

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

**MOVEREMUS: UMA APLICAÇÃO DE AUXÍLIO A**  
**TRATAMENTOS FISIOTERAPÊUTICOS UTILIZANDO O**  
**MICROSOFT KINECT**

**ANDRÉ RICARDO GIELOW**

**BLUMENAU**  
**2015**

**2015/2-04**

**ANDRÉ RICARDO GIELOW**

**MOVEREMUS: UMA APLICAÇÃO DE AUXÍLIO A  
TRATAMENTOS FISIOTERAPÊUTICOS UTILIZANDO O  
MICROSOFT KINECT**

Trabalho de Conclusão de Curso apresentado ao curso de graduação em Ciência da Computação do Centro de Ciências Exatas e Naturais da Universidade Regional de Blumenau como requisito parcial para a obtenção do grau de Bacharel em Ciência da Computação.

Prof. Aurélio Faustino Hoppe, Mestre - Orientador

**BLUMENAU  
2015**

**2015/2-04**

**MOVEREMUS: UMA APLICAÇÃO DE AUXÍLIO À  
TRATAMENTOS FISIOTERAPÊUTICOS UTILIZANDO O  
MICROSOFT KINECT**

Por

**ANDRÉ RICARDO GIELOW**

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

Presidente: \_\_\_\_\_  
Prof. Aurélio Faustino Hoppe, Mestre – Orientador, FURB

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

Membro: \_\_\_\_\_  
Profa. Joyce Martins, Mestre – FURB

Blumenau, 09 de dezembro de 2015

Dedico este trabalho à minha família, que sempre me incentivou incansavelmente à estudar e me aprimorar cada vez mais.

## **AGRADECIMENTOS**

À minha família pela educação que me deram, que fizeram-me a pessoa que sou, e pelo constante incentivo à educação.

Aos amigos, pela compreensão de minha ausência durante o desenvolvimento deste trabalho, dicas e conselhos.

Aos colegas da faculdade e do trabalho, pelos auxílios prestados, pelo compartilhamento de conhecimento e pela passagem de experiência.

À minha sogra por me ajudar no experimento da aplicação e ao sogro por ceder a sala e seu tempo do futebol para realização deste.

À minha namorada pelo incentivo diário para realização deste trabalho e por todo apoio prestado.

Ao meu orientador Aurélio Hoppe, pela ideia e apoio na escolha do tema que levou ao desenvolvimento deste trabalho. Pelas assistências, empenho e transmissão de conhecimento mesmo em meio a todas as dificuldades e mar de caos que estive no decorrer deste trabalho.

Cada um que passa em nossa vida, passa sozinho, mas não vai só, nem nos deixa sós; leva um pouco de nós mesmos, deixa um pouco de si mesmo.

Antoine de Saint-Exupéry

## **RESUMO**

Este trabalho apresenta o desenvolvimento de uma aplicação de auxílio à tratamentos fisioterapêuticos utilizando o sensor Microsoft Kinect. Esta aplicação permite ao fisioterapeuta criar exercícios específicos para cada caso clínico. O paciente ao executar um exercício cadastrado, visualiza na aplicação a posição que deverá repetir. A aplicação exibe em tempo real o esqueleto do paciente, articulações coloridas para indicar o percentual de acerto em cada uma delas e as orientações de como a posição deve ser executada. Após a execução do exercício pelo paciente, ambos, fisioterapeuta e paciente podem visualizar o desempenho alcançado na repetição. A aplicação foi validada com um fisioterapeuta e sua usabilidade testada com usuários comuns. A aplicação obteve um percentual de satisfação de 87%, sendo que 75% dos usuários conseguiram executar os exercícios propostos.

Palavras-chave: Fisioterapia. Reabilitação. Kinect.

## **ABSTRACT**

This work presents the development of an application to aid the physiotherapy treatment using the sensor Microsoft Kinect. This application allows the physiotherapist to create specific exercises for each clinic case. The patient, while performing the recorded exercise, will visualize in the application a position that him should repeat. The application shows in real time the skeleton of the patient, and in the marked joints a color according with de hit percentage of each one, the application also display guidance on how the position should be executed. After the patient's exercise execution, both physiotherapist and patient could visualize the performance achieved in the repetition. The application was validated with a physiotherapist and your usability was tested with common users. The application has obtained a satisfaction percentage of 87%, being that 75% of the users succeed in the execution of the proposed exercises.

Key-words: Physiotherapy. Rehabilitation. Kinect.

## LISTA DE FIGURAS

Figura 1 - Componentes Microsoft Kinect .....	14
Figura 2 - Sistema de coordenadas do Kinect .....	15
Figura 3 - Juntas reconhecidas pelo Kinect .....	16
Figura 4 - Simulador Sensorama .....	17
Figura 5 - Cubo desenhado pelo Three.JS .....	20
Figura 6 - Cubo Three.JS com material e iluminação .....	20
Figura 7 - Execução do exercício .....	22
Figura 8 - Software Jintronix.....	23
Figura 9 - Execução de exercício para reabilitação motora de membros inferiores.....	24
Figura 10 - Diagrama de casos de uso .....	27
Figura 11 - Diagrama de classes.....	28
Figura 12 - Fluxo geral de comunicação entre camadas .....	30
Figura 13 - Esqueleto desenhado com informações do Kinect .....	40
Figura 14 - Degradê de cores utilizadas na avaliação das juntas.....	40
Figura 15 - Aplicação do degradê de validação .....	41
Figura 16 - Cálculo de graus em relação à origem .....	44
Figura 17 - Fluxo de atividades da aplicação .....	46
Figura 18 - Cadastrar/editar exercício .....	47
Figura 19 - Gravação da posição do exercício a ser repetida pelo usuário .....	48
Figura 20 - Posição gravada com sucesso .....	48
Figura 21 - Gravação de posições concluída .....	49
Figura 22 - Tela de execução do exercício .....	50
Figura 23 - Próxima posição selecionada .....	51
Figura 24 - Execução concluída .....	52
Figura 25 - Avaliação de desempenho do exercício.....	52
Figura 26 - Posicionamento usuário .....	53
Figura 27 - Esqueleto do usuário representado no ambiente virtual .....	54
Figura 28 - Validação das instruções a serem executadas pelo paciente.....	55
Figura 29 - Percentual de acerto abaixo do esperado .....	56
Figura 30 – Percentual de acerto elevado .....	57
Figura 31 - Usuários de diferentes estaturas.....	58

Figura 32 - Posicionamento dos usuários com estaturas diferentes .....	59
Figura 33 - Diagrama de entidade e relacionamento da base de dados .....	73
Figura 34 - Cadastro de médicos .....	74
Figura 35 - Cadastrar ou editar médico .....	74
Figura 36 - Cadastro de pacientes.....	75
Figura 37 - Cadastrar ou editar paciente.....	75
Figura 38 - Cadastro de exercícios .....	76

## LISTA DE QUADROS

Quadro 1 - Principais objetos do Three.JS .....	19
Quadro 2 - Criando um cubo em Three.JS .....	19
Quadro 3 - Código Three.JS para adicionar iluminação e alterar material .....	20
Quadro 4 - Comparação entre trabalhos correlatos .....	24
Quadro 5 - Método de conexão com o sensor Kinect.....	31
Quadro 6 - Recuperando dados dos esqueletos .....	32
Quadro 7 - Inicialização da API de reconhecimento de voz .....	33
Quadro 8 - Recuperando comandos de voz reconhecidos.....	33
Quadro 9 - Configuração do vocabulário da gramática para reconhecimento de voz.....	34
Quadro 10 - Classe derivada SignalR.....	35
Quadro 11 - Conexão com a camada de negócio utilizando SignalR .....	35
Quadro 12 - Inicialização do objeto câmera.....	36
Quadro 13 - Criação da forma geométrica para representar o chão do cenário .....	36
Quadro 14 - Inicialização da iluminação direcional e ambiente da cena .....	37
Quadro 15 - Adição dos objetos na cena .....	37
Quadro 16 - Renderização da cena e adição à página .....	37
Quadro 17 - Atualização do esqueleto no cenário .....	38
Quadro 18 - Montar cor da junta .....	38
Quadro 19 - Desenhar ligações das juntas.....	39
Quadro 20 - Métodos auxiliares para ligar esferas ao cenário .....	39
Quadro 21 - Informações do fisioterapeuta e paciente .....	42
Quadro 22 - Cálculo para validar posicionamento .....	43
Quadro 23 - Cálculo de direção horizontal e vertical.....	45
Quadro 24 - Cálculo da direção diagonal .....	45
Quadro 25 - Respostas quanto à lista de tarefas .....	61
Quadro 26 - Respostas do fisioterapeuta em relação ao questionário de usabilidade .....	61
Quadro 27 - Perfis dos usuários envolvidos nos testes de usabilidade.....	63
Quadro 28 - Respostas dos usuários em relação ao questionário de usabilidade .....	64
Quadro 29 - UC01 Cadastrar médicos .....	70
Quadro 30 - UC02 Cadastrar pacientes .....	70
Quadro 31 - UC03 Manter exercícios .....	71

Quadro 32 - UC04 Gravar exercícios .....	71
Quadro 33 - UC05 Visualizar progresso.....	72
Quadro 34 - UC06 Executar exercícios .....	72
Quadro 35 - Questionário de perfil de usuário .....	78
Quadro 36 - Questionário de tarefas a serem realizadas pelo fisioterapeuta.....	78
Quadro 37 - Questionário de usabilidade aplicado no usuário fisioterapeuta .....	79
Quadro 38 - Questionário de usabilidade aplicado ao usuário paciente.....	81

## **LISTA DE ABREVIATURAS E SIGLAS**

API – Application Programming Interface

CMOS – Complimentary Metal-Oxide Semiconductor

E3 – Eletronic Entertainment Expo

HTML – Hyper Text Markup Language

JSON – Javascript Object Notation

RGB – Red Green Blue

RV – Realidade Virtual

SDK – Software Development Kit

UML – Unified Modeling Language

# SUMÁRIO

<b>1 INTRODUÇÃO.....</b>	<b>10</b>
1.1 OBJETIVOS.....	11
1.2 ESTRUTURA.....	11
<b>2 FUNDAMENTAÇÃO TEÓRICA .....</b>	<b>12</b>
2.1 REABILITAÇÃO FISIOTERAPÊUTICA .....	12
2.2 MICROSOFT KINECT.....	14
2.3 REALIDADE VIRTUAL.....	17
2.3.1 THREE.JS .....	18
2.4 TRABALHOS CORRELATOS .....	21
2.4.1 Ferramenta de auxílio à tratamentos fisioterapêuticos utilizando o Kinect .....	21
2.4.2 JINTRONIX: Melhorando a reabilitação física através de tecnologias de captura de movimento. ....	22
2.4.3 Sistema para fisioterapia baseado na plataforma Kinect.....	23
2.4.4 Comparação com trabalhos correlatos e discussões.....	24
<b>3 DESENVOLVIMENTO .....</b>	<b>26</b>
3.1 REQUISITOS.....	26
3.2 ESPECIFICAÇÃO .....	27
3.2.1 DIAGRAMA DE CASOS DE USO .....	27
3.2.2 DIAGRAMA DE CLASSES .....	28
3.3 IMPLEMENTAÇÃO .....	29
3.3.1 Técnicas e ferramentas utilizadas.....	29
3.3.2 Fluxo geral de comunicação entre camadas.....	30
3.3.3 Implementação da aplicação .....	31
3.3.4 Representação virtual do esqueleto na aplicação .....	36
3.3.5 Validação e instruções.....	41
3.3.6 Operacionalidade da implementação .....	46
3.4 RESULTADOS E DISCUSSÕES.....	53
3.4.1 Experimento 01: representação do esqueleto no ambiente virtual.....	53
3.4.2 Experimento 02: validação das instruções dadas ao paciente .....	54
3.4.3 Experimento 03: validação do percentual de acerto .....	56
3.4.4 Experimento 04: validação do percentual de acerto entre diferentes estaturas.....	57

3.4.5 Experimento 05: usabilidade da aplicação .....	60
<b>4 CONCLUSÕES .....</b>	<b>66</b>
4.1 EXTENSÕES .....	67
<b>APÊNDICE A – DETALHAMENTO DOS CASOS DE USO .....</b>	<b>70</b>
<b>APÊNDICE B – MODELO DE ENTIDADE RELACIONAMENTO .....</b>	<b>73</b>
<b>APÊNDICE C – CADASTROS INICIAIS DA APLICAÇÃO.....</b>	<b>74</b>
<b>APÊNDICE D – ROTEIRO E QUESTIONÁRIO DE PERFIL DE USUÁRIO E USABILIDADE.....</b>	<b>77</b>

## 1 INTRODUÇÃO

Cada vez mais a relação entre o mundo virtual e o mundo real estreita-se. A interação entre humanos e máquinas torna-se mais amigável e natural, seja ela dada por gestos ou comandos de voz por exemplo (VALLI, 2004). Através do Kinect, a Microsoft promoveu o objetivo de trazer experiências envolventes baseada em gestos, com o lema “Você é o controle” afirma Lutz (2014). De acordo com Lutz (2014), posteriormente com o lançamento do Kinect for Windows, o Kinect passou a ser vastamente utilizado em outras áreas de aplicação, como na área da saúde em assistente de cirurgia, na área militar como detector de intrusos e até mesmo na área espacial, controlando um braço robótico por gestos. O sensor Microsoft Kinect foi apresentado como um controle para jogos juntamente com o console Xbox 360. Isto permitiu a criação de jogos que interagissem mais com o jogador, onde o jogo passou a responder aos gestos e movimentos feitos pelo jogador no mundo real, incorporando-os ao ambiente virtual. Isto aumentou e muito, a sensação de participação no jogo por parte do jogador.

Segundo Araújo e Kirner (1996), jogos são atividades que envolvem diversão e interação com objetivo e desafios motivadores. Sua principal característica é a existência de regras pré-estabelecidas. Estes elementos característicos de qualquer jogo propiciam a aprendizagem, criatividade e rapidez das respostas.

Devido a modernidades advindas de avanços tecnológicos, uma possibilidade atual para propiciar jogos à população é a utilização de jogos virtuais, que podem ser definidos como uma atividade lúdica com as mesmas características dos jogos convencionais, mas praticados em ambiente virtual. Nesses ambientes as regras prevalecem e há interação contínua com o jogador, gerando ações e reações até que o objetivo seja atingido. Uma categoria que tem chamado a atenção de profissionais de diversas áreas são os *serious games*, cujo objetivo vai além do entretenimento, visando a transmitir algum conhecimento ao jogador (MACHADO et al., 2011).

Além da utilização de *serious games* no dia a dia de diferentes populações, verifica-se uma procura por profissionais da área de reabilitação na utilização dos avanços tecnológicos em intervenções clínicas. Reabilitação é um processo orientado à recuperação física e psicológica do indivíduo, que previne e trata os distúrbios gerados por alterações genéticas, traumas ou por outras doenças adquiridas ao longo da vida (COFFITO, 2012). Muitas vezes a utilização de técnicas tradicionais de reabilitação desestimula o paciente a executar os exercícios dificultando a recuperação sensório-motora. No processo de reabilitação, a execução de exercícios que estimulem gradativamente determinado movimento é imprescindível para a

recuperação do paciente. Tal prática auxilia a aperfeiçoar os aspectos morfológicos, fisiológicos e psicológicos do indivíduo. Entretanto, tais atividades necessitam ser adaptadas à condição de cada indivíduo (MONTEIRO, 2011).

