utilização do conjunto de tecnologias *Entity Framework*. Além disso, o SQL Server 2012, também é um banco de dados bem conceituado no mercado, mostrando robustez e bom desempenho, entre os demais bancos de dados relacionais disponíveis. A versão utilizada foi a Express, que é gratuita e supre tranquilamente às necessidades da ferramenta a ser desenvolvida.

As interfaces da ferramenta foram criadas usando a tecnologia *Windows Presentation Foundation* (WPF), que é uma tecnologia que facilita a criação de GUI ricas, pois utiliza a *eXtensible Application Markup Language* (XAML) e oferece uma separação clara entre a interface com o usuário e a lógica de negócios. Além de oferecer suporte aos desenhos 2D necessários na ferramenta.

A implementação da ferramenta foi feita utilizando o ambiente de desenvolvimento Visual Studio 2012. Ferramenta desenvolvida pela Microsoft foi escolhida devida à facilidade no uso e por ser propriamente criada para o desenvolvimento de aplicações para a plataforma .NET. O Visual Studio oferece recursos que auxiliam desde a criação das GUI com WPF, até o uso da tecnologia *Entity Framework* no controle de objetos do banco de dados.

Para realizar a entrada dos comandos para controle da ferramenta, foi utilizada a tecnologia de reconhecimento de fala da Microsoft, a *Microsoft Speech Technologies*. O uso desta tecnologia visa melhorar a interação do usuário com a ferramenta, permitindo o seu controle à distância sem o uso dos dispositivos de entrada tradicionais mouse e teclado. Foi optado pelo idioma inglês, pois é o idioma com o maior aprimoramento dentre os demais disponíveis, e também porque até o momento não existe suporte para reconhecimento de fala em português brasileiro.

3.3.2 Implementação da ferramenta

Para realizar a implementação da ferramenta foram criadas as classes especificadas no diagrama de classes da Figura 6, nesta seção serão demonstrados os trechos de código mais relevantes, seguidos de uma breve descrição do seu respectivo funcionamento.

Para controlar a ferramenta podem ser usados comandos por voz, que são reconhecidos utilizando a biblioteca `Microsoft.Speech`. Os comandos são compostos por palavras específicas que devem ser ditas no idioma inglês. Esta lista de palavras possíveis forma uma gramática, que é especificada no recurso `SpeechGrammar` existente dentro da ferramenta. Um trecho da especificação da gramática pode ser visto no Quadro 2, onde são inseridas na gramática as palavras *hip*, *cancel* e *new*.

Quadro 2 - Gramática para reconhecimento dos comandos de voz

```

<item>
  <tag>HIP</tag>
  <one-of>
    <item> hip </item>
  </one-of>
</item>
<item>
  <tag>CANCEL</tag>
  <one-of>
    <item> cancel </item>
  </one-of>
</item>
<item>
  <tag>NEW</tag>
  <one-of>
    <item> new </item>
  </one-of>
</item>

```

A ferramenta possui três classes principais, as classes `MainWindow`, `GerenciadorDeExercícios` e `GerenciadorDoSensor`. A classe `MainWindow` juntamente com a classe `EsqueletoWindow`, estendem da classe `Window` e são as classes que compõem a camada de interface com o usuário.

A interface `IObjetoGrafico2D`, foi criada para ser implementada por todos os objetos da ferramenta que deverão ser desenhados na tela. Ela possui o método `desenha2D` que serve para pedir aos objetos implementadores desta interface para desenharem à si mesmos. A interface possui ainda as propriedades `PosicaoObjeto` e `Visivel`, indicando a posição do objeto no espaço 3D e se está visível ou não, respectivamente.

Inicialmente foram criadas as classes `Esqueleto`, `Junta` e `Osso`. As classes `Osso` e `Junta` implementam a interface `IObjetoGrafico2D`, pois serão objetos que deverão ser

desenhadas na tela, ambas possuem propriedades para configurar a cor e a espessura delas ao serem desenhadas. Na classe `Junta`, tem-se o campo `PosicaoObjeto`, que define a posição da junta no espaço 3D, um osso é a ligação entre duas juntas, ou seja, sua posição varia de acordo com a posição das suas respectivas juntas.

No Quadro 3 é exibida parte da codificação da classe `Junta`. Nela pode ser visto um exemplo da implementação da interface `IObjetoGrafico2D`. Como dito anteriormente, o método `desenha2D` da linha 34 serve para pedir para que o objeto desenhe a si próprio na posição indicada pela propriedade `PosicaoObjeto`. no caso da junta, é feito de acordo com o estado dela, sendo que ela pode estar rastreada, inferida ou não rastreada. A verificação do estado da junta inicia na linha 40. No caso do objeto peça, na tela de desenho, ela é representada por uma circunferência de raio e cor definidos conforme a verificação do estado da mesma. O desenho é feito na linha 62 utilizando o método `DrawEllipse` da classe `System.Windows.Media.DrawingContext`.

Quadro 3 - Implementação da interface `IObjetoGrafico2D` na classe `Junta`

```

30 // região IObjetoGrafico2D
31 public DepthImagePoint PosicaoObjeto { get; set; }
32 public bool Visivel { get; set; }
33
34 public void desenha2D(System.Windows.Media.DrawingContext contexto)
35 {
36     double raio;
37     Brush cor;
38
39     // define a cor e o tamanho de acordo com o estado do rastreamento
40     if (this.EstadoRastreamento == JointTrackingState.Tracked) // verifica se a junta está rastreada
41     {
42         this.Visivel = true;
43         raio = this.RaioRastrada;
44         cor = this.CorRastreada;
45     }
46     else if (this.EstadoRastreamento == JointTrackingState.Inferred) // verifica se a junta está inferida
47     {
48         this.Visivel = true;
49         raio = this.RaioInferida;
50         cor = this.CorInferida;
51     }
52     else
53     {
54         // caso não esteja nem inferida nem rastreada, não a desenha
55         this.Visivel = false;
56         raio = 0;
57         cor = null;
58     }
59
60     // se deve ser visível, desenha a elipse na cor e tamanho pegos acima
61     if (this.Visivel)
62         contexto.DrawEllipse(cor, null, new Point(this.PosicaoObjeto.X, this.PosicaoObjeto.Y), raio, raio);
63 }
64 // fim região IObjetoGrafico2D

```

A classe `GerenciadorDoSensor`, é responsável por prover a conexão com o sensor. O método `conectaSensor` da linha 43 (Quadro 4) percorre uma lista estática de possíveis sensores retornados pela classe `Microsoft.Kinect.KinectSensor` verificando se algum deles está conectado. Encontrando o primeiro, atribui este à variável `sensor`, o que o tornará o sensor usado por toda a ferramenta.

Quadro 4 - Rotina que conecta ao sensor Kinect

```

42 // procura e conecta no primeiro sensor disponível
43 public void conectaSensor()
44 {
45     // verifica se existe sensores na lista
46     if (KinectSensor.KinectSensors.Count == 0)
47         throw new Exception("No Kinect Found!");
48     //percorre os possíveis sensores
49     foreach (var possivelSensor in KinectSensor.KinectSensors)
50     {
51         // verifica se o sensor está conectado
52         if (possivelSensor.Status == KinectStatus.Connected)
53         {
54             // pega o primeiro sensor conectado
55             this.sensor = possivelSensor;
56             break;
57         }
58     }
59 }

```

Ainda na classe `GerenciadorDeSensor`, existe o método `iniciaSensor`. Este, atribui à variável `speechEngine` as funções para reconhecimento da fala do usuário no idioma inglês. Na mesma classe o método `atualizaEsqueleto` da linha 155 (Quadro 5) foi criado para manter atualizado os esqueletos na ferramenta. Nas linha 162 é criada uma nova instancia de `OpenSkeletonFrame` e após isso na linha 169 são copiados para uma lista todos os esqueletos que o sensor está decifrando no momento.

Quadro 5 - Rotina para atualizar os esqueletos (parte 1)

```

154 // atualiza os esqueletos da aplicação
155 public void atualizaEsqueleto(SkeletonFrameReadyEventArgs skeletonFrameReadyEventArgs)
156 {
157     Skeleton[] esqueletosKinect = new Skeleton[0];
158     Skeleton esqueletoKinectPrincipal = null;
159     this.esqueletos.Clear(); // limpa a lista de esqueletos
160
161     // abre o frame
162     using (SkeletonFrame skeletonFrame = skeletonFrameReadyEventArgs.OpenSkeletonFrame())
163     {
164         // verifica se abriu
165         if (skeletonFrame != null)
166         {
167             // copia os esqueletos para um array
168             esqueletosKinect = new Skeleton[skeletonFrame.SkeletonArrayLength];
169             skeletonFrame.CopySkeletonDataTo(esqueletosKinect);
170         }
171         // percorre os esqueletos
172         foreach (var esqueletoKinect in esqueletosKinect)
173         {
174             // verifica se está rastreado
175             if (esqueletoKinect.TrackingState == SkeletonTrackingState.Tracked)
176             {

```

Quando chamado, o método `atualizaEsqueleto` percorre a lista de esqueletos encontrada, o primeiro esqueleto será considerado como sendo o principal e será atualizado na linha 198 (Quadro 6) através do método `atualizaEsqueleto` da classe `Esqueleto`. Caso a

lista possua mais de um esqueleto rastreado, é instanciado um novo objeto `Esqueleto` para cada um deles, que são atualizados da mesma maneira que o esqueleto principal.

Quadro 6 - Rotina para atualizar os esqueletos (parte 2)

```

194         // se achou um esqueleto principal
195         if (esqueletoKinectPrincipal != null)
196         {
197             // atualiza o esqueleto principal
198             this.esqueletoPrincipal.atualizaEsqueleto(esqueletoKinectPrincipal);
199             this.esqueletos.Add(this.esqueletoPrincipal);
200         }

```

Como especificado na seção 3.2.2, as classes `MainWindow` e `EsqueletoWindow` compõem juntas a camada de interface com o usuário. A classe `MainWindow` é responsável pela maior parte do controle da ferramenta, nela existe o método `frameEsqueleto` da linha 547 que é invocado a cada novo frame do esqueleto captado pelo sensor, todo o método pode ser visto no Quadro 7. Na linha 558 é feita a chamada para o gerenciador do sensor atualizar os esqueletos da ferramenta. Já na linha 580 é passada para a tela auxiliar a lista de objetos gráficos que devem ser desenhados naquele momento, a lista foi retornada pelo gerenciador de exercícios na linha 578.

Quadro 7 - Procedimento executado a cada novo frame captado pelo sensor Kinect

```

547     private void FrameEsqueleto(object sender, SkeletonFrameReadyEventArgs e)
548     {
549         // verifica se já está desenhando
550         if (this.desenhando)
551             return;
552
553         // marca como desenhando
554         this.desenhando = true;
555
556         // se deve atualizado o esqueleto, o faz
557         if (this.atualizarEsqueleto)
558             this.gerenciadorDoSensor.atualizaEsqueleto(e);
559
560         // verifica se está calibrando
561         if (this.gerenciadorDeExercicios.EstadoExercicio == EstadoExercicio.CALIBRANDO
562             || this.gerenciadorDeExercicios.EstadoExercicio == EstadoExercicio.ESPERANDO_TEMPO_CALIBRAGEM)
563         {
564             // pede para verificar a calibragem
565             this.gerenciadorDeExercicios.verificaCalibragem();
566
567             // se calibrou com sucesso, volta ao estado inicial
568             if (this.gerenciadorDeExercicios.EstadoExercicio == EstadoExercicio.CALIBRADO)
569             {
570                 this.voltaEstadoInicial();
571             }
572         }
573
574         // verifica se a tela auxiliar está iniciada
575         if (this.esqueletoWindow.IsInitialized)
576         {
577             // se está, solicita os objetos gráficos a desenhar
578             this.objetosGraficosADesenhar = this.gerenciadorDeExercicios.getObjetosGraficosADesenhar();
579             // envia para a tela auxiliar desenhar
580             this.esqueletoWindow.desenhaObjetosGraficos(this.objetosGraficosADesenhar);
581         }
582
583         // volta para não desenhando
584         this.desenhando = false;
585     }

```

Para controle da ferramenta na classe `MainWindow`, foi criado o método `verificaEstadoAplicacao`, onde é passado como parâmetro o comando no formato de *string* para que o método verifique o estado da atual ferramenta e notifique o gerenciador de exercícios sobre o que fazer. Um exemplo pode ser visto no Quadro 8, onde o método reconhece o comando armazenado na variável `resposta` para iniciar a calibragem (linha 210) e em seguida invoca o método do gerenciador de exercícios (linha 216).

Quadro 8 - Rotina para solicitar o início da calibragem ao gerenciador de exercícios

```

209 // se estava no estado inicial e o usuário pediu para iniciar a calibragem
210 if (estadoAplicacao == EstadoAplicacao.INICIAL && resposta == "CALIBER")
211 {
212     // a palavra CALIBER já foi reconhecida, portanto não é uma palavra composta
213     this.podeSerUmaPalavraComposta = false;
214     reconheceu = true;
215     this.estadoAplicacao = EstadoAplicacao.CALIBRAGEM_FASE_1; // indica o novo estado da aplicação
216     gerenciadorDeExercicios.iniciaCalibragem(); // pede para o gerenciador de exercícios iniciar a calibragem
217     return reconheceu; // retorna
218 }

```

Outro método essencial criado na classe `MainWindow` é o `falaReconhecida` presente na linha 150 (Quadro 9). Ele é invocado sempre que o sensor captar alguma palavra dita pelo usuário. O reconhecimento da palavra deve atender ao limite de confiança especificado (linha 165). A palavra, se reconhecida, é passada para o método `verificaEstadoAplicacao` (linha 175) que, como mostrado anteriormente, vai tratar de atualizar a ferramenta. O método `falaReconhecida` pode ser visto abaixo no Quadro 9.

Quadro 9 - Procedimento executado quando um comando de voz é reconhecido

```

150 private void falaReconhecida(object sender, SpeechRecognizedEventArgs e)
151 {
152     // verifica se é para reconhecer os comandos
153     if (!this.reconheceComandosPorVoz)
154         return;
155     // verifica se já está processando algum texto
156     if (this.processandoReconhecimento)
157         return;
158     // marca com processando
159     processandoReconhecimento = true;
160
161     // define um nível de confiança do reconhecimento
162     const double ConfidenceThreshold = 0.3;
163
164     // verifica se atingiu o nível imposto
165     if (e.Result.Confidence >= ConfidenceThreshold)
166     {
167         // pega a string de resposta
168         String resposta = e.Result.Semantics.Value.ToString();
169
170         // verifica se tem algum inicio de palavra composto acumulada
171         if (this.podeSerUmaPalavraComposta)
172             resposta = palavraComposta + " " + resposta;
173
174         // chama metodo que atualiza a aplicação de acordo com a resposta
175         Boolean reconheceu = verificaEstadoAplicacao(resposta);
176
177         // atualiza as respostas e as perguntas
178         atualizaTextoRespostas(resposta, reconheceu);
179         atualizaTextoPerguntas();
180     }
181
182     // marca como não processando
183     processandoReconhecimento = false;
184 }

```

Para que o esqueleto possa ser exibido em uma janela separada, foi criada a janela `EsqueletoWindow`. Ela é formada por um componente `ViewBox`, que possui um `Image`, neste último é que são desenhados os objetos que implementam a interface `IObjetoGrafico2D`, retornados pela ferramenta. O método na classe `EsqueletoWindow`, que desenha os objetos é o `desenhaObjetosGraficos`. Com o intuito de melhorar ao máximo a sensação de profundidade na classe `EsqueletoWindow`, foi implementado o método `ordenaListaPorZ` (linha 64 do Quadro 10) para o qual, é enviada a lista de objetos gráficos antes de desenhá-los para que sejam ordenados decrescentemente pelo valor da propriedade `Depth`, que define a profundidade dos objetos no espaço 3D. O algoritmo utilizado para ordenar a lista foi o método bolha, a implementação do mesmo pode ser vista no Quadro 10.

Quadro 10 - Rotina para ordenar a lista de objetos gráficos

```

64 private void ordenaListaPorZ(List<IObjetoGrafico2D> objetosGraficos2D)
65 {
66     // percorre os elementos
67     for (int i = 0; i < objetosGraficos2D.Count; i++)
68     {
69         // j começa sempre sempre 1 elemento a frente de i
70         for (int j = i + 1; j < objetosGraficos2D.Count; j++)
71         {
72             // verifica qual é mais profunda (Z maior)
73             if (objetosGraficos2D[j].PosicaoObjeto.Depth > objetosGraficos2D[i].PosicaoObjeto.Depth)
74             {
75                 // se o elemento da frente j tiver o Z maior, troca com o elemento atual i
76                 IObjetoGrafico2D objeto = objetosGraficos2D[j];
77                 objetosGraficos2D[j] = objetosGraficos2D[i];
78                 objetosGraficos2D[i] = objeto;
79             }
80         }
81     }
82 }

```

Para controlar as principais funções da ferramenta foi criada classe `GerenciadorDeExercicios`. Ela foi implementada para receber “ordens” do restante da ferramenta e, a partir disso, manusear os objetos que compõem o cadastro e execução dos exercícios, além da calibragem do esqueleto.

Para realizar a calibragem do esqueleto a classe `GerenciadorDeExercicios`, possui uma lista com duas instancias da classe `BarraDeCalibragem` que implementam a interface `IObjetoGrafico2D`. As barras de calibragem são duas barras retangulares que serão desenhadas na tela. Uma das barras vai aparecer sempre sobre pés do esqueleto e a outra vai ficar a uma distancia fixa acima da primeira barra. O objetivo é que o usuário se mova para frente e para trás até que a barra superior fique sobre a cabeça do esqueleto por mais de dois segundos (linha 376 do Quadro 11). Assim, não importará qual a estatura do paciente, o esqueleto sempre manterá a mesma proporção na tela. O trecho do método `verificaCalibragem` que confere a calibragem pode ser visto no Quadro 11.

Quadro 11 - Procedimento para verificar a calibragem do esqueleto

```

343 // verifica o estado de cada barra pra ver se foi atingida
344 foreach (BarraCalibragem b in this.barrasCalibragem)
345 {
346     b.verificaEstado();
347     // se um barra ainda não foi marca como false
348     if (!b.Atingida)
349     {
350         todasPecasAtingidas = false;
351         break;
352     }
353 }
354
355 // se todas as barras fora atingidas, guarda o tempo atual e muda o status para ESPERANDO_TEMPO_CALIBRAGEM
356 if (todasPecasAtingidas)
357 {
358     this.EstadoExercicio = EstadoExercicio.ESPERANDO_TEMPO_CALIBRAGEM;
359     if (this.dataPosicaoCalibragem == null)
360         this.dataPosicaoCalibragem = DateTime.Now;
361 }
362 else
363 {
364     // senão volta para o stats CALIBRANDO
365     this.EstadoExercicio = EstadoExercicio.CALIBRANDO;
366     this.dataPosicaoCalibragem = null;
367 }
368
369 // se toda as peças fora atingidas e foi iniciado o tempo de contagem
370 if (todasPecasAtingidas && this.dataPosicaoCalibragem.HasValue)
371 {
372     // subtrai do tempo atual o tempo armazenado
373     TimeSpan tempoQueEstaCalibrado = DateTime.Now.Subtract(this.dataPosicaoCalibragem.Value);
374
375     // verifica se são mais de 2 segundos
376     if (tempoQueEstaCalibrado.Seconds >= 2)
377     {
378         // caso sim, guarda os valores da calibragem e muda o status para CALIBRADO
379         this.assumePosicaoCalibragem();
380         this.gerenciadorDoSensor.Esqueleto.Calibrando = false;
381         this.EstadoExercicio = EstadoExercicio.CALIBRADO;
382     }
383 }

```

Quanto ao controle dos exercícios e das peças, podem se destacar dois métodos da classe `GerenciadorDeExercicios`. O primeiro é o `iniciaPosicionamentoDasPecas` da linha 138 (Quadro 12) que durante o cadastramento, depois do usuário ter informado o número de peças e estágios, e depois de todas as peças terem recebido as juntas que as carregarão, faz com que a ferramenta inicie o posicionamento das peças no espaço 3D. A classe gerenciadora de exercícios vai iniciar carregando a primeira peça do primeiro estágio (linha 150) até que o usuário informe o comando para a peça ser solta na posição pretendida.