Considerando tais premissas, o emprego de ambientes virtuais e, em especial, jogos virtuais, pode ser uma alternativa viável, motivadora e de baixo custo para tratamentos de reabilitação fisioterapêutica. Esta ideia já foi implementada e validada por Pilon (2013). No entanto, averiguou-se a necessidade de aprimoramento da interface gráfica, avaliação dos movimentos executados utilizando a tecnologia WebGL e o sensor Kinect, armazenamento de dados das sessões fisioterapêuticas, além da disponibilização via *web*.

## 1.1 OBJETIVOS

O objetivo deste trabalho é fazer uma reformulação do trabalho de Pilon (2013) tornando-o uma aplicação *web*.

Os objetivos específicos do trabalho são:

- a) disponibilizar um mecanismo para que o fisioterapeuta possa cadastrar exercícios específicos utilizando Microsoft Kinect;
- b) disponibilizar um mecanismo que valide, de forma visual, se os exercícios foram realizados corretamente pelos pacientes;
- c) orientar o paciente na execução dos exercícios;
- d) armazenar os dados gerados nas sessões fisioterapêuticas para que o médico e paciente possam acompanhar o progresso do tratamento.

## 1.2 ESTRUTURA

Este trabalho está dividido em quatro capítulos, onde no primeiro capítulo é dada a introdução do trabalho e seus objetivos. O segundo capítulo trata de apresentar a fundamentação teórica, onde aborda-se reabilitação fisioterapêutica, o sensor Microsoft Kinect e por último, realidade virtual. No terceiro capítulo é demonstrada a especificação e requisitos da aplicação, assim como o seu desenvolvimento e bibliotecas utilizadas, posteriormente é demonstrada a operacionalidade da aplicação. No quarto e último capítulo, são relatadas as conclusões e resultados obtidos e suas possíveis extensões.

## 2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo está organizado em quatro seções. A seção 2.1 aborda a reabilitação fisioterapêutica, seus conceitos e como esta é aplicada. Na seção 2.2 é apresentado o sensor Microsoft Kinect, sua história e funcionamento. A seção 2.3 apresenta o conceito de realidade virtual, sua história e como é composta. Finalmente, na seção 2.4 são apresentados os trabalhos correlatos.

### 2.1 REABILITAÇÃO FISIOTERAPÊUTICA

Conforme Partridge (2006), reabilitação fisioterapêutica é um processo multiprofissional, no qual tem-se o objetivo de restaurar movimentos e funções motoras comprometidas, seja esta incapacidade resultante de um trauma físico ou mental, ou em decorrência de alguma doença. Com a restauração das funções motoras, seja ela completa ou parcial, o paciente poderá ser reinserido na sociedade. Porém, segundo Teixeira (2013), não pode-se afirmar que o tratamento foi um sucesso caso o paciente não consiga desempenhar suas funções sociais de origem, igualmente ou similar antes do trauma ou doença.

Segundo definição do Conselho Federal de Fisioterapia e Terapia Ocupacional (COFFITO) o fisioterapeuta é um “profissional de saúde, com formação acadêmica superior, habilitado à construção dos distúrbios cinéticos funcionais (Diagnóstico Fisioterapêutico), a prescrição das condutas fisioterapêuticas, a sua ordenação e indução no paciente bem como, o acompanhamento da evolução do quadro clínico funcional e as condições para alta do serviço”, resolução do COFFITO, Lei 8.856/94, 1 de março de 1994.

Os exercícios fisioterapêuticos estão subdivididos em dois grupos: movimento ativo e movimento passivo. O movimento ativo é executado ou controlado pela ação voluntária dos músculos, sendo exercitado em oposição a uma força externa. Já os movimentos passivos, são executados por uma força externa durante a inatividade, permitindo o movimento.

Os exercícios livres, pertencentes à categoria dos movimentos ativos, são os mais utilizados pelos fisioterapeutas, já que não causam maiores sofrimentos ao paciente e são de fácil execução. Nos exercícios livres, os exercícios são executados apenas com o próprio esforço muscular do paciente, sem nenhum tipo de auxílio. O propósito deste tipo de exercício é treinar e melhorar a coordenação e aumentar a confiança em executar e controlar os movimentos.

Conforme Gardiner (1995), a grande vantagem dos exercícios livres é que, uma vez que o paciente domina a técnica de sua execução e tem conhecimento de seu propósito, pode-se praticá-lo a qualquer momento e em qualquer lugar. De acordo com a extensão da área afetada,

os movimentos podem ser classificados como localizados e globais. Os exercícios localizados dedicam-se a produzir efeitos locais e são bem específicos, enquanto nos exercícios globais, vários grupos de articulações e músculos são envolvidos.

Para realização dos exercícios livres, faz-se necessário alguns cuidados com o paciente e a prática dos exercícios (GARDINER, 1995):

- a) a posição inicial é escolhida e ensinada com cuidado, para assegurar a máxima eficiência postural com base para o movimento;
- b) a instrução é dada de maneira que desperte o interesse e cooperação do paciente e o leve a entender o padrão quanto o propósito do exercício;
- c) a velocidade na qual o exercício é feito depende do efeito exigido. É geralmente lenta durante o período de aprendizado e mais tarde, quando o paciente estiver familiarizado pode-se executar com o seu próprio ritmo natural ou com a velocidade que é recomendada pelo fisioterapeuta. O paciente é frequentemente ajudado a manter seu ritmo natural em casa, durante a execução dos exercícios, é incentivado a contar em voz alta;
- d) a duração do exercício depende da capacidade do paciente. Geralmente três vezes para cada exercício, com curtos períodos de descanso, ou com atividades intercaladas, assegurando a prática sem fadiga excessiva.

Tratamentos de reabilitação terapêutica, são importantes onde pacientes possuem determinada deficiência física ou motora, em decorrência de traumas físicos ou de doenças. O tipo de tratamento pode variar de acordo com o problema do paciente, sendo necessário a aplicação de diferentes técnicas a fim de reintegrá-lo a sua função na sociedade assim como era antes do seu trauma.

Ainda nos dias atuais, a maioria das clínicas fazem o controle dos tratamentos de forma manual através de anotações no prontuário do paciente, sendo que, 90% dos profissionais não registram os procedimentos executados e/ou quaisquer anotações sobre o progresso e resposta do paciente (MELANI; SILVA, 2006).

Com o surgimento de novas tecnologias capazes de detectar movimentos do mundo real a um baixo custo, tornou-se possível o desenvolvimento de aplicações para diversas áreas, inclusive a área da fisioterapia (CHANG; CHEN; HUANG, 2011). A aplicação de tecnologias de captura de movimento tem sido abordada e discutida nos principais portais, onde busca-se a melhor forma de aplicar e desenvolver jogos e aplicações para auxiliar tratamentos fisioterapêuticos regulares (SILVA et al., 2012). Segundo os mesmos autores, os aplicativos

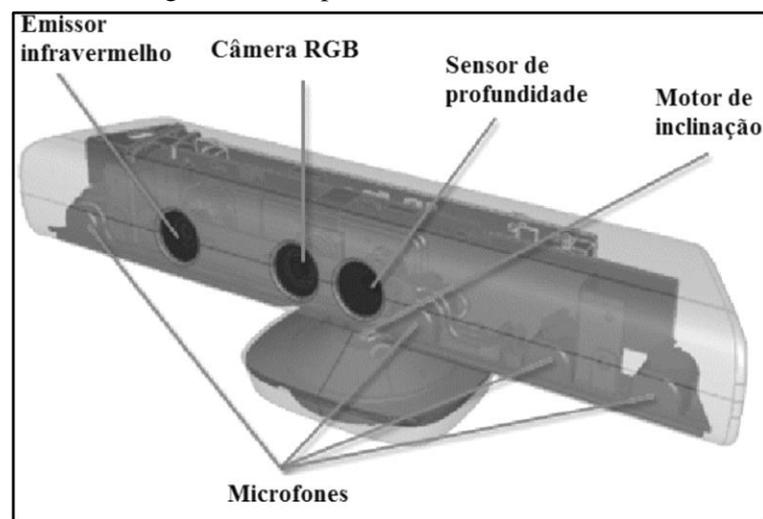
desenvolvidos devem gerar dados confiáveis sobre o movimento realizado, assim como garantir a execução dos movimentos relevantes condizentes à condição do paciente, adicionando também certo grau de ludicidade e dinamismo ao tratamento, o qual deve ser adequado às limitações do paciente.

## 2.2 MICROSOFT KINECT

O Microsoft Kinect é um sensor de movimentos produzido pela Microsoft que foi anunciado primeiramente em junho de 2009, na Eletronic Entertainment Expo (E3) e oficialmente lançado na América do Norte em 4 de novembro de 2010 (LOWENSOHN, 2011). A primeira versão do *Software Development Kit* (SDK) foi lançada apenas em junho de 2011 (BIGGS, 2011).

O Kinect é uma combinação sofisticada de câmeras, microfones e software. Inicialmente, foi apresentado como um dispositivo complementar para o Xbox (console de videogame) como um reforço na interação entre jogo e jogador. Posteriormente, a Microsoft anunciou o Kinect for Windows, inspirando assim empresas e desenvolvedores ao redor do mundo a criarem aplicações em diferentes nichos, como saúde, militar e espacial (LUTZ, 2014). Na Figura 1 é mostrada a localização dos componentes do Kinect.

Figura 1 - Componentes Microsoft Kinect



Fonte: Crawford (2010).

O Kinect é composto dos seguintes componentes (CRAWFORD, 2010):

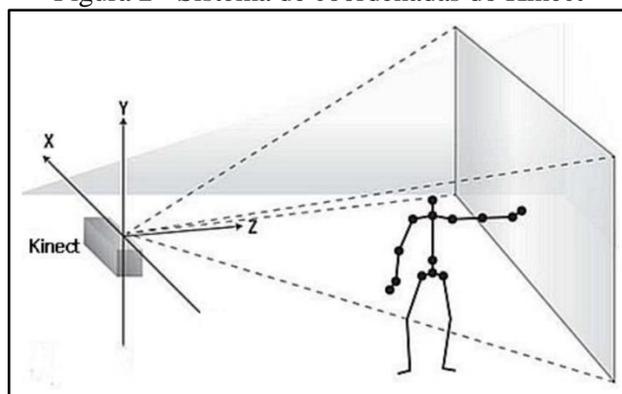
- câmera RGB (Red Green Blue): câmera utilizada no reconhecimento facial e em outros recursos de detecção;
- microfones: quatro microfones distribuídos isolam as vozes dos jogadores do ruído do ambiente, permitindo ao jogador ficar distante do sensor e ainda assim utilizar comandos de voz;

- c) emissor infravermelho: emite raios infravermelhos para serem posteriormente capturados pelo sensor de profundidade;
- d) sensor de profundidade: um projetor infravermelho e um sensor monocromático CMOS (*Complimentary metal-oxide semiconductor*) trabalham em conjunto para identificar o ambiente em três dimensões, independente das condições de iluminação.

De acordo com Khoshelhman e Elberink (2012), os inventores do Kinect descrevem a mensuração da profundidade com um processo de triangulação. O emissor de raios infravermelhos emite um único feixe que é dividido em múltiplos feixes por uma rede de difração, para criar um padrão constante de pontos projetados na cena. Este padrão é capturado pela câmera infravermelha, e está correlacionado contra o padrão de referência. O padrão de referência é obtido pela captura de um plano a uma distância conhecida do sensor, e é armazenado na memória do sensor. Quando um espectro é projetado sobre um objeto, cuja distância para o sensor é menor ou maior que o plano de referência a posição do espectro na imagem infravermelha será deslocada na direção da linha base entre o emissor infravermelho e o centro em perspectiva da câmera infravermelha. Essas mudanças da imagem são medidas para todas manchas da malha de pontos capturada, por um procedimento de correlação de imagem simples, que produz uma imagem de disparidade. Para cada pixel, a distância pode ser obtida a partir da disparidade correspondente.

Com isso, Khoshelhman e Elberink (2012) complementam que, para expressar as coordenadas em três dimensões, é considerado a partir da perspectiva central da sensor de profundidade, um sistema de coordenadas de profundidade. O eixo Z (profundidade) é ortogonal ao plano da imagem, em relação ao objeto. O eixo X, horizontal da esquerda para direita, é perpendicular ao eixo Z. O eixo Y, vertical de baixo, para cima é ortogonal ao eixo X e Z. A Figura 2 mostra o sistema de coordenadas em relação ao Kinect.

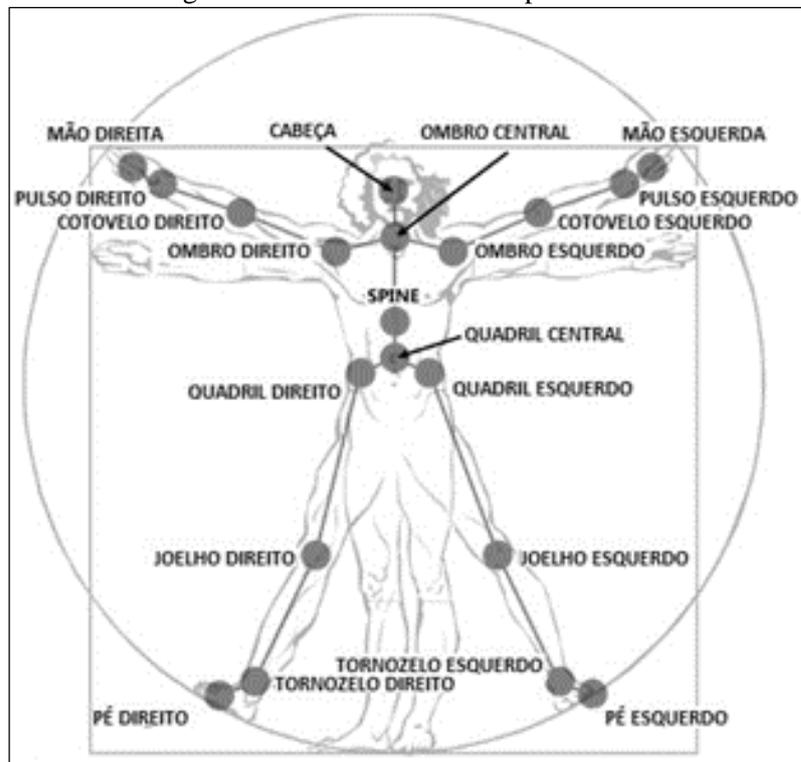
Figura 2 - Sistema de coordenadas do Kinect



Fonte: adaptado de Tang et al. (2014).

A *Application Programming Interface* (API) do Kinect já disponibiliza aos desenvolvedores recursos de processamento de áudio, nuvem de pontos de todo ambiente capturado, esqueletos reconhecidos e outros eventos. O Kinect na primeira e segunda versão pode respectivamente, capturar quatro e seis esqueletos simultaneamente, porém apenas dois destes possuem informações da posição de cada junta, enquanto os demais apenas possuem a informação sobre sua posição (MILES, 2012). A Figura 3 mostra quais juntas são reconhecidas pelo Microsoft Kinect.

Figura 3 - Juntas reconhecidas pelo Kinect



Fonte: adaptado de Sá (2011, p. 18).

Segundo Catuhe (2012), cada junta é definida por uma posição (X, Y e Z) representada no espaço do esqueleto. O espaço do esqueleto é todo o ambiente ao redor do sensor, o qual é a posição de origem estando logicamente na coordenada (0, 0, 0). As coordenadas são representadas em metros, em vez de milímetros. Através da API pode-se obter a lista de esqueletos reconhecidos pelo Kinect. A classe que representa o esqueleto reconhecido possui a lista de todas as juntas e suas respectivas coordenadas (CATUHE, 2012).

O SDK do Kinect permite controlar o motor da base do Kinect para obter um melhor ângulo da cena, identificação de pessoas e até mesmo a localização do som no ambiente. Segundo Miles (2012), com todas essas informações é possível criar sistemas de realidade virtual, representar os usuários dentro do sistema, permitindo a estes interagirem através de gestos e/ou comandos de voz, sentindo-se assim imersos e envolvidos.

### 2.3 REALIDADE VIRTUAL

Segundo Burdea e Coiffet (2003), Realidade Virtual é uma interface de alta tecnologia que envolve simulações e interações em tempo real através de canais multissensoriais. Estas modalidades sensoriais são visão, audição, paladar, olfato e tato. Burdea e Coiffet (2013) ainda afirmam que realidade virtual é interativa e imersiva, e que não é apenas uma interface de média ou alta tecnologia, mas também possui aplicação que envolvem soluções para problemas reais nas áreas de engenharia, medicina e militar por exemplo.

Realidade Virtual não é uma invenção atual. Nos Estados Unidos no ano de 1962, Morton Heilig patenteou sua invenção intitulada de Sensorama Simulator, o qual foi o primeiro fliperama com Realidade Virtual. Este simulador tinha um visor o qual emulava três dimensões através de duas telas lado-a-lado, movimentos (através de cadeira que vibrava), som estéreo, aromas, vento (através de um soprador perto da cabeça do usuário). Utilizando estes mecanismos era possível simular um passeio de motocicleta por exemplo. A Figura 4 mostra a estrutura do simulador.

Figura 4 - Simulador Sensorama



Fonte: Heilig (1995).

Como pode-se ver na Figura 4, o usuário fica sentado na cadeira que vibra e com a cabeça parcialmente inserida dentro da máquina, de forma a privar o usuário o máximo possível de fatores externos e imergi-lo o máximo possível no ambiente criado pelo simulador.

Rodrigues e Porto (2013) afirmam que a Realidade Virtual têm como base três ideias base: imersão, interação e envolvimento, sendo que:

- a) imersão: é propiciar ao usuário a sensação de estar dentro do cenário, fazendo com que este interaja com o cenário virtual através de dispositivos de entrada do mundo real para que possam ser processados e representados no mundo virtual, onde o usuário passa a participar em vez de apenas observar;
- b) interação: é a capacidade de detectar às entradas do mundo real, processar, modificar e apresentá-las em tempo hábil de modo que o usuário tenha a percepção de que isto ocorreu em tempo real;
- c) envolvimento: pode ser ativo onde o usuário participa de um jogo ou passivo onde o usuário assiste a uma cirurgia virtual, por exemplo, sendo que é ligada diretamente ao grau de estimulação para comprometer o usuário a cumprir determinada tarefa.

Atualmente a principal forma de representar o mundo real de maneira virtual é utilizar-se de tecnologias que permitem a criação de cenários em três dimensões. Inicialmente estas tecnologias de criação de ambientes em três dimensões tinham restrições e rodavam somente em computadores e em aplicações *desktop*. A tecnologia mais utilizada para criar cenas 3D é o OpenGL, a qual foi lançada em 1992 e atualmente é difundida como um padrão da computação gráfica. Conforme a evolução de hardware e software e a grande difusão da *internet* para grande maioria dos usuários, o OpenGL tornou-se a tecnologia padrão para desenvolvimento de aplicações gráficas, porém para aplicações *web* este foi liberado como WebGL. Desenvolver uma aplicação gráfica utilizando somente o WebGL nativo, sem o auxílio de nenhum tipo de biblioteca é uma tarefa árdua. Entretanto, há algumas bibliotecas que auxiliam no desenvolvimento. ThreeJS é uma destas, a qual é bastante difundida e de código aberto, a qual recebe contribuições de vários usuários da comunidade de software é o ThreeJS, com esta biblioteca o desenvolvimento de aplicações gráficas torna-se mais fácil e produtivo.

### 2.3.1 THREEJS

ThreeJS é uma biblioteca que abstrai e facilita o uso da tecnologia WebGL. Foi criada por Ricardo Cabello e está disponível no GitHub desde 2010. Desde então, recebeu muitas contribuições e sua base de usuários vem crescendo cada dia mais (DANCHILLA, 2012). WebGL é atualmente o novo padrão para gráficos 3D na *web*. Com WebGL, desenvolvedores podem aproveitar todo poder de renderização gráfica de um computador, manipulando-o exclusivamente através de um conjunto de instruções JavaScript. O acesso ao WebGL é feito utilizando o elemento `Canvas` do HTML, obtendo um contexto especial para o WebGL.

Segundo Dirksen (2015), entretanto a API WebGL não é muito fácil de ser utilizada, pois requer dos desenvolvedores um conhecimento profundo de como esta API funciona

internamente. Dirksen (2015) afirma ainda que com Three.JS têm-se uma API de fácil utilização e toda gama de materiais e formas que se tem utilizando diretamente a API do WebGL. No Quadro 1, estão os principais objetos do Three.JS (DANCHILLA, 2012).

Quadro 1 - Principais objetos do Three.JS

Objeto	Descrição
THREE.Renderer	O objeto responsável por renderizar a cena. Sua implementação pode ser CanvasRenderer, DOMRenderer, SVGRenderer ou WebGLRenderer
THREE.Scene	Armazena os objetos e iluminações contidos na cena
THREE.Camera	Câmera virtual, podendo ser de perspectiva ou ortográfica
THREE.Object3D	Multiplos objetos, incluindo Mesh, Line, Particle, Bone e Sprite
THREE.Light	Modelo de luz, podendo ser ambiente, direcional, local ou de ponto

Fonte: Danchilla (2012, p. 175).

Utilizando os principais objetos listados no Quadro 1 e objetos derivados, pode-se rapidamente montar uma cena simples. No Quadro 2 é possível visualizar um exemplo do código necessário para desenhar um cubo na cor azul.

Quadro 2 - Criando um cubo em Three.JS

```

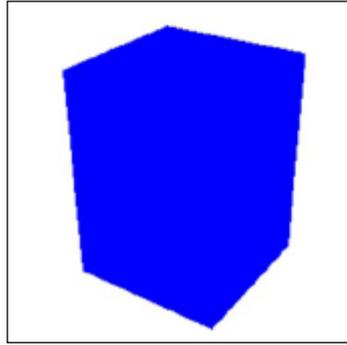
01 <html>
02 <script>
03     var renderer = null, scene = null, camera = null;
04     function initWebGL(){
05         setupRenderer();
06         setupScene();
07         setupCamera();
08         renderer.render(scene, camera);
09     }
10     function setupRenderer(){
11         renderer = new THREE.WebGLRenderer();
12         renderer.setSize( 400, 300 );
13         // Adicionar o elemento canvas gerado à página
14         document.body.appendChild( renderer.domElement );
15     }
16     function setupScene(){
17         scene = new THREE.Scene();
18         var cube = new THREE.Mesh(
19             new THREE.CubeGeometry( 5, 7, 5 ),
20             new THREE.MeshBasicMaterial( { color:0x0000FF } ) );
21         scene.add(cube);
22     }
23     function setupCamera(){
24         camera = new THREE.PerspectiveCamera(35, 400, 300, .1,
25         10000);
26         camera.position.set( -15, 10, 10 );
27         camera.lookAt( scene.position );
28         scene.add( camera );
29     }
30 }
31 </script>
32 <body onload='initWebGL()'></body>
33 </html>

```

Fonte: adaptado de Danchilla (2012).

O desenho gerado pelo código do Quadro 2 é exibido na Figura 5.

Figura 5 - Cubo desenhado pelo Three.JS



Fonte: Danchilla (2012).

A partir da Figura 5 fica difícil visualizar a forma geométrica cubo, pois a textura utilizada é sólida e nenhuma iluminação foi adicionada à cena. Para melhorar a visualização 3D da forma geométrica basta alterar o tipo de textura e adicionar uma iluminação direcional à cena, conforme código do Quadro 3.

Quadro 3 - Código Three.JS para adicionar iluminação e alterar material

```

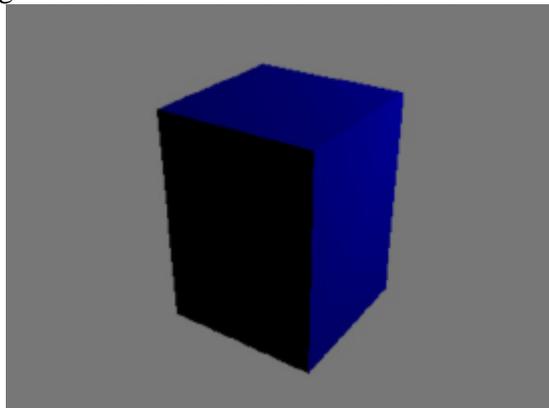
01 function addMesh() {
02     var cube = new THREE.Mesh(
03         new THREE.CubeGeometry( 5, 7, 5),
04         new THREE.MeshBasicMaterial({ color: 0x0000FF })
05     );
06     scene.add(cube);
07 }
08 function addLight() {
09     var light = new THREE.PointLight( 0xFFFFFF );
10     light.position.set( 20, 20, 20);
11     scene.add(light);
12 }

```

Fonte: adaptado de Danchilla (2012).

A Figura 6 mostra o resultado da alteração feita no código do Quadro 2.

Figura 6 - Cubo Three.JS com material e iluminação



Fonte: Danchilla (2012).

Comparando o cubo da Figura 5 com o cubo da Figura 6, pode-se perceber que em poucas linhas de código, pode-se alterar drasticamente o resultado final de uma cena. O *framework* Three.JS faz uma grande abstração às chamadas de baixo nível da API do WebGL,

e a combinação destas é capaz de fornecer ótimos resultados em um rápido intervalo de tempo visto que simplifica e agiliza o desenvolvimento gráfico (DANCHILLA, 2012).

## 2.4 TRABALHOS CORRELATOS

A seguir são apresentados dois trabalhos e uma aplicação comercial que utilizam o sensor Microsoft Kinect para auxiliar em tratamentos de reabilitação fisioterapêutica. O primeiro trabalho é o de Pilon (2013) e o segundo é a aplicação comercial Jintronix (JINTRONIX, 2015). Por último é apresentado o trabalho de Campos (2013).

### 2.4.1 Ferramenta de auxílio à tratamentos fisioterapêuticos utilizando o Kinect

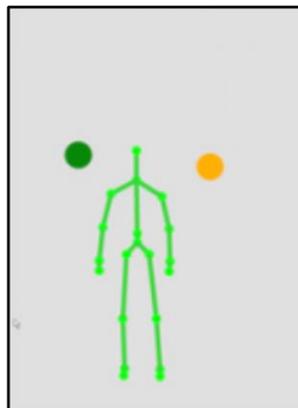
Segundo Pilon (2013), o trabalho tinha dois objetivos, o primeiro era o desenvolvimento de uma ferramenta de auxílio a tratamentos de reabilitação fisioterapêutica utilizando o sensor Kinect, de forma que motivasse os pacientes a executar os exercícios elaborados pelos fisioterapeutas. Como segundo objetivo, desejou-se avaliar a viabilidade da ferramenta com sensor Kinect em tratamentos de reabilitação fisioterapêutica e foram levados em consideração os seguintes critérios: satisfação dos pacientes, precisão da aplicação, facilidade de uso e nível de contribuição ao tratamento.

A ferramenta permite aos usuários utilizar comandos de voz em inglês para interagir com a ferramenta, isto facilita muito a usabilidade da ferramenta durante o cadastramento ou execução dos exercícios. A ferramenta também possui uma funcionalidade de calibragem, a qual é utilizada para que pessoas de diferentes tamanhos fiquem com a mesma proporção na tela de cadastro ou execução do exercício. A calibragem deverá ser feita para todas pessoas que forem utilizar a ferramenta, seja para cadastrar ou executar um exercício.

Para cadastrar um novo exercício o fisioterapeuta deverá inserir o comando “*NEW*”, informar o número de peças e então a quantidade de estágios. A ferramenta então irá para o estágio onde devem ser informadas as peças (juntas do esqueleto) que irão ser rastreadas no exercício, a cada junta informada o fisioterapeuta deverá inserir o comando “*FINISH*” para terminar ou informar a próxima junta. Após informadas todas juntas, a ferramenta vai para o estágio de informar estágio por estágio as posições das peças. A aplicação automaticamente indica visualmente na tela a junta atual, o fisioterapeuta deve então inserir o comando “*HERE*” e então “*FINISH*” para ir para próxima junta. Quanto todos estágios forem informados a aplicação irá solicitar ao fisioterapeuta uma confirmação, caso o usuário opte por salvar o exercício este deverá inserir o comando “*SAVE*”.

Para executar um exercício previamente cadastrado, o paciente ou fisioterapeuta deverá inserir o comando “*LOAD*” e então informar o código do exercício. A ferramenta irá solicitar o estágio que o usuário deseja iniciar, deve-se então informar o número do estágio. Após informado o estágio, a aplicação irá carregar o exercício e juntamente com o esqueleto do usuário, inserir os pontos de referência do exercício. Para alternar entre estágios, o usuário deverá inserir o comando “*ALTER STEP*” e informar o número do estágio que deseja carregar. A Figura 7 abaixo mostra a tela da ferramenta durante a execução de um exercício, com os pontos de referência para auxiliar o paciente de como o exercício deve ser executado.

Figura 7 - Execução do exercício



Fonte: Pilon (2013, p. 51).

Os pontos de referência os quais são inseridos na tela, não possuem nenhum tipo de ligação com a junta e não são validados de nenhuma forma. Nenhum tipo de informação é gravado durante a execução dos exercícios.

A ferramenta foi testada com seis paciente adultos, de ambos sexos que tinham algum tipo de disfunção motora e, ao final das sessões, foi aplicado um questionário a cada paciente. Avaliados os resultados da pesquisa, os exercícios e uso do Kinect foram considerados fáceis pela maioria dos pacientes que também afirmaram que gostariam de usar o Kinect em futuros tratamentos.

#### 2.4.2 JINTRONIX: Melhorando a reabilitação física através de tecnologias de captura de movimento.

Jintronix é uma plataforma virtual fácil de usar, desenvolvida para terapia física e ocupacional. Combina movimentos comuns de reabilitação, jogos e sensores de movimentos para oferecer uma ferramenta divertida e efetiva para tratamentos de reabilitação física. Jintronix é uma aplicação que pode ser instalada em qualquer computador pessoal, desde que atenda aos requisitos mínimos especificados. A ferramenta contém todas atividades para que o paciente se envolva para participar de seu tratamento (JINTRONIX, 2015).