Quadro 12 - Procedimento para iniciar o posicionamento das peças

```

138 public void iniciaPosicionamentoDasPecas()
139 {
140     // desativa todas as peças
141     this.setAtivadaTodasAsPecas(false);
142     // não carrega nenhuma peça
143     this.setCarregandoTodasAsPecas(false);
144
145     // posiciona os ponteiros no primeiro estágio da primeira peça
146     this.PonteiroPecaAtual = 0;
147     this.PonteiroEstagioAtual = 0;
148     // ativa e carrega a peça atual
149     this.getPecaAtual().Ativada = true;
150     this.getPecaAtual().Carregando = true;
151 }

```

O segundo método relevante na classe `GerenciadorDeExercicios` é o `iniciaExercicio` da linha 206 (Quadro 13). Este, após um exercício ter sido carregado do banco de dados, marca todas as peças como ativas para que depois, o método `getObjetosGraficosADesenhar`, retorne para desenhar na tela as peças do estágio selecionado pelo usuário. No Quadro 13 está a codificação do método `iniciaExercicio`.

Quadro 13 - Procedimento que inicia a execução de um exercício

```
206 public void iniciaExercicio()
207 {
208     // configura todas as peças do exercicio
209     this.setAtivadaTodasAsPecas(true);
210     this.setCarregandoTodasAsPecas(false);
211
212     // marca exercicio como iniciado
213     this.EstadoExercicio = EstadoExercicio.INICIADO;
214 }
```

Outro ponto que vale destacar é como a ferramenta interage com o banco de dados. Para realizar essa interação foi criada a classe `GerenciadorDeEntidades`, que possui dois métodos conversores: o primeiro é o `persisteExercicio`, que transforma uma instancia da classe `Exercicio` em entidades `T_EXERCICIO` e `T_EXERCICIO_PECA` que são persistidas por ele no banco utilizando a tecnologia *Entity Framework*. O segundo método é o `carregaExercicio` da linha 43 (Quadro 14), que faz a tarefa inversa, carregando as entidades de um exercício do banco de dados, instanciando um novo objeto da classe `Exercicio` e retornando-o para a ferramenta. No Quadro 14, se pode ver o método `carregaExercicio`, que recebe como parâmetro o código do exercício que se deseja carregar e, a partir do código, consulta as entidades no banco de dados (linhas 51 e 59).

Quadro 14 - Procedimento que carrega um exercício do banco de dados

```

43 public Exercício carregaExercicio(int id)
44 {
45     // instancia um novo exercício
46     Exercício exercicio = new Exercício();
47     // inicia um novo objeto do banco de dados
48     using (ASSIST_FISIOEntities entities = new ASSIST_FISIOEntities())
49     {
50         // pesquisa na tabela T_EXERCICIO pelo exercício de código passado como parâmetro
51         T_EXERCICIO _T_EXERCICIO = entities.T_EXERCICIO.Find(id);
52         // pega o id e cria uma nova lista de estágios
53         exercicio.Id = _T_EXERCICIO.COD_EXERCICIO.Value;
54         exercicio.Estagios = new List<Estagio>();
55         // coloca o id numa nova variável para consultar as peças
56         int idExercicio = _T_EXERCICIO.ID;
57
58         // para cada peça retornada pela consulta em T_EXERCICIO_PECA
59         foreach (T_EXERCICIO_PECA _T_EXERCICIO_PECA in (
60             from ep in entities.T_EXERCICIO_PECA
61             where ep.ID_EXERCICIO == idExercicio orderby ep.ESTAGIO ascending select ep))
62         {
63             // se o código do estágio não contiver um índice na lista
64             if (exercicio.Estagios.Count <= _T_EXERCICIO_PECA.ESTAGIO)
65             {
66                 // instancia um novo estágio e o coloca na lista de estágios do exercício
67                 Estagio estagio = new Estagio(_T_EXERCICIO_PECA.ESTAGIO);
68                 estagio.Pecas = new List<Peca>();
69                 exercicio.Estagios.Add(estagio);
70             }
71             // gera nova peça, define sua posição e adiciona na lista do estágio correspondente
72             Peca peca = new Peca();
73             peca.Cor = Util.getProximaCor();
74             peca.PosicaoAtual = Util.getNewDepthImagePoint(
75                 _T_EXERCICIO_PECA.POS_X, _T_EXERCICIO_PECA.POS_Y, _T_EXERCICIO_PECA.POS_Z);
76             peca.assumePosicaoAtual();
77             exercicio.Estagios[_T_EXERCICIO_PECA.ESTAGIO].Pecas.Add(peca);
78         }
79     }
80     // retorna o exercício carregado
81     return exercicio;
82 }

```

Para efetuar a troca de um estágio, deve-se atualizar a variável `PonteiroEstagioAtual` da classe `GerenciadorDeExercicios` com o código do estágio desejado. Assim, o método `getObjetosGraficosADesenhar` da mesma classe vai retornar apenas as peças correspondente ao estágio indicado na variável `PonteiroEstagioAtual`, como pode ser visto na linha 269 do Quadro 15.

Quadro 15 - Procedimento que retorna as peças a serem desenhadas

```

263 // verifica se tem algum exercício carregado e o status
264 if (this.exercicio != null)
265 {
266     if (this.EstadoExercicio == EstadoExercicio.INICIADO || this.EstadoExercicio == EstadoExercicio.FINALIZADO)
267     {
268         // desenha todas as peças do estágio atual
269         foreach (Peca peca in this.exercicio.Estagios[this.PonteiroEstagioAtual].Pecas)
270         {
271             //adiciona elas a lista
272             objetosGraficosADesenhar.Add(peca);
273         }
274     }

```

3.3.3 Operacionalidade da ferramenta

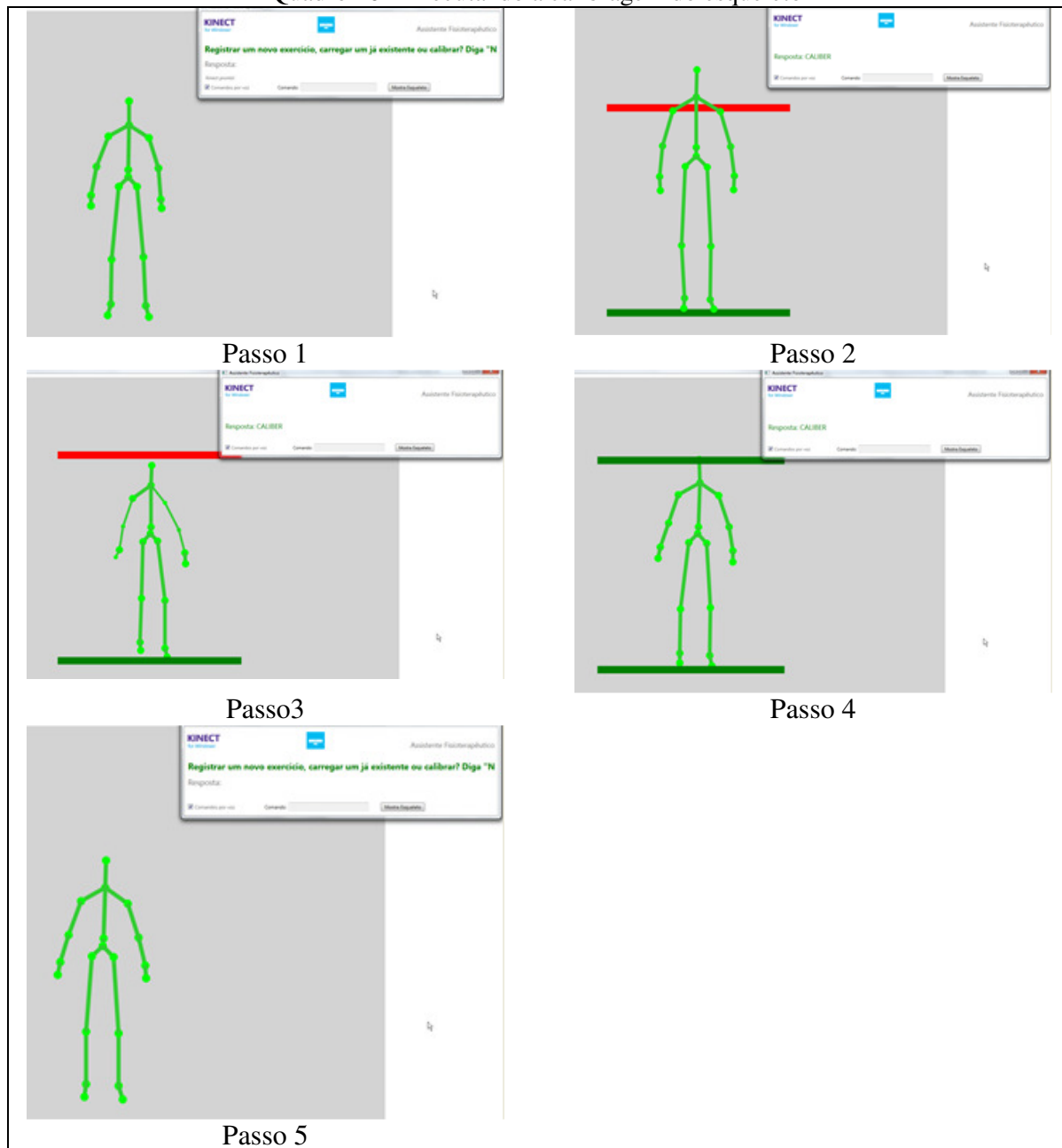
A seguir serão demonstrados, utilizando um exemplo básico, como funciona a calibragem de um esqueleto, o cadastramento e execução de um exercício. A ferramenta é controlada por comandos digitados no campo `Comando` ou por comandos de voz. Para que a

ferramenta reconheça os comandos de voz, a caixa de marcação *Comandos por voz* da tela principal deve estar marcada. Essa opção de não reconhecimento de comandos fez-se necessária devida à necessidade de usar a ferramenta em ambientes barulhentos, onde palavras involuntárias poderiam ser captadas pela ferramenta, atrapalhando o seu uso.

3.3.3.1 Calibrando o esqueleto

No Quadro 16 são demonstrados os passos do processo de calibragem do esqueleto feito pelo usuário. Inicialmente a ferramenta encontra-se no estado inicial (passo 1). Para solicitar o início da calibragem deve-se dizer/digitar a palavra “*caliber*”. Feita a solicitação, serão desenhadas as duas barras de calibragem que podem ser vistas no passo 2.

Quadro 16 - Executando a calibragem do esqueleto



Observando o Quadro 16, é possível notar que, com as duas barras desenhadas, o usuário deve posicionar-se de modo que as barra inferior e superior fiquem sobre os pés e cabeça do esqueleto. No passo 2 pode-se ver que, no exemplo, quando iniciada a calibragem o esqueleto está à frente da posição indicada, pois a altura atual do esqueleto é maior que a altura base indicada pelas barras. Já no passo 3, o esqueleto ficou menor, pois o usuário afastou-se do sensor. Por fim, com o usuário aproximando-se um pouco do sensor, foi encontrada a distância ideal, sendo que as duas barras ficaram sobre a cabeça e pés do usuário. Para indicar o posicionamento correto ambas as barras ficaram na cor verde, como pode ser visto no passo 4. Após isso o usuário deve manter a posição por dois segundos, para

que a calibragem seja confirmada, as barras desapareçam e a ferramenta volte ao estado inicial (passo 5).