Após o fisioterapeuta disponibilizar acesso ao paciente e configurar os exercícios que este deve executar, o paciente então poderá acessar a ferramenta de sua casa ou em qualquer outro lugar. Durante as sessões de tratamento, a ferramenta salva automaticamente dados da execução dos exercícios feitos pelo paciente. A ferramenta possui uma interface gráfica onde o fisioterapeuta pode verificar todos os dados do paciente, como desempenho, assertividade dos exercícios executados e progresso.

O fisioterapeuta tem a sua disposição jogos e exercícios pré-definidos com vários níveis de dificuldade, os quais este pode disponibilizar ao paciente. O fisioterapeuta, após uma análise do quadro atual do paciente e com sua necessidade, verifica quais atividades serão liberadas para o paciente e então monta um cronograma de exercícios com várias escalas de dificuldade. Como a ferramenta captura e armazena os dados das sessões fisioterapêuticas feitas pelo paciente, o fisioterapeuta com base nestas informações, pode ajustar os exercícios do paciente conforme necessidade. A Figura 8 mostra a tela principal do Jintromix.

Figura 8 - Software Jintromix



Fonte: Jintromix (2015).

A proposta do software Jintromix (2015) é ser utilizado como ferramenta de auxílio à tratamentos. Pode ser utilizado em clínicas onde o paciente executa seus exercícios com o acompanhamento presencial de um fisioterapeuta ou ser instalado em um computador pessoal que tenha o sensor Microsoft Kinect instalado possibilitando assim ao paciente executar seus exercícios em casa.

#### 2.4.3 Sistema para fisioterapia baseado na plataforma Kinect

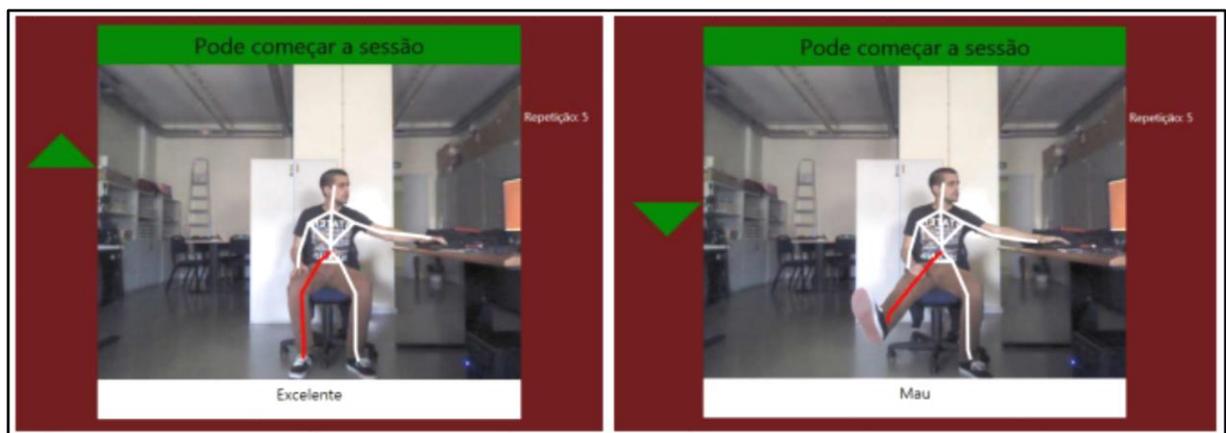
Segundo Campos (2013), seu projeto tinha dois principais objetivos. O primeiro objetivo era desenvolver uma ferramenta de apoio à reabilitação motora de membros inferiores, sendo esta capaz de realizar o acompanhamento de todo o processo de reabilitação do paciente. O

segundo objetivo foi a verificação e validação da ferramenta, onde foi realizado um breve estudo com vinte indivíduos e seus resultados analisados.

A ferramenta disponibiliza uma interface para o fisioterapeuta e outra para o paciente. Na interface do fisioterapeuta, este pode criar, alterar ou visualizar a ficha dos pacientes e também criar sessões de reabilitação. Para criar novas sessões de reabilitação para os pacientes, o fisioterapeuta deve informar o início, o número de sessões e repetições e o tipo de exercício. A ferramenta possui exercícios já cadastrados e não permite ao fisioterapeuta cadastrar exercícios específicos.

O paciente pode visualizar seu histórico e realizar os exercícios de reabilitação. Na execução dos exercícios, a ferramenta faz uma avaliação da qualidade dos movimentos efetuados pelo paciente e os exibe em tempo real através de um gráfico. A Figura 9 mostra o funcionamento da ferramenta de Campos (2013).

Figura 9 - Execução de exercício para reabilitação motora de membros inferiores



Fonte: Campos (2013, p. 48).

Os resultados de cada sessão são devidamente registrados, para que possam ser extraídos e analisados. Também pode-se optar por gravar em vídeo a execução dos exercícios, para posterior visualização pelo fisioterapeuta.

#### 2.4.4 Comparação com trabalhos correlatos e discussões

No Quadro 4 é demonstrado um comparativo das principais características dos trabalhos citados na seção 2.4.

Quadro 4 - Comparação entre trabalhos correlatos

características / trabalhos relacionados	Pilon (2013)	Jintronix (2015)	Campos (2013)
permite a inserção de novos exercícios?	Sim	Não	Não
pode-se utilizar em diferentes tipos de tratamentos fisioterápicos?	Sim	Sim	Não

disponibiliza ao paciente instruções de como o exercício deve ser executado?	Sim	Não	Não
avalia a qualidade dos exercícios executados?	Não	Sim	Sim
armazena dados das sessões fisioterapêuticas?	Não	Sim	Sim
plataforma de desenvolvimento <i>web</i> ?	Não	Não	Não

A partir do Quadro 4 percebe-se que o trabalho de Pilon (2013) difere em relação ao Jintronix (2015) e Campos (2013) pois permite ao fisioterapeuta cadastrar novos exercícios e ao paciente visualizar o seu objetivo. Os trabalhos de Jintronix (2015) e Campos (2013) além de armazenarem os dados das sessões fisioterapêuticas, avaliam a qualidade dos movimentos executados e mostram em tempo real um *feedback* ao paciente durante à execução dos exercícios. Embora os trabalhos correlatos Jintronix (2015) e Campos (2013) sejam muito semelhantes, o trabalho desenvolvido por Campos (2013) efetua o acompanhamento e avaliação apenas de membros inferiores, enquanto o trabalho correlato Jintronix (2015) efetua o acompanhamento de membros superiores e inferiores. Das características listadas, percebe-se que nenhum dos trabalhos permite inserir novos exercícios e instruir o paciente como executá-los. Além disso, nenhum trabalho foi desenvolvido para -a plataforma *web*.

### 3 DESENVOLVIMENTO

As próximas seções descrevem os requisitos, a especificação e a implementação da aplicação. Também são apresentadas as ferramentas e bibliotecas utilizadas em cada parte da aplicação. Para finalizar, é mostrada a operacionalidade da aplicação.

#### 3.1 REQUISITOS

A aplicação para tratamentos fisioterapêuticos utilizando Kinect tem os seguintes Requisitos Funcionais (RF), Requisitos Não Funcionais (RNF) e Regras de Negócio (RN):

- a) permitir ao fisioterapeuta cadastrar paciente (RF);
- b) permitir ao fisioterapeuta cadastrar exercícios específicos utilizando Microsoft Kinect (RF);
- c) disponibilizar ao paciente, de forma visual e num ambiente 3D, o exercício que deverá ser executado (RF);
- d) capturar e salvar automaticamente os movimentos executados pelo paciente na execução dos exercícios (RF);
- e) exibir ao paciente, de forma visual e em tempo real, uma avaliação de desempenho (RF);
- f) permitir ao fisioterapeuta e paciente consultar o desempenho dos exercícios executados pelo paciente (RF);
- g) utilizar o sensor Microsoft Kinect com o SDK Kinect *for* Windows versão 1.8 para captura de movimentos (RNF);
- h) o *backend* da aplicação deverá ser implementado na linguagem de programação C# e utilizar os SDKs Kinect *for* Windows e *Entity Framework* (RNF);
- i) o *frontend* da aplicação deverá ser implementado em ASP.NET MVC (RNF);
- j) a interface gráfica deverá ser implementada utilizando a biblioteca ThreeJS (RNF);
- k) comunicação entre interface gráfica e Kinect deve ser assíncrona e através de *Websockets* (RNF);
- l) ser executado por qualquer navegador que suporte HTML5, Javascript, *WebSockets* e WebGL (RNF);
- m) utilizar o banco de dados SQL Server (RNF);
- n) capturar movimentos de apenas um usuário (RN).

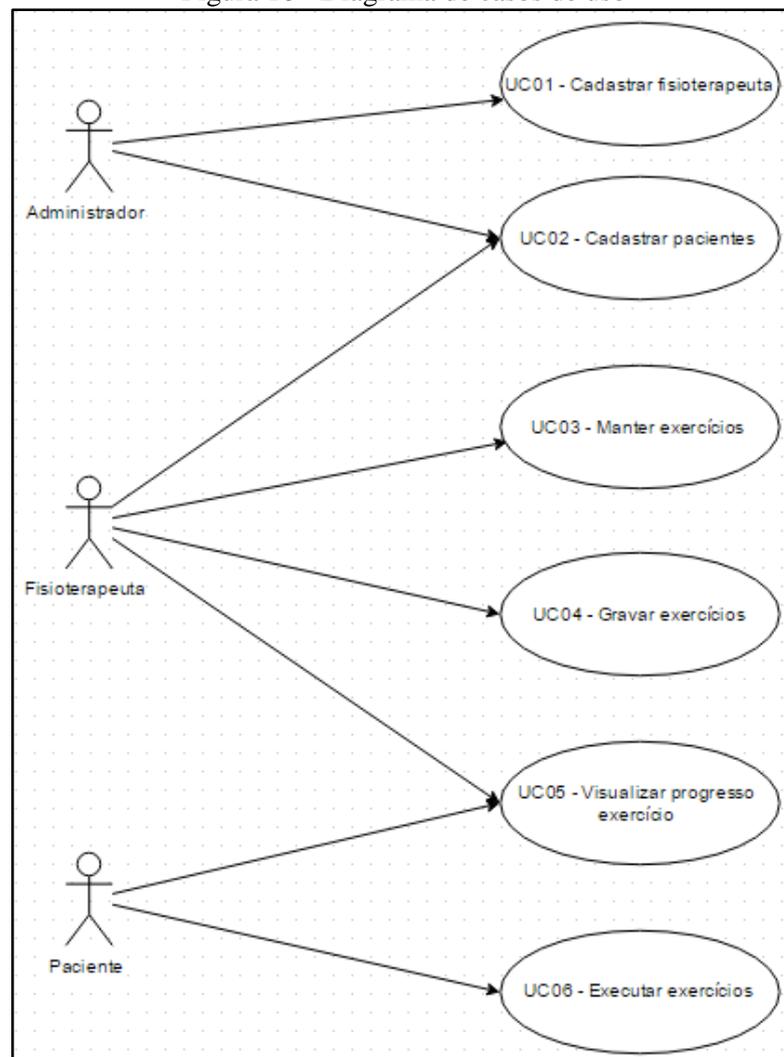
### 3.2 ESPECIFICAÇÃO

Nesta seção é apresentada toda especificação da aplicação. Foram elaborados os diagramas de casos de uso, classes e atividades da *Unified Modeling Language* (UML), os quais foram desenvolvidos utilizando a ferramenta draw.io.

#### 3.2.1 DIAGRAMA DE CASOS DE USO

Nesta seção são descritos os casos de uso referentes à Figura 10. A aplicação possui seis cenários e três atores, sendo eles: Administrador, Fisioterapeuta e Paciente.

Figura 10 - Diagrama de casos de uso



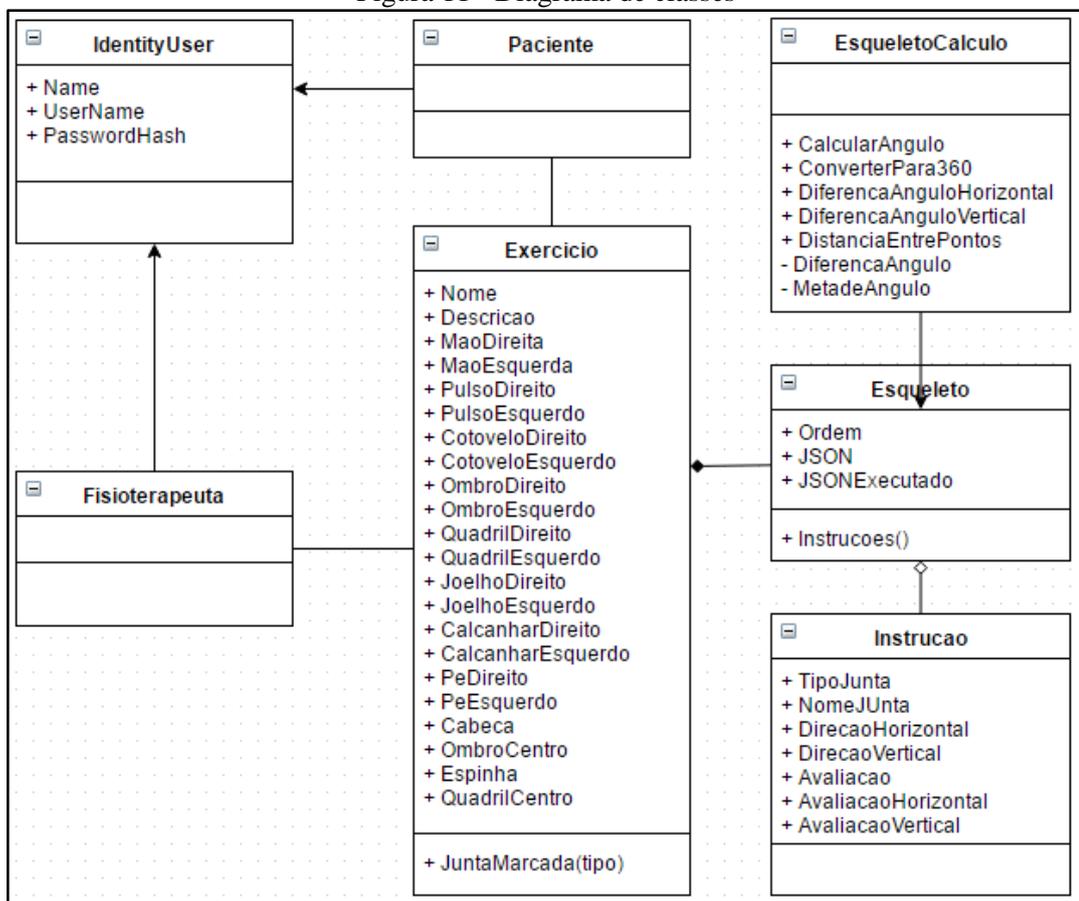
No caso de uso UC01: Cadastrar fisioterapeuta o Administrador pode cadastrar novos fisioterapeutas para que estes tenham acesso à aplicação. O caso de uso UC02: Cadastrar pacientes permite ao Administrador e Fisioterapeuta cadastrar novos pacientes. O caso de uso UC03: Manter exercícios serve para o Fisioterapeuta inserir, editar e excluir exercícios. O caso de uso UC04: Gravar exercícios permite ao

Fisioterapeuta cadastrar as posições do exercício, as quais serão posteriormente executadas pelo Paciente. No caso de uso UC05: Visualizar progresso exercício, o Fisioterapeuta e o Paciente podem visualizar o progresso da execução do exercício, após este ter sido executado pelo Paciente. O caso de uso UC06: Executar exercícios permite ao Paciente carregar as posições a serem executadas, repetir e gravar os dados da execução no exercício. O detalhamento dos casos de uso da aplicação está no Apêndice A.

### 3.2.2 DIAGRAMA DE CLASSES

Nesta seção está descrita a estrutura de classes da aplicação desenvolvida. A Figura 11 representa o diagrama de classes que compõe a aplicação assim como seus relacionamentos.

Figura 11 - Diagrama de classes



As classes `Fisioterapeuta` e `Paciente` herdam da classe `IdentityUser` do *framework* ASP.NET MVC, fazendo que cada paciente e médico cadastrado possuam um usuário vinculado para que possam acessar o sistema. A classe `Exercicio` é a classe que contém informações básicas do exercício e quais juntas do corpo devem ser avaliadas.

A classe `Esqueleto` e `EsqueletoCalculo` são as principais classes da aplicação. Elas possuem a lógica de avaliação das posições e as instruções de como o paciente deve executar o

exercício. A classe `Esqueleto` controla a posição do exercício, orienta o paciente durante a execução do exercício, efetua os cálculos de percentual de acerto, assim como, armazena os dados de execução realizados pelo paciente. A classe `EsqueletoCalculo` possui métodos auxiliares utilizados nos cálculos feitos pela classe `Esqueleto`. Por sua vez, a classe `Instrucao` é a classe utilizada para representar uma instrução a qual é posteriormente enviada ao usuário durante a execução do exercício. A classe `Instrucao` também possui os percentuais de acerto, sendo eles vertical, horizontal e diagonal.

### 3.3 IMPLEMENTAÇÃO

Nesta seção são descritas as técnicas e ferramentas utilizadas para o desenvolvimento da aplicação, a descrição do desenvolvimento do trabalho e a operacionalidade do aplicativo.

#### 3.3.1 Técnicas e ferramentas utilizadas

A aplicação é dividida em duas camadas, sendo estas a camada de apresentação e a camada de negócio, formando assim uma arquitetura cliente-servidor. A camada de apresentação possui a responsabilidade de enviar e receber informações da camada de negócio e apresentá-las ao usuário de forma amigável. A camada de negócios por sua vez possui a responsabilidade de receber as informações do sensor Kinect e enviá-las à camada de apresentação. É nela que ocorrem os cálculos de validação e instruções ao usuário, assim como a armazenagem dos dados.

A camada de apresentação foi desenvolvida utilizando um conjunto de linguagens como ASP.NET MVC, JavaScript, CSS e HTML. Além das linguagens citadas, também utilizou-se as seguintes bibliotecas: (a) *framework web* Bootstrap, para facilitar o desenvolvimento das páginas além de deixá-las mais amigável ao usuário; (b) a biblioteca SignalR para auxiliar na comunicação via *WebSocket* entre a camada de apresentação e a camada de negócio; (c) Three.js a qual é utilizada para desenhar o ambiente e o boneco na mesma posição do usuário no mundo exterior; (d) a biblioteca RainbowVis que é utilizada para calcular a cor de degrade utilizada na avaliação da execução.

Para desenvolvimento da camada de negócio, utilizou-se da linguagem C# da plataforma Microsoft .NET. Também foram utilizadas algumas bibliotecas, as quais são:

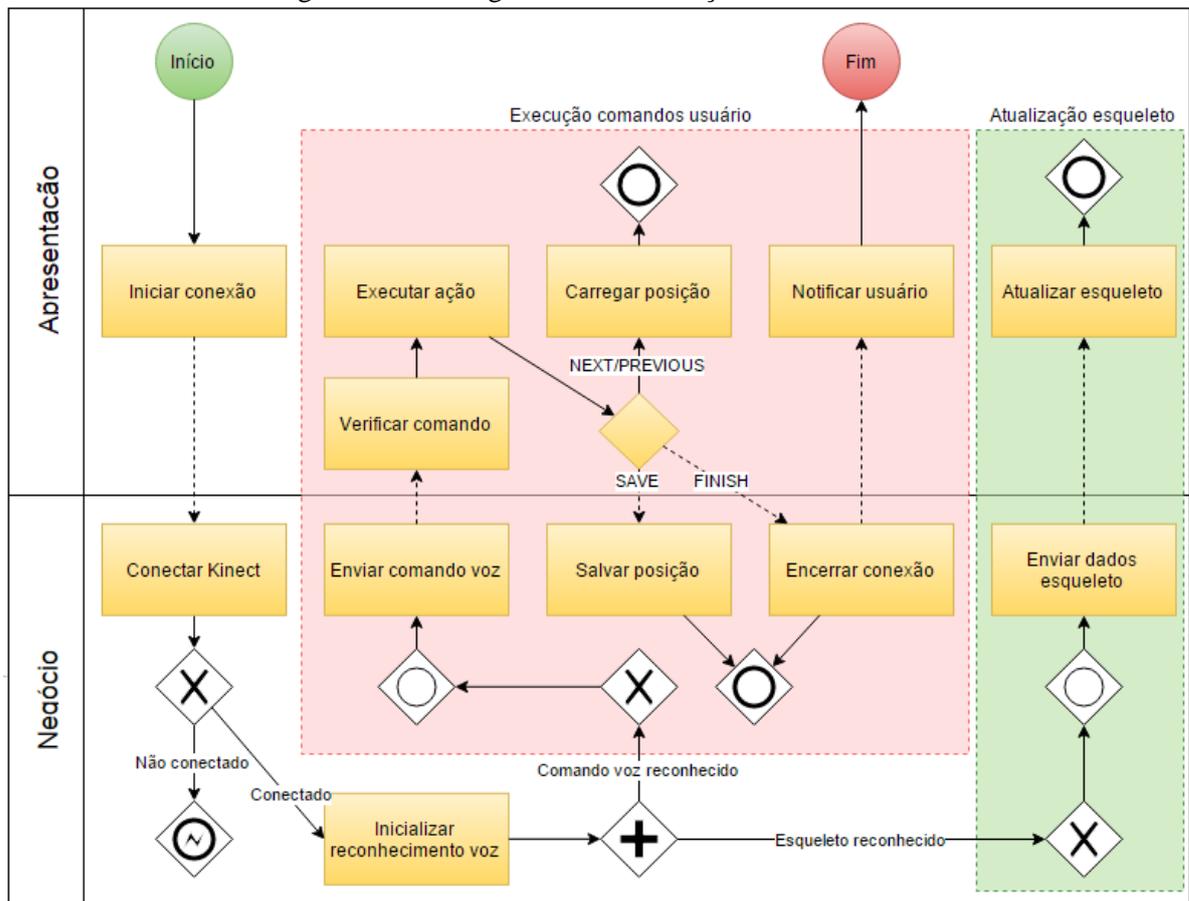
- a) Kinect *for* Windows: conectar e carregar dados do sensor Kinect;
- b) Microsoft *Speech Engine*: reconhecimento de palavras por áudio;
- c) *Entity Framework*: persistir e carregar dados da base de dados;
- d) ASP.NET SignalR: comunicação via *WebSocket* com a camada de apresentação.

Para persistência dos dados, o banco de dados SQL Server foi escolhido por possuir biblioteca oficial desenvolvida para plataforma .NET, sendo assim sua a integração e acesso a dados se torna mais fácil. O modelo de entidade e relacionamento da aplicação está no Apêndice B.

### 3.3.2 Fluxo geral de comunicação entre camadas

Para melhor representação e entendimento da estrutura de comunicação entre as camadas de apresentação e negócio da aplicação foi elaborado o diagrama de atividades da Figura 12.

Figura 12 - Fluxo geral de comunicação entre camadas



Na Figura 12 é possível notar que o fluxo é iniciado e finalizado pela camada de apresentação. A camada de apresentação inicia o serviço de conexão no servidor e então permanece recebendo notificações assíncronas da camada de negócio por tempo indeterminado. Basicamente, internamente a aplicação possui dois subfluxos. Um destes, o qual está destacado pela cor verde, é responsável pela atualização das informações do esqueleto, que contém as informações de coordenadas das articulações e instruções de orientação. O outro subfluxo destacado pela cor rosa, é responsável por orquestrar a interação com o usuário através dos

comandos de voz, onde de acordo com o comando será disparada uma ação. O fluxo por sua vez, só termina quando o usuário falar o comando “*FINISH*”, para que então a camada de apresentação envie uma mensagem para a camada de negócio solicitando o término da conexão para posteriormente notificar o usuário.

### 3.3.3 Implementação da aplicação

Nesta seção é mostrado de forma detalhada como é feita a captura dos movimentos pelo Kinect até a representação do esqueleto do usuário no ambiente virtual, assim como a validação das posições do exercício.

#### 3.3.3.1 Conexão com o Sensor Kinect

Para se conectar ao sensor Kinect, utiliza-se a classe `KinectSensor` do SDK Kinect *for* Windows. O Quadro 5 mostra o trecho de código que efetua a conexão.

Quadro 5 - Método de conexão com o sensor Kinect

```

01  _sensor = KinectSensor.KinectSensors.FirstOrDefault(k => k.Status ==
    KinectStatus.Connected);
02  if (_sensor == null) return null;
03  _sensor.SkeletonStream.Enable();
04  _sensor.SkeletonFrameReady += Sensor_SkeletonFrameReady;
05  try
06  {
07      _sensor.Start();
08  }
09  catch (Exception ex)
10  {
11      Log("Erro ao iniciar Kinect: " + ex.Message);
12  }

```

Para capturar os esqueletos reconhecidos pelo Kinect é necessário assinar o evento `SkeletonFrameReady` da classe `KinectSensor`. Desta forma, quando o Kinect atualizar os dados dos esqueletos reconhecidos, este método é chamado com as informações atualizadas. As informações dos esqueletos são passadas por parâmetro através da classe `SkeletonFrameReadyEventArgs`, com isto é possível recuperar as posições das juntas do esqueleto reconhecido que posteriormente são passados para o método `EsqueletoAtualizado` que serializa e envia as informações para a camada de apresentação, conforme mostra o Quadro 6.

Quadro 6 - Recuperando dados dos esqueletos

```

01 private void Sensor_SkeletonFrameReady(object sender,
                                SkeletonFrameReadyEventArgs e){
02     var frame = e.OpenSkeletonFrame();
03     if (frame != null) {
04         var esqueleto = frame.GetFirstOrDefault();
05
06         if (esqueleto.TrackingState ==
                                SkeletonTrackingState.Tracked) {
                                EsqueletoAtualizado(esqueleto.Joints);
07     }
08     else
09         Log("Esqueleto/junta nao localizado.
TrackState: " + esqueleto.TrackingState);
10     }
11 }
12 }

```

Se nenhum esqueleto for localizado, será enviada a mensagem “Esqueleto/junta não localizado.” através do método `Log`.

### 3.3.3.2 Inicialização da API de reconhecimento de voz

O método `InicializarReconhecimentoVoz` é chamado juntamente com a inicialização do sensor Kinect. O código do Quadro 7 mostra como é feita a inicialização da API de reconhecimento de voz e a passagem do áudio capturado pelo Kinect para o objeto de reconhecimento do tipo `SpeechRecognitionEngine` da biblioteca `Microsoft.Speech`.

Quadro 7 - Inicialização da API de reconhecimento de voz

```

01 private RecognizerInfo getReconhecimentoDeFala() {
02     foreach (var recognizer in
03         SpeechRecognitionEngine.InstalledRecognizers()) {
04         string value;
05         recognizer.AdditionalInfo.TryGetValue("Kinect", out
06         value);
07         if ("True".Equals(value,
08             StringComparison.OrdinalIgnoreCase) && "en-
09             US".Equals(recognizer.Culture.Name,
10             StringComparison.OrdinalIgnoreCase))
11             return recognizer;
12     }
13     return null;
14 }
15 private void InicializarReconhecimentoVoz() {
16     RecognizerInfo ri = getReconhecimentoDeFala();
17     this.speechEngine = new SpeechRecognitionEngine(ri.Id);
18     using (var memoryStream = new MemoryStream(
19         Encoding.ASCII.GetBytes
20         (FisioK.Properties.Resources.SpeechGrammar))) {
21         var g = new Grammar(memoryStream);
22         speechEngine.LoadGrammar(g);
23     }
24     speechEngine.SetInputToAudioStream(
25         _sensor.AudioSource.Start(), new SpeechAudioFormatInfo
26         (EncodingFormat.Pcm, 16000, 16, 1, 32000, 2, null));
27     speechEngine.RecognizeAsync(RecognizeMode.Multiple);
28     speechEngine.SpeechRecognized += SpeechEngine_SpeechRecognized;
29 }

```

O áudio capturado pelo Kinect é enviado à API de reconhecimento de voz, que irá processar os dados recebidos. Se for constatada sua presença, o método `SpeechRecognized` será chamado, conforme mostra o Quadro 7. Os dados do áudio reconhecido são passados por parâmetro através da classe `SpeechRecognizedEventArgs`, conforme mostra o Quadro 8.

Quadro 8 - Recuperando comandos de voz reconhecidos

```

01 private void SpeechEngine_SpeechRecognized(object sender,
02     SpeechRecognizedEventArgs e)
03 {
04     ComandoVoz(e.Result.Text);
05 }

```

O comando de voz reconhecido é retornado numa propriedade do tipo `String`, que é então enviada à camada de apresentação através do método `ComandoVoz`.

### 3.3.3.3 Configuração da gramática para reconhecimento de voz

Para utilizar a API de reconhecimento de voz é necessário configurar um vocabulário (gramática) com as palavras que este irá reconhecer. O Quadro 9 mostra como é feita a inclusão de palavras na gramática e quais palavras foram utilizadas na aplicação.

Quadro 9 - Configuração do vocabulário da gramática para reconhecimento de voz

```

<item>
  <tag>NEXT</tag>
  <one-of>
    <item> next </item>
  </one-of>
</item>
<item>
  <tag>PREVIOUS</tag>
  <one-of>
    <item> previous </item>
  </one-of>
</item>
<item>
  <tag>SAVE</tag>
  <one-of>
    <item> save </item>
  </one-of>
</item>
<item>
  <tag>FINISH</tag>
  <one-of>
    <item> finish </item>
  </one-of>
</item>

```

Na gramática foram adicionadas somente as palavras reconhecidas como comandos dentro da aplicação, para diminuir a probabilidade da API reconhecer de forma incorreta uma palavra dita pelo usuário. Os comandos foram colocados em inglês devido a API de reconhecimento de voz reconhecer somente palavras em inglês.

#### 3.3.3.4 Comunicação entre camada de negócio e apresentação

Conforme citado anteriormente, utilizou-se a biblioteca SignalR para auxiliar a comunicação via *websockets*. Para utilizar esta biblioteca criou-se na camada de negócio a classe `KinectHub` que herda da classe `Hub`, a qual pertence ao SignalR, conforme mostra o Quadro 10.