Durante o processo de calibragem do esqueleto o usuário pode optar por cancelar o processo. Para tanto, basta entrar com o comando “*cancel*” a qualquer momento, que o sistema volta ao estado inicial.

3.3.3.2 Criando um novo exercício

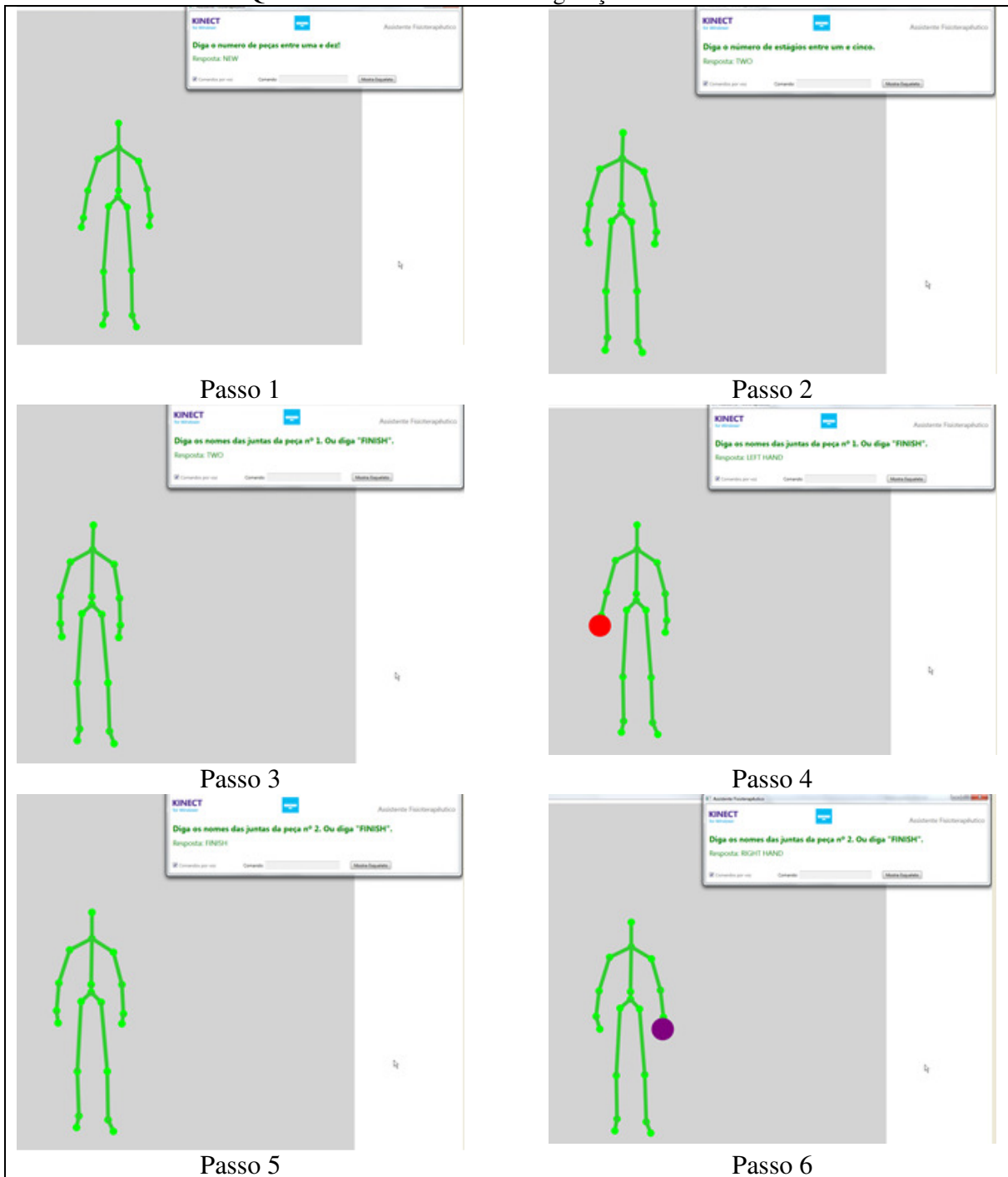
O segundo processo é o de cadastramento de exercícios, tendo como início o passo a passo na Quadro 17. Para iniciá-lo o usuário deve, com a ferramenta no estado inicial, dizer/digitar a palavra “*new*”, como aparece no passo 1. Feito isto, a ferramenta vai solicitar que seja informado o número de peças que o exercício conterà, o usuário deverá informar um número inteiro. No exemplo, no passo 2, nota-se que na resposta exibida na tela foram informadas duas (*two*) peças. Logo em seguida o sistema solicita o número de estágios que o exercício terá. Outra vez deve ser informado um valor inteiro que, no exemplo, foi novamente dois (passo 3).

Ainda no passo 3, do exemplo exibido no Quadro 17, a ferramenta solicita que seja informada a junta do esqueleto que vai conduzir a primeira peça. Logo em seguida o usuário a informa. No exemplo dado pode-se ver, no passo 4, que foi escolhida a junta que corresponde à mão esquerda (*left hand*) do usuário. Assim, a peça aparece sobre a junta passando a acompanhar os movimentos da mesma. Nesse estado, o usuário pode informar mais juntas, nesses casos a peça vai aparecer no ponto central entre as duas primeiras juntas listadas.

Um ponto importante a ser observado, é como devem ser ditas as expressões com duas palavras. No exemplo dado no Quadro 17, o usuário informa a expressão “*left hand*” no passo 4 indicando que a junta correspondente à mão esquerda vai movimentar a primeira peça. Nesses casos, as palavras devem ser ditas separadamente, com uma breve pausa para que o sensor possa separá-las. Quando um comando for digitado, ao invés de falado, as palavras podem ser escritas tanto separadas como juntas. Sempre que a ferramenta identificar que uma expressão dita/digitada pode ter uma segunda palavra, o campo “resposta” da tela assumirá a cor amarela até que a segunda palavra que compõe a expressão seja dita/escrita.

Seguindo a explicação sobre o cadastramento das juntas que conduzem as peças é possível ver, no passo 5 do Quadro 17, que após dizer as juntas que conduzem uma peça, o usuário deverá dizer o comando “*finish*” para ir para próxima peça a ser configurada, se houver.

Quadro 17 - Definindo as configurações do exercício



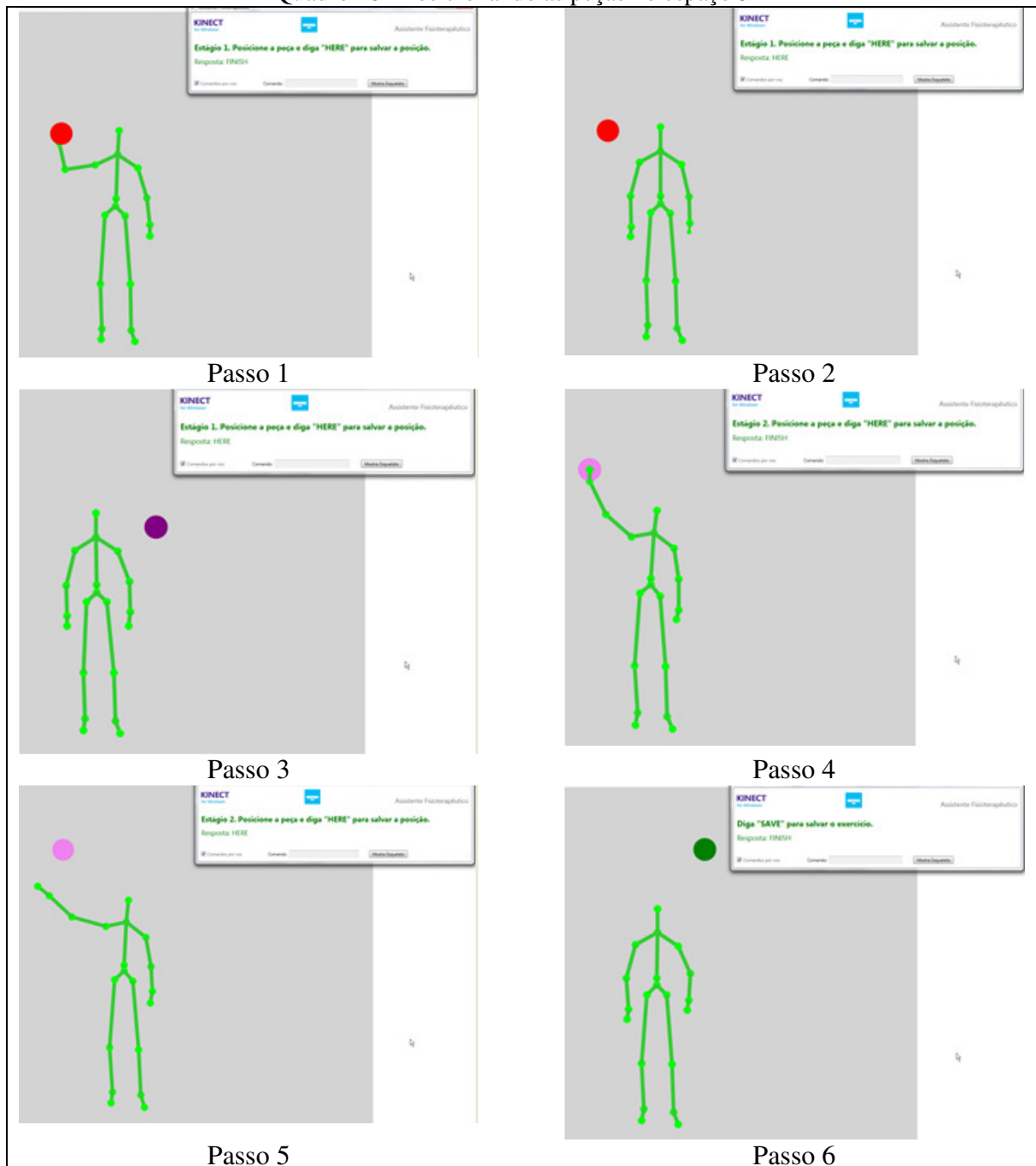
Com todas as peças configuradas com as juntas que as irão conduzir, deve ser finalizado o cadastramento do exercício posicionando as peças no espaço 3D. No exemplo usado, o usuário informou que o exercício conteria duas peças e teria dois estágios, assim, como pode ser visto no Quadro 18, é iniciado o posicionamento da primeira peça no primeiro estágio. Então, como visto no passo 1, o usuário conduz a peça até o local desejado. Nessa parte o usuário deve usar o comando “*here*” para soltar a peça, como visto no passo 2.