Quadro 10 - Classe derivada SignalR

```

01 public class KinectHub : Hub
02 {
03     public void Iniciar(long idExercicio)
04     {
05         try
06         {
07             Parar();
08             Log("Iniciando Microsoft Kinect.");
09             Log(Sensor != null ? "Kinect conectado" : "Nenhum
kinect encontrado.");
10
11             using (var db = new DataContext())
12             {
14                 ExercicioAtual =
db.Exercicio.Find(idExercicio);
15                 Log(string.Format("Exercício {0}
iniciado.", ExercicioAtual.Nome));
16             }
17         }
18         catch (Exception ex)
19         {
20             Log("Erro ao iniciar Kinect: " + ex.Message);
21         }
22     }
23 }

```

Quando o projeto da aplicação é compilado, são geradas classes em JavaScript para serem utilizadas na camada de apresentação. A biblioteca SignalR encarrega-se de gerenciar a comunicação entre a camada de negócio e a de apresentação. Para criar uma conexão com a camada de negócio, utiliza-se a classe gerada em JavaScript a partir da classe da camada de negócio, conforme mostra o Quadro 11:

Quadro 11 - Conexão com a camada de negócio utilizando SignalR

```

01 var kinect = $.connection.kinectHub;
02 $(function () {
03     kinect.client.log = log;
04     kinect.client.esqueletoAtualizado = desenharEsqueletoKinect;
05     kinect.client.comandoVoz = comandoVoz;
06     // Inicializar conexao com kinect e posicionar exercicio
07     $.connection.hub.start().done(function () {
08         kinect.server.iniciar(@Model.Id);
09     });
10 });

```

Da mesma forma que é feita na camada de negócio (linguagem C#), faz-se na camada de apresentação em JavaScript. É necessário assinar os métodos necessários para receber atualizações de eventos, como os eventos `esqueletoAtualizado` e `comandoVoz`, conforme exhibe o Quadro 11.

### 3.3.4 Representação virtual do esqueleto na aplicação

A biblioteca Three.JS foi utilizada para representar o esqueleto do usuário do mundo real no mundo virtual. Para uma melhor visualização e entendimento de como o exercício deve ser executado, a aplicação possui dois cenários: visualização frontal e visualização diagonal lateral.

Para criação do ambiente utilizou-se dos seguintes objetos: THREE.PerspectiveCamera, THREE.PlaneBufferGeometry, THREE.Mesh, THREE.Scene, THREE.AmbientLight, THREE.DirectionalLight e THREE.WebGLRenderer.

Primeiramente inicializa-se o objeto de câmera, conforme mostra o Quadro 12.

Quadro 12 - Inicialização do objeto câmera

01	this.camera = new THREE.PerspectiveCamera(30, 300 / 400, 0.1, 100);
02	this.camera.position.x = this.cameraPosition.X;
03	this.camera.position.y = this.cameraPosition.Y;
04	this.camera.position.z = this.cameraPosition.Z;
05	
06	if (this.direcaoCamera == null)
07	this.camera.lookAt(new THREE.Vector3(0, 0, -4));
08	else
09	this.camera.lookAt(this.direcaoCamera);

As posições x, y e z podem variar de acordo com o ângulo desejado (frontal ou diagonal). O Quadro 13 demonstra como criar uma forma geométrica para representar o chão no cenário. Com isto, o usuário tem maior percepção de participação no cenário por ter um ponto para basear-se dentro do cenário virtual.

Quadro 13 - Criação da forma geométrica para representar o chão do cenário

01	var geo = new THREE.PlaneBufferGeometry(20, 25);
02	this.chao = new THREE.Mesh(geo, new THREE.MeshBasicMaterial({ color: 0xcfcfcf }));
03	this.chao.material.side = THREE.DoubleSide;
04	this.chao.rotation.x = Math.PI / 2;
05	this.chao.position.y = -1;
06	this.chao.receiveShadow = true;

O Quadro 14 mostra como foi feita a iluminação do cenário, onde criou-se uma iluminação geral e outra direcional para criar a sombra dos objetos do cenário.

Quadro 14 - Inicialização da iluminação direcional e ambiente da cena

01	<code>this.iluminacao = new THREE.DirectionalLight(0xe0e0e0);</code>
02	<code>this.iluminacao.position.set(this.iluminacaoPosition.X,</code>
03	<code>this.iluminacaoPosition.Y, this.iluminacaoPosition.Z).normalize();</code>
04	<code>this.iluminacao.castShadow = true;</code>
05	<code>this.iluminacao.shadowDarkness = 0.5;</code>
06	<code>this.iluminacao.shadowCameraRight = 5;</code>
07	<code>this.iluminacao.shadowCameraLeft = -5;</code>
08	<code>this.iluminacao.shadowCameraTop = 4.5;</code>
09	<code>this.iluminacao.shadowCameraBottom = -5;</code>
10	<code>this.iluminacao.shadowCameraNear = 1;</code>
11	<code>this.iluminacao.shadowCameraFar = 100;</code>
12	
13	<code>this.luzAmbiente = new THREE.AmbientLight(0x101010);</code>

O Quadro 15 demonstra a inserção dos objetos previamente criados ao cenário.

Quadro 15 - Adição dos objetos na cena

01	<code>this.scene = new THREE.Scene();</code>
02	<code>this.scene.add(this.chao);</code>
03	<code>this.scene.add(this.iluminacao);</code>
04	<code>this.scene.add(this.luzAmbiente);</code>

O Quadro 16 demonstra como efetuar a renderização da cena e adição da cena final à página para que então o usuário possa visualiza-la.

Quadro 16 - Renderização da cena e adição à página

01	<code>this.renderizador = new THREE.WebGLRenderer();</code>
02	<code>this.renderizador.setSize(300, 400);</code>
03	<code>this.renderizador.setClearColor(0x000000, 1);</code>
04	<code>this.renderizador.shadowMapEnabled = true;</code>
05	<code>this.renderizador.render(this.scene, this.camera);</code>
06	
07	<code>var div = document.getElementById(this.idDivCanvas);</code>
08	<code>div.appendChild(this.renderizador.domElement);</code>

O Quadro 17 abaixo demonstra como são posicionadas às juntas do esqueleto do usuário no cenário. Primeiramente, as posições do esqueleto e as instruções são recebidos da camada de negócio no formato JSON, onde para cada junta é criada uma esfera e esta é adicionada e posicionada no cenário conforme sua localização.

Quadro 17 - Atualização do esqueleto no cenário

```

01 this.refreshData = function(json, instrucoes) {
02     var jsonObject = eval(json);
03     var instrucoesObject = eval(instrucoes);
04     var geoBola = new THREE.SphereGeometry(0.07, 25, 25);
05
06     for (var i = 0, count = jsonObject.length; i < count; i++) {
07         var joint = jsonObject[i];
08         var locX = joint.X;
09         var locY = joint.Y;
10         var locZ = (joint.Z * -1) - 1.5;
11
12         var cor = this.getJointColor(instrucoesObject, joint);
13         var objCor = new THREE.MeshPhongMaterial({ color: cor });
14         var sphere = new THREE.Mesh(geoBola, objCor);
15         sphere.position.set(locX, locY, locZ);
16         sphere.castShadow = true;
17         sphere.tipoJunta = joint.TipoJunta;
18         this.scene.add(sphere);
19     }
20
21     this.ligarEsferas(jsonObject);
22     this.render();
23 };

```

O método `getJointColor` é responsável por retornar a cor de cada junta. Ele retornará a cor azul por padrão e vermelho quando a junta estiver marcada para ser verificada. Caso existam instruções associadas à junta, será utilizada a biblioteca `RainbowVis` para mostrar uma escala de cor conforme a avaliação do posicionamento da junta em relação à posição do fisioterapeuta, conforme exibe o trecho de código do Quadro 18.

Quadro 18 - Montar cor da junta

```

01 this.getJointColor = function (instrucoes, joint) {
02     if (instrucoes == null)
03         return (joint.Marcada ? this.corMarcada : this.corPadrao)
04
05     for (var i = 0; i < instrucoes.length; i++)
06         if (instrucoes[i].TipoJunta == joint.TipoJunta)
07             return '#' +
08 this.rainbow.colourAt(parseInt(instrucoes[i].Avaliacao, 10));
09
10     return this.corPadrao;
11 }

```

Para uma maior nitidez da representação do esqueleto em relação à realidade, as juntas do esqueleto são representadas por esferas e interligadas por cilindros através do método `ligarEsferas`, conforme mostra o Quadro 19.

Quadro 19 - Desenhar ligações das juntas

```

01 this.ligarEsferas = function (jsonObject) {
02     var ligacoes = this.getLigacoes();
03     for (var i = 0; i < ligacoes.length; i++) {
04         var j1 = this.getJoint(ligacoes[i].j1);
05         var j2 = this.getJoint(ligacoes[i].j2);
06         this.desenharCilindro(j1.position, j2.position,
this.scene);
07     }
08 }
09
10 this.desenharCilindro = function (vstart, vend, scene) {
11     var distance = vstart.distanceTo(vend);
12     var cylinder = new THREE.CylinderGeometry(.03, .03, distance,
20, false);
13     var orientation = new THREE.Matrix4();
14     orientation.lookAt(vstart, vend, new THREE.Vector3(0, 1, 0));
15     var offsetRotation = new THREE.Matrix4();
16     offsetRotation.makeRotationX(Math.PI / 2);
17     orientation.multiply(offsetRotation);
18     cylinder.applyMatrix(orientation)
19     cylinder.castShadow = true;
20
21     var material = new THREE.MeshLambertMaterial({ color:
22 this.corPadrao });
23     var mesh = new THREE.Mesh(cylinder, material);
24     var position = vend.clone().add(vstart).divideScalar(2);
25     mesh.position.x = position.x;
26     mesh.position.y = position.y;
27     mesh.position.z = position.z;
28     mesh.castShadow = true;
29     scene.add(mesh);
30 }

```

O Quadro 20 mostra os métodos auxiliares `getLigacoes` e `getJoint` utilizados no Quadro 19.

Quadro 20 - Métodos auxiliares para ligar esferas ao cenário

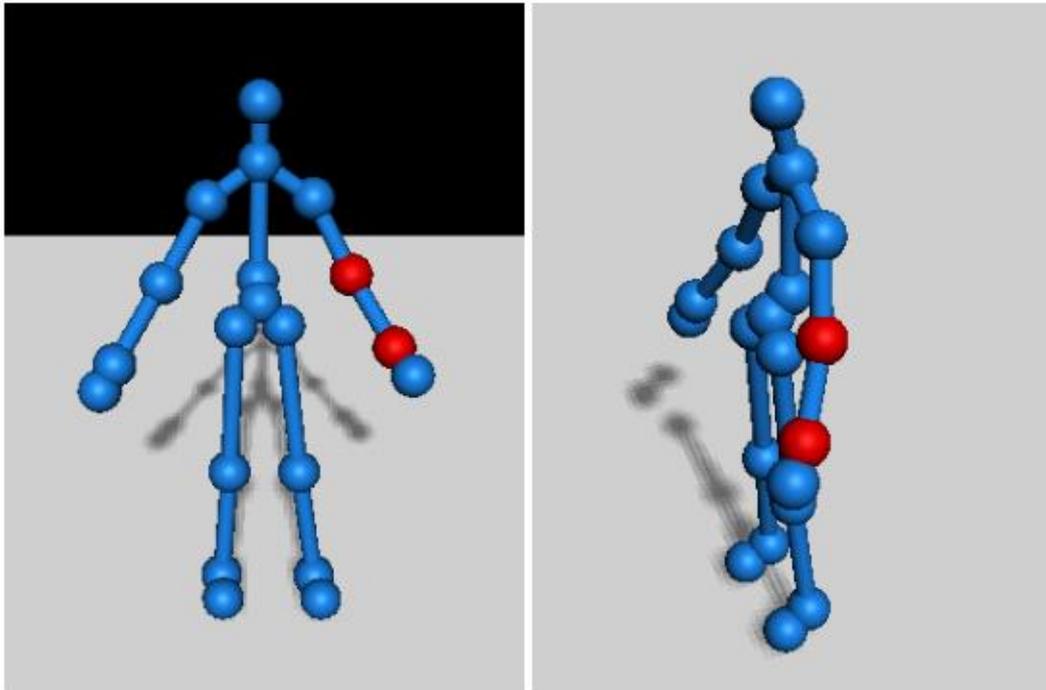
```

01 this.getLigacoes = function() {
02     return [{j1:1,j2:2}, {j1:16,j2:17}, {j1:12,j2:13},
03         {j1:17,j2:18}, {j1:13,j2:14}, {j1:5,j2:6}, {j1:9,j2:10},
04         {j1:2,j2:4}, {j1:2,j2:8}, {j1:4,j2:5}, {j1:8,j2:9},
05         {j1:2,j2:3}, {j1:6,j2:7}, {j1:10,j2:11}];
06 }
07 this.getJoint = function (jointType) {
08     for (var i = 0, count = this.scene.children.length; i < count;
09 i++) {
10         if (this.scene.children[i].tipoJunta == jointType)
11             return this.scene.children[i];
12     }
13     return null;
14 }

```

A Figura 13 mostra o esqueleto desenhado com as posições provenientes do Kinect.

Figura 13 - Esqueleto desenhado com informações do Kinect



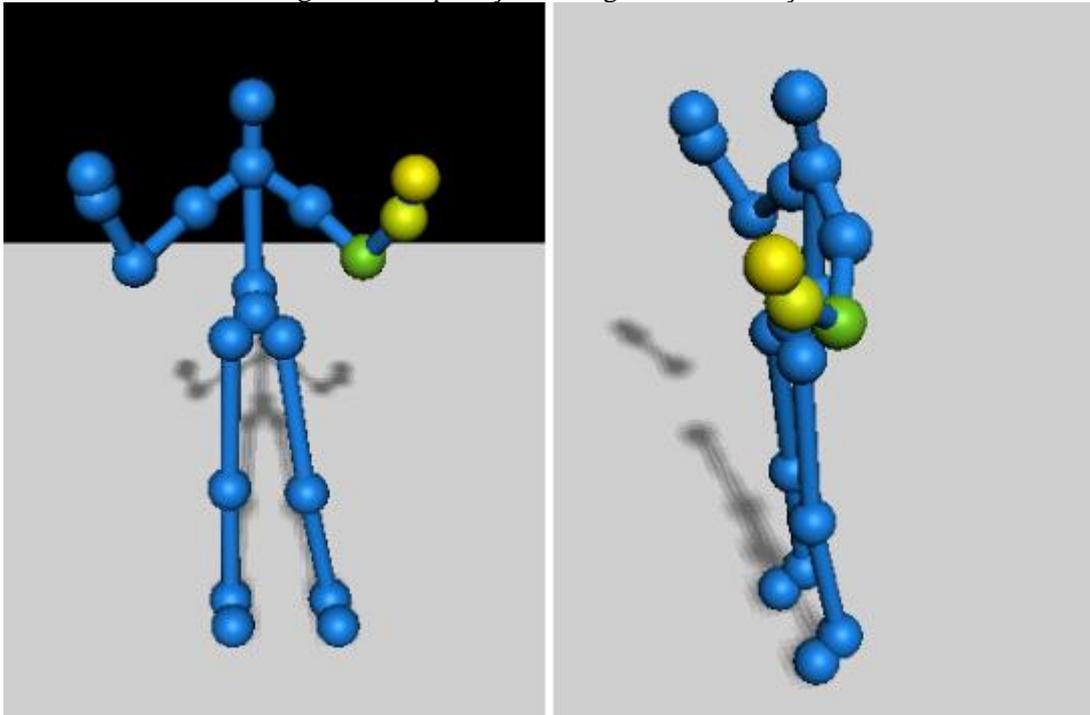
A partir da Figura 13, pode-se observar que o esqueleto foi desenhado corretamente. As juntas na cor vermelha são as juntas marcadas para serem validadas. Quando o exercício estiver em execução pelo paciente, as juntas marcadas terão uma cor que varia da cor verde à cor vermelha de acordo com o percentual de acerto em relação à posição do fisioterapeuta, conforme representado na Figura 14.