Caso uma peça tenha sido posicionada de maneira equivocada, o usuário tem a possibilidade de corrigir levando à(s) junta(s) até o local correto depois dizendo/escrevendo novamente o comando “*here*”. Isso pode ser feito quantas vezes for necessário.

Para passar para a próxima peça, se houver, o usuário deve entrar com o comando “*finish*”. Caso todas as peças de um estágio tenham sido cadastradas, é passado para a primeira peça do próximo estágio, e assim por diante (passo 2 ao 6 do Quadro 18). Quando não houver mais peças nem estágios para cadastrar, o sistema pede se o usuário deseja salvar através do comando “*save*”, como mostrado no passo 6 do Quadro 18. Após salvar, a ferramenta volta ao estado inicial.

Em qualquer passo do cadastramento dos exercícios é possível cancelar a operação, voltando a ferramenta ao seu estado inicial. O comando usado para isto deve ser o “*cancel*”.

Quadro 18 - Posicionando as peças no espaço 3D



3.3.3.3 Executando um exercício

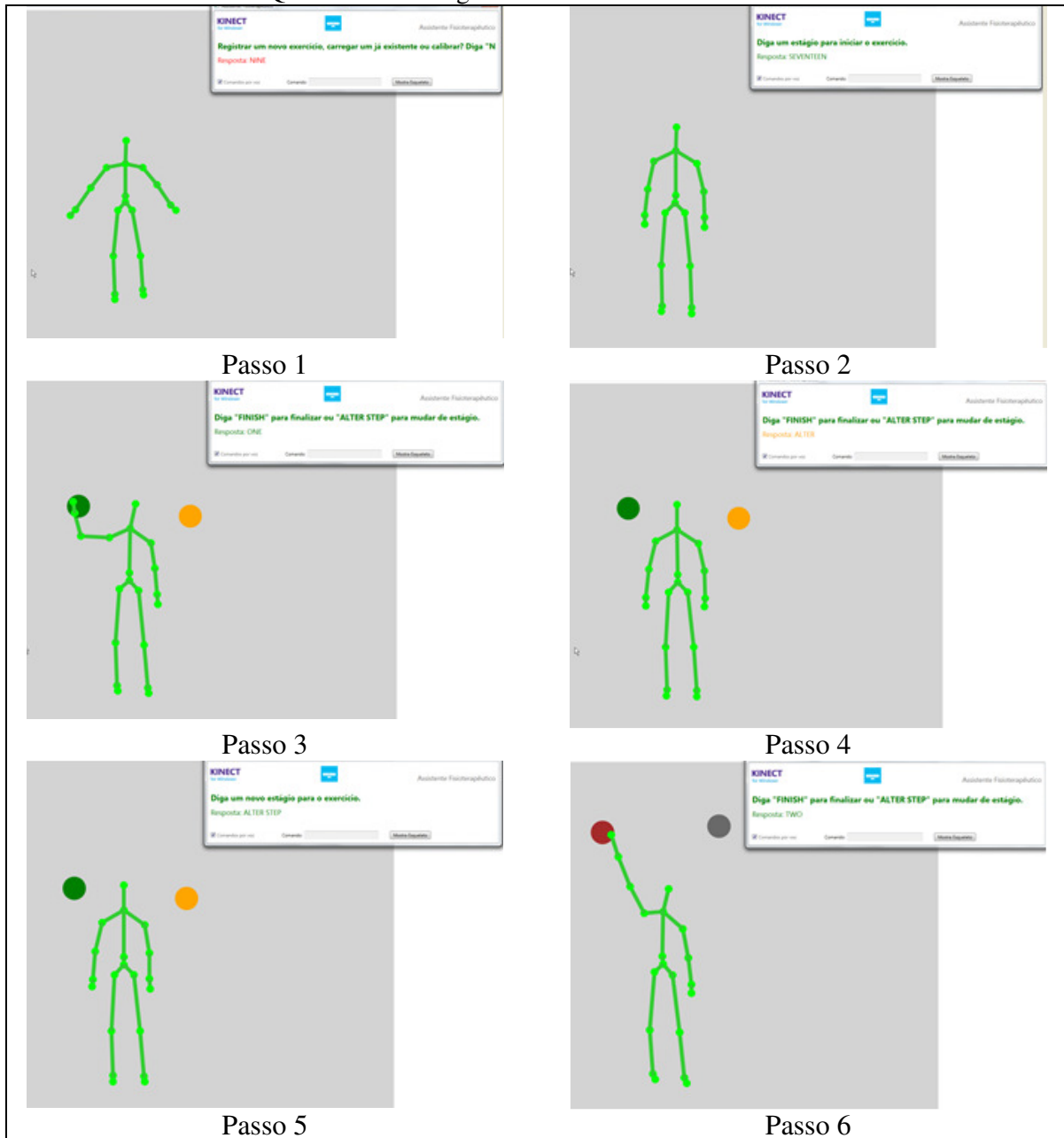
Tendo exercícios salvos no banco de dados com seus respectivos códigos, a ferramenta permite carregá-los para que o usuário os execute. Para iniciar o carregamento, o comando utilizado é o “load”, visto no passo 1 (Quadro 19). Logo na sequência será solicitado o código do exercício desejado. Deverá ser informado então o código (numérico inteiro) de um exercício cadastrado. É possível ver um exemplo desse processo no Quadro 19, onde, no

passo 2, o usuário informou que desejava carregar o exercício de código dezessete (*seventeen*).

Ao carregar o exercício, a ferramenta solicita que seja informado o número do estágio desejado, como pode ser observado no passo 3 (Quadro 19) do exemplo, onde o usuário informou o estágio um (*one*). Assim, como pode ser visto no passo 4, o exercício carregado está pronto para que o usuário possa usá-lo no estágio selecionado. Para alterar um estágio de um exercício em execução basta informar o comando “*alter step*”, o sistema então solicitará que seja informado um novo estágio (passo 5). Assim que este último seja informado, ele é carregado na tela e estará pronto para ser utilizado.

Para terminar a execução de um exercício e voltar para tela inicial, pode-se usar os comandos “*cancel*” ou “*finish*”.

Quadro 19 - Carregando e executando um exercício



3.4 RESULTADOS E DISCUSSÃO

Esta seção tem por objetivo apresentar a metodologia e os resultados obtidos durante o processo de avaliação da ferramenta desenvolvida.

3.4.1 Avaliação da ferramenta

O que mais se objetiva com os testes é avaliar a aplicabilidade da ferramenta nos tratamentos fisioterapêuticos, levantando possíveis aperfeiçoamentos a serem feitos desde que o uso da tecnologia aplicada mostre-se viável.

Para os testes, optou-se por fazer uma pesquisa qualitativa do tipo experimental exploratória que, segundo Neves (1996, p. 1), costuma ser direcionada e não busca enumerar ou medir eventos. Geralmente, esse tipo de pesquisa não emprega instrumental estatístico para análise dos dados. Seu foco é bastante amplo partindo de uma perspectiva diferente da adotada pelos métodos quantitativos. Neves ainda complementa que “nas pesquisas qualitativas, é frequente que o pesquisador procure entender os fenômenos, segundo a perspectiva dos participantes da situação estudada e, a partir, daí situe sua interpretação dos fenômenos estudados”.

Com o objetivo de fazer uma pré-avaliação da ferramenta antes de usá-la com pacientes reais, na tentativa de aperfeiçoá-la ao máximo antes do seu uso em testes mais sérios, este trabalho contou com a participação de duas acadêmicas do curso de Fisioterapia da Fundação Universidade Regional de Blumenau – FURB, também em fase de conclusão. O objetivo foi cooperar do desenvolvimento de ambos os trabalhos de conclusão de curso, onde o trabalho feito pelas acadêmicas de Fisioterapia é intitulado como “A aplicabilidade do sensor Kinect como ferramenta de auxílio na reabilitação neurológica”.

3.4.2 Versão inicial reprovada

Inicialmente, foi criado um protótipo da ferramenta com objetivo de funcionar como uma espécie de jogo, onde os pacientes precisavam atingir alvos com os membros definidos no cadastramento para que um alvo se apagasse e um próximo ficasse visível. Esse ciclo era repetido durante um número de vezes definido ao carregar o exercício.

Porém, com os primeiros testes feitos utilizando os próprios acadêmicos como pacientes, constatou-se que poderia haver uma grande dificuldade em executar os exercícios por parte de indivíduos com uma maior incapacidade motora. Isso poderia ocasionar um esforço demasiado e ainda assim, não conseguiriam prosseguir para os próximos alvos, e conseqüentemente, não finalizariam os exercícios com sucesso, prejudicando de maneira grave o tratamento.

Uma alteração fez-se necessária: os alvos não deveriam mais se apagar para que outro ficasse visível, mas sim, todos os alvos de um estágio deveriam ficar visíveis de uma só vez. Com isso, a ferramenta passaria a ter um papel apenas direcional para o paciente durante um exercício, ou seja, os alvos passariam a servir apenas como referências para qual direção os membros dos pacientes deveriam ser movimentados. A reponsabilidade de verificar se um paciente atingiu um alvo ou não, deixou de ser da ferramenta e passou para o fisioterapeuta

que estará acompanhado o paciente, ficando sobre seu critério decidir se o ponto alcançado pelo paciente está ou não dentro do perímetro esperado.

Retirando a obrigatoriedade em atingir os alvos com membros específicos, a ferramenta ficou mais flexível, pois uma configuração de exercício, antes criados para fortalecimento de determinada lista de membros, passou a permitir o uso da mesma configuração para outros exercícios com finalidades diferentes.

3.4.3 Versão final

Após a rejeição da versão inicial, foi feita uma readequação nos requisitos e na especificação da ferramenta. Ao final chegou-se à uma nova versão que pôde ser utilizada nos testes com os pacientes. Nos tópicos a seguir serão apresentados e debatidos os resultados obtidos a partir desta versão final.

3.4.4 Amostragem e instrumentos de coleta de dados

Para os testes com pacientes reais, foi selecionada um total de seis pacientes de idade adulta, de ambos os sexos, que apresentavam algum tipo de disfunção motora. Os critérios de exclusão na filtragem dos pacientes foram: crianças, indivíduos com déficit cognitivo associado ou que tenham algum déficit visual.

Para coleta dos dados foram utilizados os seguintes instrumentos:

- a) termo de Consentimento Livre e Esclarecido (TCLE) disponível no APÊNDICE B. Todos os pacientes foram voluntários para participar e deram seu consentimento através deste;
- b) Mini Exame do Estado Mental (MEEM) disponível no APÊNDICE C. O objetivo do exame é avaliar o status cognitivo dos pacientes, uma vez que é necessário que eles entendam as tarefas que serão propostas na pesquisa. Na avaliação, os indivíduos que concluíram a 5^o série do ensino fundamental deverão apresentar uma pontuação mínima de 23, já para os que não concluíram a 5^o série a pontuação mínima é 19. Somente os indivíduos que apresentarem desempenho cognitivo correspondente ao seu grau de instrução estarão aptos a participar do estudo;
- c) a ferramenta desenvolvida. Foi utilizada a versão final da ferramenta especificada e desenvolvida neste trabalho;
- d) questionário disponível no APÊNDICE D. O questionário contém questões visando a captação de alguns dados pessoais dos pacientes juntamente com 10 perguntas sobre a aplicabilidade da ferramenta desenvolvida.

3.4.5 Procedimentos para coleta de dados

A pesquisa foi realizada no campus III na Clínica de Fisioterapia na Universidade Regional de Blumenau – FURB, localizado no endereço: Rua São Paulo, nº 2171, Bairro Itaipava Seca, Cidade de Blumenau – Santa Catarina.

Após o paciente ter assinado o TCLE e ter atingido à pontuação mínima no seu respectivo MEEM, cada um dos seis pacientes foi orientado a efetuar uma série de até sete exercícios disponibilizados através da ferramenta desenvolvida que variavam dependendo do grau de incapacidade motora de cada indivíduo. A seção de exercícios demorava em média 1 hora por paciente. As datas de realização dos testes de cada paciente estão disponíveis no Quadro 20. Ao fim das sessões os pacientes foram submetidos ao questionário qualitativo.

Quadro 20 - Data dos testes

Paciente	Data do teste
Paciente A	22/05/2013
Paciente B	22/05/2013
Paciente C	22/05/2013
Paciente D	27/05/2013
Paciente E	27/05/2013
Paciente F	27/05/2013

3.4.6 Análise qualitativa dos resultados

Todos os seis pacientes atingiram a pontuação mínima do MEEM. Todos também realizaram a seção de exercícios sem dificuldades que impedissem a finalização dos testes com sucesso. Ao fim, os pacientes responderam ao questionário qualitativo. Com base nas respostas de cada um, foi montado o Quadro 21.

Quadro 21 - Respostas dos pacientes ao questionário

Pergunta / Paciente	A	B	C	D	E	F
1 - Você gosta das sessões de Fisioterapia convencional?	Sim	Sim	Sim	Sim	Sim	Sim
2 - Há quanto tempo você recebe tratamento fisioterapêutico? (meses)	8	24	24	10	36	11
3 - Você já havia recebido algum tipo de tratamento fisioterapêutico que utilizasse algum recurso diferente (videogame, computador)?	Não	Não	Não	Não	Não	Não
4 - Se sim na resposta anterior, você ainda realiza esse tratamento?						
5 - Você acredita ter sido fácil compreender a proposta das atividades do Kinect?	Sim	Não	Sim	Sim	Sim	Sim
6 - Você acredita que a realização das atividades propostas na sessão de fisioterapia, tornaram-se mais fáceis com o uso do Kinect?	Sim	Não	Sim	Não	Sim	Sim
7 - Existe alguma atividade proposta que você considerou de difícil compreensão?	Não		Não	Não	Não	Não
8 - Existe alguma atividade proposta que você acredita ter sido mais difícil de executar?	Não		Sim	Não	Não	Não
9 - Você acredita que as atividades realizadas com o auxílio do Kinect, sejam próximas de atividades que você realiza no seu dia-a-dia? (comer, alcançar algum objeto, sentar, caminhar).	Sim	Sim	Sim	Sim	Sim	Sim
10 - Se você tivesse oportunidade, gostaria de participar de algum tratamento em Reabilitação Neurológica que utilize o sensor Kinect como ferramenta?	Sim	Não	Sim	Sim	Sim	Sim

No Quadro 21 estão representadas todas as respostas dos pacientes que participaram dos testes da ferramenta. As perguntas com respostas vazias, com exceção a pergunta 4, estão sem respostas pelo fato de que o paciente não soube responder.

Observando o Quadro 21, é possível ver que o uso de ferramentas computacionais em tratamentos fisioterápicos foi novidade para todos os pacientes, isso se comprova com base nas respostas à pergunta 3. Outro ponto importante que correspondeu de maneira positiva à pesquisa, foi o fato de todos os exercícios propostos ficarem parecidos com atividades cotidianas dos pacientes, como pode ser visto nas respostas à pergunta 9.

Como visto na pergunta 6, a maioria dos pacientes concordou que as atividades ficaram mais fáceis utilizando a ferramenta se comparado aos métodos tradicionais de fisioterapia.

As perguntas 7 e 8, complementam o resultado da questão 6, onde pode-se deduzir que os exercícios foram de fácil entendimento para a maioria, não encontrando maiores dificuldades na hora de executá-los.

Na pergunta 10 nota-se que os pacientes ficaram satisfeitos com o uso da ferramenta no tratamento, sendo que a maioria gostaria de participar novamente em outras oportunidades.

3.4.7 Comparativo com os trabalhos correlatos

A partir dos resultados obtidos pela ferramenta proposta neste trabalho através dos testes com os pacientes, no Quadro 22 é demonstrado um comparativo com os trabalhos correlatos citados na seção 2.3.

Quadro 22 - Comparação com os trabalhos correlatos

características / trabalhos relacionados	Trabalho Proposto	Bresciani e Conto (2012)	Ferraz e Yamashita (2012)	Freitas et al. (2012)
1 - É uma ferramenta específica?	Sim	Não	Sim	Sim
2 - Realizou testes com pacientes reais?	Sim	Sim	Não	Sim
3 - Permite a inserção de novos exercícios?	Sim	Não	Não	Não
4 - Pode ser utilizar em diferentes tipos de tratamento fisioterápicos?	Sim	Não	Não	Não
5 - Faz avaliação dos movimentos?	Não	Não	Não	Sim

Com base nos dados mostrados no Quadro 22, pode-se perceber que a ferramenta desenvolvida neste trabalho se sobressai em alguns pontos se comparada às ferramentas propostas nos demais trabalhos citados na seção 2.3. A ferramenta proposta neste trabalho é a única que permite a inserção de exercícios dinamicamente (questão 3), ou seja, o fisioterapeuta pode cadastrar inúmeros exercícios de acordo com a sua necessidade (questão 4). Porém vê-se que a ferramenta desenvolvida por Freitas et al. (2012) fornece a avaliação dos movimentos, dando um *feedback* em tempo real se estes forem feitos de maneira errada, funcionalidade esta que não foi implementada na ferramenta desenvolvida neste trabalho (questão 5).

4 CONCLUSÕES

Avaliando os resultados e observando os aspectos computacionais da pesquisa, percebe-se que o uso do Kinect foi considerado fácil pela maior parte dos pacientes, pois a maioria relatou que o seu uso é de fácil execução. Quanto aos exercícios propostos também afirmaram que gostariam de usar o Kinect nos tratamentos em outras oportunidades.

Observando os pacientes durante os testes, pôde-se confirmar que, apesar de os movimentos do esqueleto mapeado pelo Kinect corresponderem bem aos movimentos realizados no mundo real pelos usuários, existe algumas melhorias que contribuiriam ainda mais no uso desse tipo de ferramenta na área da fisioterapia. A principal dificuldade encontrada com relação à precisão da leitura dos movimentos surge quando o usuário cruza dois membros ou quando membros do corpo de dois ou mais usuários se sobrepõe, isso fez com que alguns exercícios fossem descartados antes dos testes.

Outro problema que foi contornado de maneira provisória na ferramenta é o fato de pessoas com as mais diversificadas estaturas precisarem executar os mesmos exercícios. O módulo de calibragem desenvolvido na ferramenta foi à maneira mais fácil encontrada de resolver esse problema, tornando a ferramenta menos confortável, uma vez que pode ser considerado um obstáculo a mais para pacientes com dificuldades de locomoção, por exemplo.

A opção de usar comandos de voz para controlar a ferramenta também teve seu uso aprovado. Porém, existem locais onde ela não deve ser utilizada. Geralmente nestes locais existem ruídos externos, conversa paralela durante o uso da ferramenta ou são locais onde a fala reproduz eco, tudo isso, dificulta a interpretação das palavras pelo sistema de reconhecimento. Outra dificuldade está no fato de o reconhecimento de voz só funcionar no idioma inglês, visto que nem todos os usuários têm domínio deste idioma.

Todas as ferramentas utilizadas na construção da ferramenta, não apresentaram nenhuma limitação, destaque para o ambiente de desenvolvimento Visual Studio 2012, que supriu com tranquilidade a criação de todos os componentes utilizados no projeto, desde a especificação da ferramenta até os testes finais da mesma.

Os pacientes mostraram otimismo em relação ao uso deste tipo de recurso computacional nos seus respectivos tratamentos de reabilitação. Em sua maioria, relataram que se sentiram mais motivados na realização dos exercícios com o Kinect em comparação ao modelo de tratamento tradicional. E também demonstraram interesse em utilizar a ferramenta novamente, em futuras oportunidades.

Por fim, levando em consideração os testes, principalmente os feitos utilizando pacientes reais, considerou-se a ferramenta, bem como o seu uso, positiva a tratamentos de reabilitação fisioterápicos. O sensor Kinect, mostrou-se hábil para uso em ferramentas voltadas à fisioterapia, apesar de ainda mostrar limitações devido à imprecisão em certos movimentos, como dito anteriormente.

4.1 EXTENSÕES

Apesar dos resultados obtidos terem sido positivos, a ferramenta ainda permite que sejam feitas uma série de melhorias, tanto referentes ao funcionamento, como referentes à interface com o usuário.