Figura 14 - Degradê de cores utilizadas na avaliação das juntas



A Figura 15 mostra a simulação do degradê ao ser aplicado na validação das juntas.

Figura 15 - Aplicação do degrade de validação



A partir da Figura 13, pode-se observar a posição gravada pelo fisioterapeuta e, na Figura 15, a execução realizada pelo paciente. Percebe-se que o cotovelo direito do paciente está melhor posicionado do que as demais juntas, fazendo com que este receba a cor verde e, consecutivamente tenha uma maior taxa de acerto.

### 3.3.5 Validação e instruções

Nesta seção é mostrado como a aplicação valida a execução das posições para retornar o percentual de acerto, assim como as instruções para auxiliar o paciente atingir o posicionamento correto da posição.

Primeiramente, recupera-se as informações das duas posições a serem comparadas (fisioterapeuta e paciente) e as demais informações de ambos esqueletos, conforme mostra o Quadro 21.

Quadro 21 - Informações do fisioterapeuta e paciente

01	<code>var instrucoes = new List&lt;Instrucao&gt;();</code>
02	<code>var juntasMarcadas =</code>
03	<code>Enum.GetValues(typeof(JointType)).Cast&lt;JointType&gt;().Where(jt =&gt;</code>
	<code>this.Exercicio.JuntaMarcada(jt));</code>
05	
06	<code>List&lt;Junta&gt; juntas1 = new</code>
	<code>JavaScriptSerializer().Deserialize&lt;List&lt;Junta&gt;&gt;(JSON);</code>
07	<code>List&lt;Junta&gt; juntas2 = new</code>
	<code>JavaScriptSerializer().Deserialize&lt;List&lt;Junta&gt;&gt;(json2);</code>
08	
09	<code>var centro1 = juntas1.FirstOrDefault(j =&gt; j.TipoJunta ==</code>
	<code>JointType.ShoulderCenter);</code>
10	<code>var centro2 = juntas2.FirstOrDefault(j =&gt; j.TipoJunta ==</code>
	<code>JointType.ShoulderCenter);</code>
11	
12	<code>var altural =</code>
	<code>EsqueletoCalculo.DistanciaEntrePontos(juntas1.FirstOrDefault(j =&gt;</code>
	<code>j.TipoJunta == JointType.Spine), centro1);</code>
13	<code>var altura2 =</code>
	<code>EsqueletoCalculo.DistanciaEntrePontos(juntas2.FirstOrDefault(j =&gt;</code>
	<code>j.TipoJunta == JointType.Spine), centro2);</code>

É criada uma lista de objetos da classe `Instrucao`, a qual contém atributos para armazenar as instruções horizontais, verticais e diagonais e suas respectivas avaliações. Busca-se então no cadastro do exercício quais são as juntas que devem ser validadas, posteriormente recupera-se as informações de todas juntas de ambos esqueletos, para então obter as coordenadas de origem utilizada para calcular os ângulos e a altura parcial utilizada no cálculo de proporcionalização. O ponto de origem é a junta do centro do ombro, conforme já mostrado na Figura 3. A altura parcial utilizada na proporcionalização é a distância entre o centro do ombro e a junta da espinha.

A verificação de acerto da execução é baseada em ângulos, assim a diferença de tamanho entre fisioterapeuta e paciente não afeta o cálculo. O ângulo é calculado para cada junta a ser validada e o ponto de origem é o centro do ombro do usuário. Para validar se o paciente está posicionando, a junta marcada no local correto, faz-se três validações, sendo estas horizontal, vertical e diagonal, conforme exibe o trecho de código do Quadro 22.

Quadro 22 - Cálculo para validar posicionamento

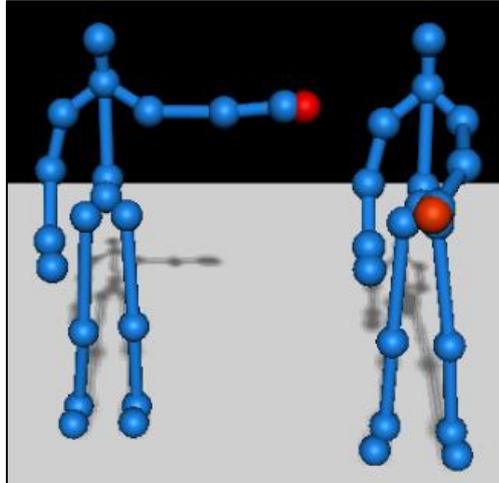
```

01 foreach (var junta in juntasMarcadas)
02 {
03     var j1 = juntas1.FirstOrDefault(j => j.TipoJunta == junta);
04     var j2 = juntas2.FirstOrDefault(j => j.TipoJunta == junta);
05
06     var angV1 = EsqueletoCalculo.CalcularAngulo(centro1.X,
07                                                centro1.Y, j1.X, j1.Y);
08     var angH1 = EsqueletoCalculo.CalcularAngulo(centro1.X,
09                                                centro1.Z, j1.X, j1.Z);
10
11     var angV2 = EsqueletoCalculo.CalcularAngulo(centro2.X,
12                                                centro2.Y, j2.X, j2.Y);
13     var angH2 = EsqueletoCalculo.CalcularAngulo(centro2.X,
14                                                centro2.Z, j2.X, j2.Z);
15
16     var pontuacaoVertical = 100 - ((100.0 / 180.0) *
17                                     EsqueletoCalculo.DiferencaAngulo(angV1, angV2));
18     var pontuacaoHorizontal = 100 - ((100.0 / 180.0) *
19                                     EsqueletoCalculo.DiferencaAngulo(angH1, angH2));
20
21     var ratio1 = EsqueletoCalculo.DistanciaEntrePontos(centro1,
22                                                         j1) /
23     altura1;
24     var ratio2 = EsqueletoCalculo.DistanciaEntrePontos(centro2,
25                                                         j2) /
26     altura2;
27     var pontuacaoProximidade = (100 / ratio1) * ratio2;
28     pontuacaoProximidade = (pontuacaoProximidade > 100) ? 100 -
29     (pontuacaoProximidade - 100) : pontuacaoProximidade;
30 }

```

O método `CalcularAngulo` retorna um número que pode variar de 0 a 360, representando o grau da junta em relação à junta de origem. Um ponto pode estar no máximo a 180° de diferença em relação à outro ponto, logo pode-se dizer que 1° (primeiro grau) corresponde a 0,55%. Utiliza-se o método `DiferencaAngulo` para retornar a diferença entre o ângulo da junta do fisioterapeuta e do paciente, onde este método retorna um número de 0 a 180, usado para calcular o percentual de acerto do paciente. A Figura 16 mostra como é obtido o ângulo da junta em relação à origem.

Figura 16 - Cálculo de graus em relação à origem



No esqueleto do lado esquerdo, considerando o efeito de espelhamento, a mão direita está posicionada à direita, alinhada com o centro do ombro, obtendo assim um ângulo  $0^\circ$ . No caso do esqueleto do lado direito, observa-se que a mão direita está posicionada à frente, obtendo assim um ângulo de  $90^\circ$  em relação ao centro do ombro. Neste caso, a rotina de cálculo de acerto irá retornar 50% de acerto, visto que a diferença de  $90^\circ$  entre às duas posições equivale à 50% da diferença máxima que é de  $180^\circ$ . O percentual de acerto vertical é feito do mesmo modo que o horizontal, porém utiliza-se de coordenadas diferentes.

Além de validar horizontalmente e verticalmente, é necessário validar a distância diagonal, já que o paciente pode estar posicionado corretamente de acordo com os ângulos vertical e horizontal, porém pode estar muito próximo ou muito afastado da posição correta. Visto que fisioterapeuta e paciente podem ter diferentes tamanhos, é necessário proporcionalizar a distância do ponto de origem de acordo com o tamanho do usuário. A altura do usuário é obtida com base na distância entre o centro do ombro e o fim da espinha através do método `DistanciaEntrePontos`. Com esta medida consegue-se proporcionalizar a distância e verificar se há a necessidade de afastar ou aproximar. Após o cálculo das três direções, é feita a média aritmética que é visualizada posteriormente a execução do exercício.

Após calcular o percentual de acerto das três direções, precisa-se calcular as direções de cada junta que o paciente precisa fazer para atingir a posição cadastrada pelo fisioterapeuta. No Quadro 23 é mostrado como calcula-se a direção horizontal e vertical.

Quadro 23 - Cálculo de direção horizontal e vertical

```

01 var dif = DiferencaAngulo(angOrigem, angDestino);
02 var soma = (angOrigem + dif);
03 var sub = (angOrigem - dif);
04 if (soma > 360) soma -= 360;
05 if (sub < 0) sub += 360;
06
07 if (Math.Round(soma, MidpointRounding.ToEven) ==
Math.Round(angDestino, MidpointRounding.ToEven))
08     return (calculo == TipoCalculo.Horizontal ? Direcao.Direita
09 :
10     Direcao.Acima);
else if (Math.Round(sub, MidpointRounding.ToEven) ==
Math.Round(angDestino, MidpointRounding.ToEven))
11 // Se igual a subtração esta à esquerda ou abaixo
12     return (calculo == TipoCalculo.Horizontal ? Direcao.Esquerda
13 :
14     Direcao.Abaixo);
return Direcao.None;

```

É verificado se a soma do ângulo de origem com a diferença entre os ângulos é igual ao ângulo de origem. Caso sejam iguais, significa que a direção ao ângulo de destino está à direita ou acima, de acordo com o tipo do cálculo que está sendo efetuado.

A direção diagonal é calculada de forma diferente, pois, conforme comentado anteriormente é preciso fazer uma proporcionalização (Quadro 24).

Quadro 24 - Cálculo da direção diagonal

```

01 var altural =
    EsqueletoCalculo.DistanciaEntrePontos(juntas1.FirstOrDefault(j
=> j.TipoJunta == JointType.Spine), centro1);
02 var altura2 =
    EsqueletoCalculo.DistanciaEntrePontos(juntas2.FirstOrDefault(j
=> j.TipoJunta == JointType.Spine), centro2);
03 var ratio1 = EsqueletoCalculo.DistanciaEntrePontos(centro1, j1) /
    altural;
04 var ratio2 = EsqueletoCalculo.DistanciaEntrePontos(centro2, j2) /
    altura2;
05 var pontuacaoProximidade = (100 / ratio1) * ratio2;
    pontuacaoProximidade = (pontuacaoProximidade > 100) ? 100 -
    (pontuacaoProximidade - 100) :
06 pontuacaoProximidade;
    var direcaoDiagonal = (ratio1 > ratio2) ? Direcao.Afastar :
    Direcao.Aproximar;

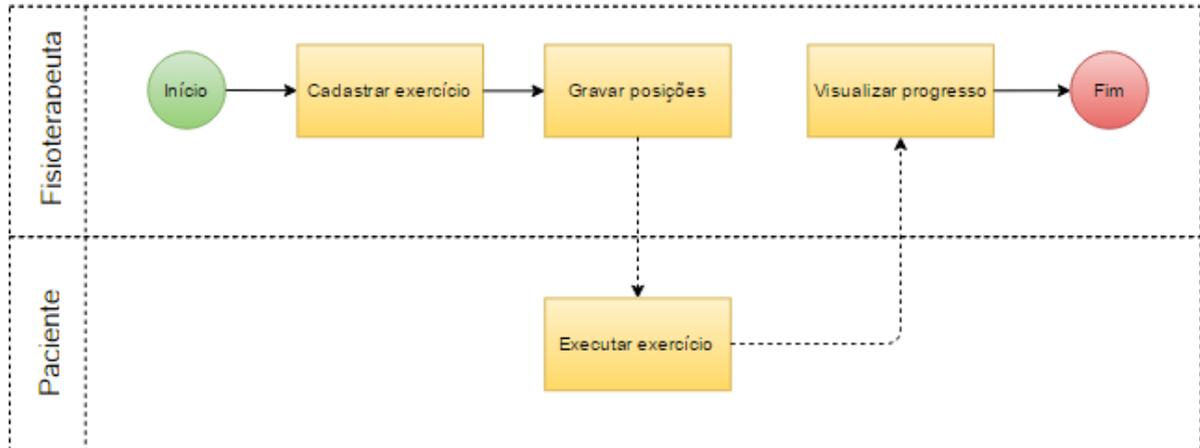
```

Primeiramente é obtida a distância entre as articulações do centro do ombro e a espinha. Posteriormente é obtida a distância entre o centro do ombro e a articulação que está sendo validada. Com estas distâncias pode-se efetuar o cálculo de proporcionalização e determinar se o paciente deve aproximar ou afastar para alcançar a posição desejada.

### 3.3.6 Operacionalidade da implementação

Nesta seção são apresentadas as funcionalidades destacadas nos casos de uso contidos na seção 3.2.1. A aplicação possui basicamente quatro etapas (cadastrar exercício, gravar posições, executar exercício, visualizar progresso), conforme demonstrado na Figura 17.

Figura 17 - Fluxo de atividades da aplicação



Os cadastros de médicos e pacientes estão descritos no apêndice C. A primeira etapa a ser realizada na aplicação é o cadastro do exercício, onde o fisioterapeuta informa os dados e marca quais articulações serão avaliadas. Posteriormente, na segunda etapa, este deve gravar as posições do exercício, que serão executas pelo paciente. Na terceira etapa, o paciente executa os exercícios repetindo as posições gravadas pelo fisioterapeuta. Finalmente, na quarta etapa, o fisioterapeuta visualiza a execução do exercício pelo paciente. Na tela de cadastro/edição do exercício deve-se informar o nome do exercício, médico, paciente e as articulações a serem validadas, conforme mostra a Figura 18. Pode-se também informar uma descrição para fins de orientação.