Referente à interface, existe desde alterações simples até alterações que envolvem um nível de computação gráfica avançado. Algumas delas são:

- a) criar campos que exibam mais informações sobre os exercícios: atualmente, a ferramenta só exibe as informações referentes aos comandos que o usuário deve informar ou que informou. Dados dos exercícios como o número de peças, número de estágios e tempo de execução não são exibidos pela versão atual;
- b) melhorar a parte gráfica do objetos interativos: o esqueleto e os demais objetos interativos, hoje, são desenhados e exibidos em um ambiente 2D, onde a noção de profundidade é significativamente limitada. No campo da computação gráfica existem alguns métodos que poderiam ser aplicados na ferramenta para melhorar a sensação de profundidade, como adicionar efeitos de sombra e iluminação nos objetos, o desenho de algo que simule chão e paredes, etc.. Outra opção seria utilizar ferramentas gráficas mais avançadas (como o XNA Framework) para mudar o espaço de desenho para uma ambiente 3D, criando uma série de possibilidades que não são possíveis em ambientes 2D;
- c) criar contextos para os exercícios: a ferramenta hoje, exibe na tela um esqueleto formado por linhas simples que ligam as juntas, as peças dos exercícios são apenas circunferências de tamanho fixo e cor única e o ambiente é formado por um fundo de cor cinza. Uma opção seria permitir criar ambientes que simulem lugares cotidianos onde os pacientes realizam suas atividades diárias. Por exemplo, desenhar na tela um ambiente que represente uma cozinha (com prateleiras, armários, etc.), onde as peças sejam representadas por figuras de panelas, pratos,

talheres, etc. Permitindo assim ao fisioterapeuta contextualizar o exercício que o paciente deverá executar;

- d) melhorar a representação do esqueleto: mudar a forma que o corpo do usuário é representado na tela. Exibir um personagem ou exibir a imagem do próprio usuário seriam maneiras de melhorar a interação com a ferramenta.

Referente à funcionalidade da ferramenta, algumas melhorias também poderão ser feitas. Uma delas é melhorar o rastreamento do esqueleto, pois há casos, como quando dois membros se cruzam, onde o sensor retorna a posição de algumas juntas totalmente incoerente à posição das mesmas no mundo real. Também seria interessante aprimorar o processo de leitura, de modo que informações como a rotação dos membros possam ser interpretadas e utilizadas na avaliação dos movimentos, por exemplo.

Outra melhoria seria criar um processo no qual a ferramenta calibre o esqueleto automaticamente, sem que precise ter uma opção exclusiva para isto. Isso garantiria uma maior agilidade durante o cadastramento e execução dos exercícios, e também maior facilidade no uso da ferramenta.

Além disso, poderia ser incorporada na ferramenta funcionalidades referente à avaliação dos movimentos feitos pelo pacientes. Como avaliar se os movimentos de ambos os braços estão similares ou se ângulo formado entre membros de uma junta estão dentro dos limites impostos nos exercícios.

REFERÊNCIAS BIBLIOGRÁFICAS

BRESCIANI, Tiago A.; CONTO, Samuel M. O impacto da tecnologia Nintendo Wii no tratamento fisioterapêutico e na satisfação de pacientes em uma clínica do Vale do Taquari. **Revista Destaques Acadêmicos**, Lajeado, v. 4, n. 1, p. 81-95, 2012.

BRÜCKEIMER, Alessandro D. **Detecção corporal 3D na reabilitação virtual**. 2011. 86 f. Trabalho de Conclusão de Curso (Graduação em Ciência da Computação) – Curso de Graduação em Ciência da Computação, Universidade Estadual de Santa Catarina, Joinville.

CABREIRA, Arthur; MÜLLING, Tobias. Perspectivas para novas interfaces: Kinect e integrações gestuais sob o panorama de interfaces naturais do usuário. In: CONGRESSO SUL AMERICANO DE DESIGN DE INTERACAO, 4., 2012, São Paulo. **Anais eletrônicos...** São Paulo: Universidade Anhembi Morumbi, 2012. Disponível em: <<http://blogs.anhembi.br/isa2012/anais/artigos/23.pdf>>. Acesso em: 5 jun. 2013.

FARIAS, Norma; BUCHALLA, Cassia M. A classificação internacional de funcionalidade, incapacidade e saúde da Organização Mundial da Saúde: conceitos, usos e perspectivas. **Revista Brasileira de Epidemiologia**, São Paulo, v. 8, n. 2, p. 187-193, 2005.

FERNANDES, Carina I. S.; SANTOS Fátima. **Reaprendizagem motora e fisioterapia neurológica – revisão bibliográfica**. 2012. 19 f. Trabalho de Conclusão de Curso (Licenciatura em Fisioterapia) – Departamento de Ciências da Enfermagem e Tecnologias de Saúde, Universidade Fernando Pessoa, Lisboa.

FERRAZ, Leonardo T. D.; YAMASHIDA, Renato K. S. **Desenvolvimento de um jogo para reabilitação utilizando um sensor de com e movimento (Kinect)**. 2012. 7 f. Trabalho de Conclusão de Curso (Graduação em Engenharia Mecatrônica) – Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos, Escola Politécnica da Universidade de São Paulo, São Paulo.

FREITAS, Daniel Q. et al. Development and evaluation of a Kinect based motor rehabilitation game. In: SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL, 11., 2012, Brasília. **Anais eletrônicos...** Brasília: SBC, 2012. Disponível em: <http://sbgames.org/sbgames2012/proceedings/papers/computacao/comp-full_18.pdf>. Acesso em: 3 jun. 2013.

KUCHINSKAS, Susan. Binary body double: Microsoft reveals the science behind project Natal for Xbox 360. **Revista Scientific American**, [S. l.], jan. 2010. Disponível em: <<http://www.scientificamerican.com/article.cfm?id=microsoft-project-natal>>. Acesso em: 3 jun. 2013.

LOWENSOHN, Josh. Timeline: a look back at Kinect's history. **CNET**, [S.l.], fev. 2011. Disponível em: <http://news.cnet.com/8301-10805_3-20035039-75.html>. Acesso em: 3 jun. 2013.

NEVES, José L. Pesquisa qualitativa: características, usos e possibilidades. **Caderno de Pesquisas em Administração**, São Paulo, v. 1., n. 3, 1996. Disponível em <<http://www.ead.fea.usp.br/cad-pesq/arquivos/C03-art06.pdf>>. Acesso em: 3 jul. 2013.

RIBEIRO, Nildo. O ambiente terapêutico como agente otimizador na neuroplasticidade em reabilitação de pacientes neurológicos. **Revista Diálogos Possíveis**, Salvador, v. 4, n. 2, p. 107-117, 2005.

ROCHA, Pollyeverlin R.; DEFAVARI, Alex H.; BRANDÃO, Pierre S. Estudo da viabilidade da utilização do Kinect como ferramenta no atendimento fisioterapêutico de pacientes neurológicos. In: SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL, 11., 2012, Brasília. **Anais eletrônicos...** Brasília: SBGames, 2012. Disponível em: <<http://sbgames.org/sbgames2012/proceedings/papers/gamesforchange/g4c-04.pdf>>. Acesso em: 5 jun. 2013.

SÁ, Jobert G. P. **Construindo uma DSL para reconhecimento de gestos utilizando Kinect**. 2011. 76 f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) - Centro de Informática, Universidade Federal de Pernambuco, Recife.

UMPHRED, Darcy A. **Reabilitação neurológica**. São Paulo: Manole, 2004.

VALERIO NETTO, Antonio V.; MACHADO, Liliane S.; OLIVEIRA, Maria C. F. Realidade virtual – definições, dispositivos e aplicações. *Revista Eletrônica de Iniciação Científica da SBC*, v. 2, n. 2, 2002. Disponível em: <http://www.di.ufpb.br/liliane/publicacoes/2002_reic.pdf>. Acesso em: 5 jun. 2013.

YAMADA, Fernando A. A. et al. Reconstrução de objetos 3D utilizando estruturas de indexação espacial com o Microsoft Kinect. In: WORKSHOP DE REALIDADE VIRTUAL E AUMENTADA, 9., 2012, Paranavaí. **Anais eletrônicos...** Paranavaí: UFJF, 2012. Disponível em: <<http://www.ufjf.br/getcomp/files/2013/03/Reconstru%C3%A7%C3%A3o-de-Objetos-3D-utilizando-Estruturas-de-Indexa%C3%A7%C3%A3o-Espacial-com-o-Microsoft-Kinect.pdf>>. Acesso em: 3 jun. 2013.

ZALEVSKY, Zeev et al. Method and system for object reconstruction. **WIPO**, [S.1.], abr. 2007. Disponível em: <<http://patentscope.wipo.int/search/en/WO2007043036>>. Acesso em: 3 jun. 2013.

APÊNDICE A – Detalhamento dos casos de uso

No caso de uso do Quadro 23 – Caso de uso são demonstrados em detalhes o processo de calibragem de um esqueleto.

Quadro 23 – Caso de uso calibrar esqueleto

UC01 – Calibrar esqueleto	
Pré-condições	01) O sensor Kinect deve estar conectado 02) No mínimo um esqueleto deve estar sendo rastreado 03) O sistema estar no estado inicial
Cenário principal	01) O usuário escolhe a opção de calibragem do esqueleto 02) O sistema exibe as barras superior e inferior para definir o correto posicionamento 03) O usuário posiciona-se na posição correta indicada pelo sistema 04) O sistema aguarda por 2 segundos para certificar-se que o usuário permanecerá na posição correta 05) O sistema ajusta o esqueleto à posição do usuário 06) O sistema volta ao estado inicial * Nos passos 2 e 3 o usuário pode cancelar o processo
Cenário de exceção 1	No passo 4 do cenário principal caso o usuário saia da posição correta indicada antes dos 2 segundos 01) O sistema volta ao passo 3 do cenário principal
Pós-condições	01) O esqueleto estará calibrado 02) O sistema estará no estado inicial

No caso de uso do Quadro 24 é demonstrado o processo de cadastramento de um exercício.

Quadro 24 - Caso de uso cadastrar exercício

UC02 – Cadastrar exercício	
Pré-condições	01) Um sensor Kinect deve estar conectado 02) No mínimo um esqueleto deve estar sendo rastreado 03) Estar conectado ao banco de dados 04) O sistema estar no estado inicial
Cenário principal	01) O usuário escolhe a opção de cadastrar novo exercício 02) O sistema solicita ao usuário o número de peças que ele deseja cadastrar 03) O usuário informa o número de peças 04) O sistema solicita o número de estágios que o exercício conterà 05) O usuário informa o número de estágios 06) O sistema assume a primeira peça como sendo a atual 07) O usuário lista as juntas que irão mover a peça atual 08) O sistema verifica se todas as peças tiveram suas juntas cadastradas 09) O sistema assume a primeira peça do primeiro estágio como sendo a atual 10) O sistema pede para o usuário posicionar a peça atual na posição desejada 11) O usuário posiciona a peça 12) O usuário confirma a posição 13) O sistema verifica se ainda existem peças a serem posicionadas para o estágio atual 14) O sistema verifica se ainda existem estágios com as peças não posicionadas 15) Solicita ao usuário a confirmação, para, se deseja salvar o exercício no banco de dados 16) O sistema salva o exercício no banco de dados 17) O sistema volta ao estado inicial * Em todos os passos do cenário principal o usuário pode cancelar o processo
Cenário exceção 1	No passo 08 se existem peças que ainda não tiveram as juntas especificadas 08.1) O sistema assume a próxima peça como sendo a atual 08.2) O sistema volta para o passo 07
Cenário exceção 2	No passo 13 se ainda existem peças a serem posicionadas no estágio atual 13.1) O sistema assume a próxima peça como sendo a atual 13.2) O sistema volta para o passo 10
Cenário exceção 2	No passo 14 se ainda existe estágios com as peças não posicionadas 14.1) O sistema assume a primeira peça do próximo estágio como sendo a atual 14.2) O sistema volta para o passo 10
Pós-condições	01) O exercício estará salvo 02) O sistema estará no estado inicial

No caso de uso do Quadro 25 está descrito o processo de carregar um exercício cadastrado.

Quadro 25 - Caso de uso carregar exercício

UC03 – Carregar exercício	
Pré-condições	01) Um sensor Kinect deve estar conectado 02) Estar conectado ao banco de dados 03) O sistema estar no estado inicial
Cenário Principal	01) O usuário escolhe a opção de cadastrar o novo exercício 02) O sistema solicita ao usuário o código do exercício desejado 03) O usuário informa o código do exercício 04) O sistema carrega o exercício * Em todos os passos do cenário principal o usuário pode cancelar o processo
Pós-condição	O exercício estará carregado

No caso de uso do Quadro 26 são expostos os procedimentos para executar um exercício carregado.

Quadro 26 - Caso de uso executar exercício

UC04 – Executar Exercício	
Pré-condições	01) Um sensor Kinect deve estar conectado 02) Um exercício ter sido carregado
Pré-condições	01) O sistema solicita ao usuário o estágio no qual ele deseja executar o exercício 02) O usuário informa o estágio 03) O sistema exibe o as peças do estágio selecionado 04) O usuário solicita o fim do exercício 05) Volta para o estado inicial * Em todos os passos do cenário principal o usuário pode cancelar o processo
Cenário alternativo 1	No passo 03 caso o usuário solicite mudar de estágio 01) Retorna ao passo 01 do cenário principal * No passo 01 do cenário alternativo 1 o usuário pode cancelar o processo
Pós-condição	O sistema estará no estado inicial

APÊNDICE B – Termo de consentimento livre e esclarecido

No Quadro 27 está a parte inicial do TCLE submetido aos pacientes antes dos mesmos participarem dos testes.

Quadro 27 – Termo de consentimento livre e esclarecido (parte 1)

TERMO DE CONSENTIMENTO LIVRE E ESCLARECIDO

Acadêmicas: Daniela Cristina Sabel e Giuvana Cássia Soares Vettorello

Título do estudo: A aplicabilidade do sensor Kinect como ferramenta de auxílio na reabilitação neurológica.

Paciente: _____ **D.N:** __/__/__

Responsável legal: _____ **D.N:** __/__/__

Você está sendo convidado a participar do estudo acima citado. O documento abaixo contém todas as informações necessárias sobre a pesquisa que estamos fazendo.

Sua colaboração voluntária neste estudo será de muita importância para nós, mas se desistir a qualquer momento, isso não causará nenhum prejuízo a você.

Estou ciente que: O objetivo geral do estudo é investigar a aplicabilidade do Console X-box e seu acessório Kinect na reabilitação neurofuncional.

Realizado no dia: _____ na Clínica de Fisioterapia da Universidade regional de Blumenau - FURB, atendido (a) pelas acadêmicas do curso de fisioterapia da Universidade Regional de Blumenau.

Em relação às informações prestadas pelas acadêmicas acima indicada durante a consulta, o(a) paciente declara:

- Ter tido oportunidade de realizar perguntas e tirar dúvidas relativas à pesquisa,

No Quadro 28 encontra-se a parte final do TCLE.

Quadro 28 - Termo de consentimento livre e esclarecido (parte 2)

sendo que a mesma foi devidamente esclarecida;

- Estar ciente que a proposta que lhe foi indicada não apresenta garantia de cura ou resultado.

Assim sendo, o(a) paciente declara que LEU, FOI INFORMADO, ENTENDEU E CONCORDOU com os termos acima e, por esta razão, AUTORIZA as acadêmicas identificadas, a realizar o estudo em seu favor. Declara ainda QUE LHE FOI ENTREGUE UMA CÓPIA DO PRESENTE TERMO DE CONSENTIMENTO INFORMADO, para sua guarda e disposição.

Blumenau, ___/___/___ Paciente: _____

Responsável Legal: _____

APÊNDICE C – Mini exame do estado mental

No Quadro 29 está a primeira parte do MEEM aplicado nos pacientes antes de submetê-los às sessões de exercícios.

Quadro 29 – Mini exame do estado mental (parte 1)

<u>Mini-mental</u>		
Identificação:	Pontuação Obtida	Pontuação Máxima
<p><u>ITENS</u></p> <p>ORIENTAÇÃO TEMPORAL: ()</p> <p>Que dia do mês é hoje? Em que mês estamos? Em que ano estamos? Em que dia da semana estamos? Qual a hora aproximada? (considere a variação de +/- 1h)</p>		5
<p>ORIENTAÇÃO ESPACIAL: ()</p> <p>Em que local nós estamos? (consultório, dormitório, sala, apontando o chão) Que local é este aqui? (apontando ao redor num sentido mais amplo: hospital, casa de repouso, própria casa) Em que bairro nós estamos ou qual o nome de uma rua próxima? Em que cidade nós estamos? Em que estado nós estamos?</p>		5
<p>MEMÓRIA IMEDIATA: ()</p> <p>Eu vou dizer 3 palavras e você irá repeti-las a seguir: Carro, vaso, tijolo.</p>		3

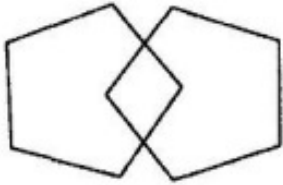
No Quadro 30 está a segunda parte do MEEM.

Quadro 30 - Mini exame do estado mental (parte 2)

<p>ATENÇÃO E CÁLCULO: () Subtração de setes seriadamente: 100-7=93-7=86-7=79-7=72- 7=65</p>	5
<p>MEMÓRIA RECENTE: () Quais as 3 palavras que você repetiu antes?</p>	3
<p>LINGUAGEM: () Mostrar um relógio de pulso e uma caneta e pedir os nomes.</p>	2
<p>Preste atenção: vou lhe dizer () uma frase e quero que você repita depois de mim: Nem aqui, nem ali, nem lá.</p>	1
<p>Comando: pegue este papel () com a mão direita (1 ponto), dobre-o ao meio (1 pontos), coloque-o no chão (1 ponto)</p>	3
<p>Leia e obedeça: “feche os olhos!”()</p>	1
<p>Peça ao indivíduo para escrever () uma frase. Se não compreender o significado ajude com: alguma frase que tenha começo, meio e fim; alguma coisa que aconteceu hoje; alguma coisa que queira dizer</p>	1

No Quadro 31 está a parte final do MEEM.

Quadro 31 - Mini exame do estado mental (parte 3)

<p>Cópia do desenho: mostre o () modelo e peça para fazer o melhor possível (estará correto se existirem 10 ângulos, dos quais dois devem estar interseccionados).</p> 	<p>1</p>
<p>TOTAL DE PONTOS ()</p>	<p>30</p>

Qualquer pontuação **igual ou superior a 25** (de um total de 30) é **efetivamente normal** (íntacto). Abaixo disso, a pontuação pode indicar **perda cognitiva grave (≤ 9 pontos)**, **moderada (10 a 20 pontos)** ou **leve (21 a 24 pontos)**. A pontuação bruta pode precisar ser corrigida de acordo com a escolaridade e idade. Pontuações baixas ou muito baixas são fortemente correlacionadas com demência, embora outros distúrbios mentais podem também levar a resultados anormais no teste MEEM. A presença de problemas puramente físicos pode também interferir com a interpretação se não levados em consideração apropriadamente; por exemplo, um paciente pode não ser capaz de ouvir ou ler instruções adequadamente ou pode possuir um déficit motor que afete a habilidade de escrever ou desenhar.

APÊNDICE D – Questionário

No Quadro 32 encontra-se a primeira parte do questionário submetido aos pacientes após a sessão de exercícios.

Quadro 32 – Questionário (parte 1)

<p><u>Questionário</u></p> <p>Gênero: (M)- Homem (F) - Mulher</p> <p>Idade:</p> <p>Data da entrevista:</p> <p>1- Você gosta das sessões de Fisioterapia convencional? Marque um X nas respostas correspondentes. () sim () não () às vezes – Descreva.</p> <hr/> <p>2- Há quanto tempo você recebe tratamento fisioterapêutico?</p> <hr/> <p>3- Você já havia recebido algum tipo de tratamento fisioterapêutico que utilizasse algum recurso diferente? (videogame, computador)? () Sim () Não</p> <p>4- Se sim na resposta anterior, você ainda realiza esse tratamento? () sim () não () às vezes</p> <p>5- Você acredita ter sido fácil compreender a proposta das atividades do Kinect? () Sim () Não</p> <p>6- Você acredita que a realização das atividades propostas na sessão de fisioterapia se tornaram mais fáceis com o uso do Kinect? () Sim () Não</p> <p>7- Existe alguma atividade proposta que você considerou de mais difícil compreensão? Se sim, diga qual.</p> <hr/>

No Quadro 33 está a segunda parte do questionário.

Quadro 33 – Questionário (parte 2)

8- Existe alguma atividade proposta que você acredita ter sido mais difícil de executar? Se sim, diga qual.

9- Você acredita que as atividades realizadas com o auxílio do Kinect sejam próximas de atividades que você realiza no seu dia-a-dia? (comer, alcançar algum objeto, sentar, caminhar).

() Sim () Não

10- Se você tivesse oportunidade, gostaria de participar de algum tratamento em Reabilitação Neurológica que utilize o sensor Kinect como ferramenta?

() Sim () Não