Figura 18 - Cadastrar/editar exercício

FisioK Pacientes Exercícios medico.01 Sair

## Editar Exercício

**Nome**

**Médico**  **Paciente**

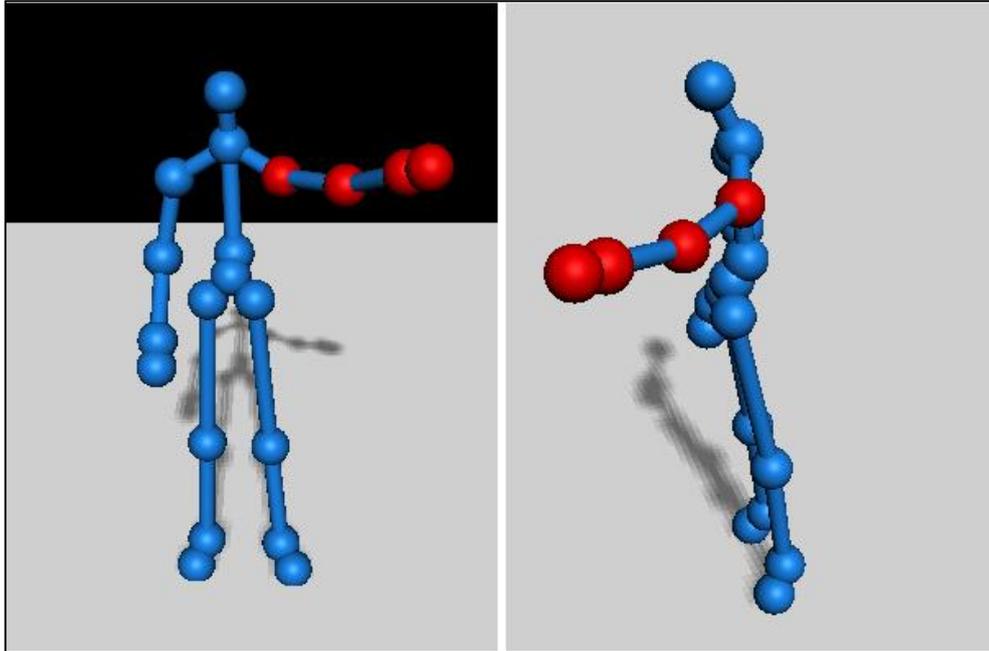
**Descrição**

**Juntas a serem validadas**



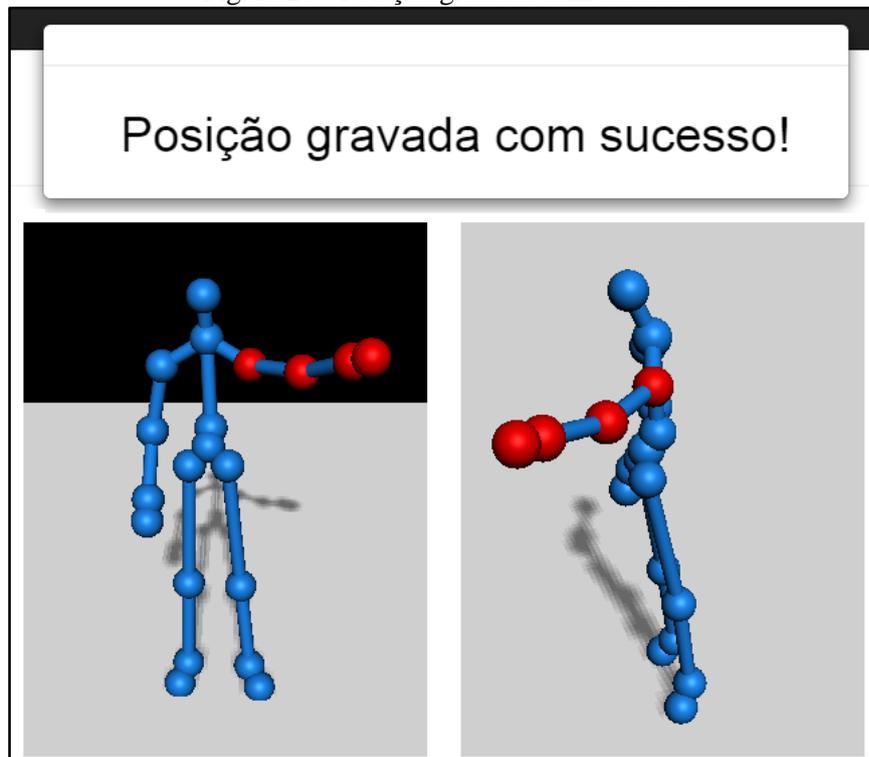
Após cadastrar o exercício e as articulações a serem validadas estarem devidamente marcadas, o fisioterapeuta deve clicar no botão gravar e posicionar-se corretamente em frente ao Kinect. A aplicação irá mostrar o esqueleto do fisioterapeuta no ambiente em dois ângulos diferentes, conforme mostra a Figura 19.

Figura 19 - Gravação da posição do exercício a ser repetida pelo usuário



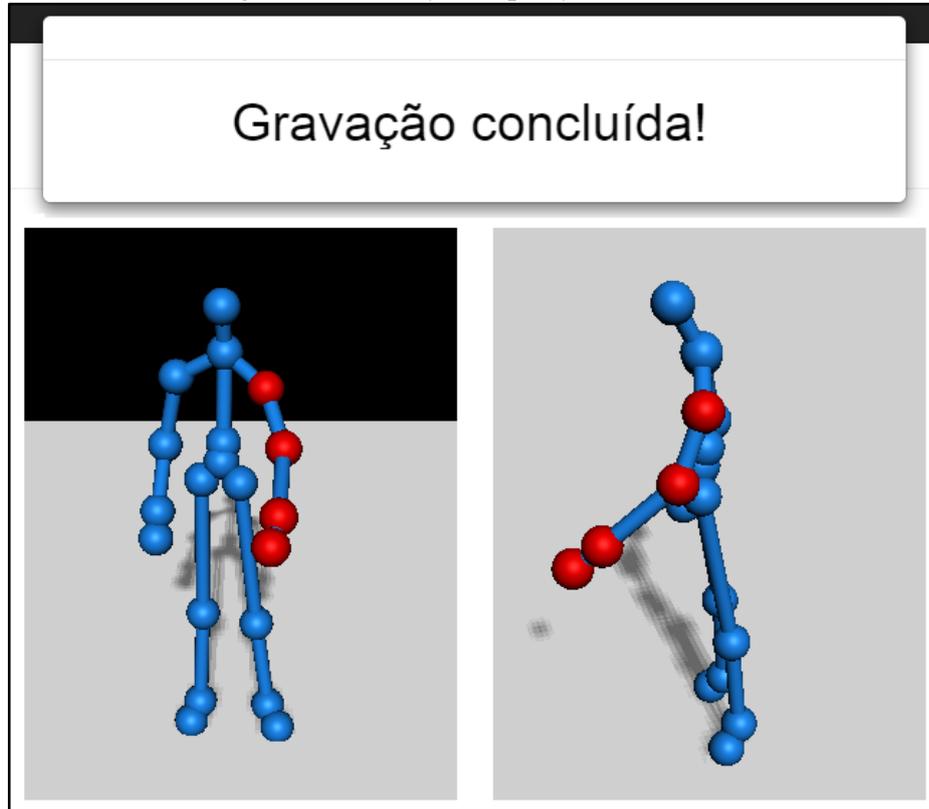
Para salvar a posição o fisioterapeuta precisa falar o comando “*SAVE*”. Ao falar o comando “*SAVE*” a aplicação irá salvar a posição atual do esqueleto como uma nova posição do exercício que deverá ser executado pelo usuário. Se a ação de inclusão da posição for realizada com sucesso, o fisioterapeuta receberá uma mensagem de confirmação, conforme mostra a Figura 20.

Figura 20 - Posição gravada com sucesso



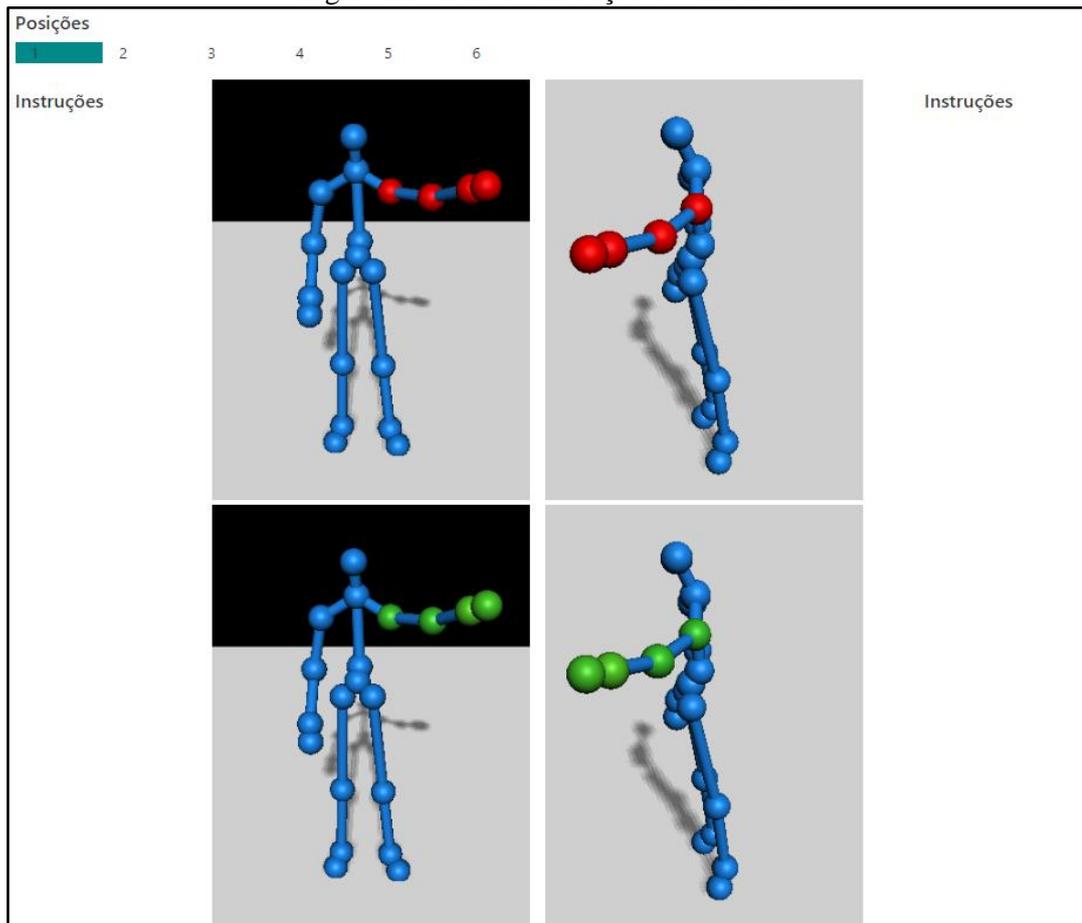
O fisioterapeuta poderá gravar quantas posições forem necessárias para cada exercício. Para concluir a gravação dos exercícios, ele deve falar o comando “*FINISH*” para que a aplicação finalize a conexão com o Kinect. Ao finalizar a gravação dos movimentos a aplicação irá notificar o fisioterapeuta do término da gravação, conforme mostra a Figura 21.

Figura 21 - Gravação de posições concluída



Após a conclusão da gravação do exercício, ele poderá ser executado pelo paciente. Para isto, o paciente deve clicar no botão executar ► e posicionar-se corretamente em frente do Kinect. A aplicação então irá mostrar dois pares de esqueletos com dois ângulos diferentes, sendo os superiores a posição previamente gravada pelo fisioterapeuta e os inferiores o esqueleto atual do paciente. Na parte superior da tela são mostradas todas posições a serem realizadas, sendo que a posição atual é destaca na cor azul. Quando necessário, nas laterais esquerda e direita são apresentadas as instruções dos membros dos respectivos lados, conforme mostra a Figura 22.

Figura 22 - Tela de execução do exercício



Como o paciente está posicionado de acordo com a posição cadastrada, nenhuma instrução é apresentada. Para que o paciente passe para a próxima posição, ele deve falar o comando “*NEXT*”. A aplicação irá posicionar na próxima posição e notificar o paciente conforme exibe a Figura 23.

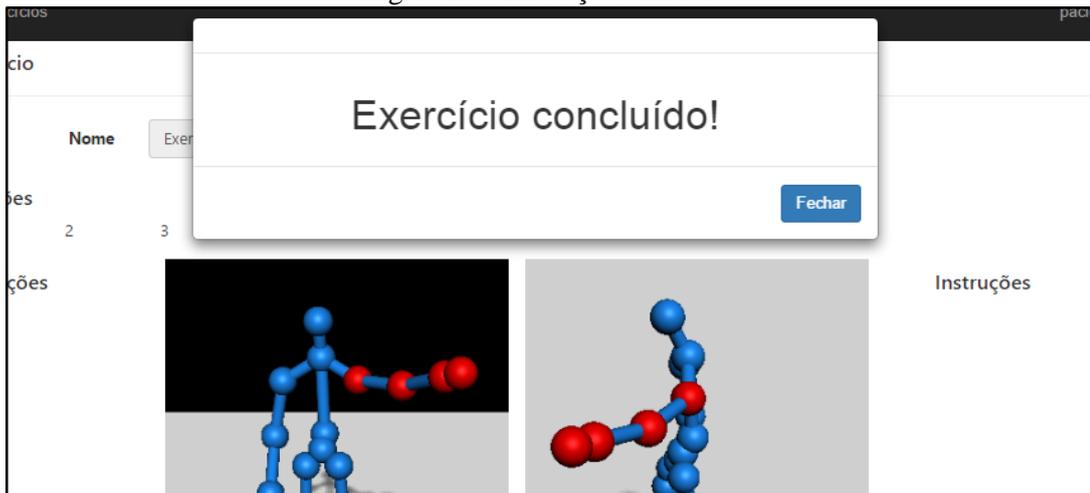
Figura 23 - Próxima posição selecionada



Quando o paciente tiver um percentual de acerto inferior à 95%, a aplicação irá apresentar nas laterais esquerda e direita instruções para que o paciente consiga alcançar o posicionamento correto. Para juntas do lado direito do corpo, as instruções serão exibidas no lado direito e para juntas do lado esquerdo, as instruções serão exibidas no lado esquerdo.

O exercício será finalizado quando o paciente efetuar a execução de todas as posições do exercício ou falar o comando “*FINISH*” em qualquer posição. Ao fazer isto, a aplicação irá finalizar a execução do exercício e notificar o paciente, conforme mostra a Figura 24.

Figura 24 - Execução concluída



Após a conclusão da execução do exercício pelo paciente, o desempenho da execução pode ser consultado pelo botão progresso 🔄 no exercício, conforme apresenta a Figura 25.

Figura 25 - Avaliação de desempenho do exercício

Junta	Avaliação
Ombro direito	99%
Cotovelo direito	99%
Punho direito	99%
Mão direita	99%

**Executado**

Na tela de avaliação do exercício, os esqueletos superiores são da posição gravada pelo fisioterapeuta e os inferiores são da posição executada pelo paciente. Fisioterapeuta e paciente podem navegar pelas posições através do *grid* na lateral esquerda, observando nos esqueletos

inferiores o posicionamento do paciente e a avaliação feita pela aplicação no *grid* da lateral direita. É importante ressaltar que somente são apresentadas as avaliações das juntas marcadas para avaliação.

### 3.4 RESULTADOS E DISCUSSÕES

Nesta seção são apresentados os experimentos feitos com a aplicação. A seção 3.4.1 apresenta o teste de representação do esqueleto do usuário no ambiente virtual da aplicação. A seção 3.4.2 mostra o teste realizado para verificar se orientações indicadas para os pacientes durante a execução dos exercícios estão corretas. A seção 3.4.3 mostra o teste de validação do percentual de acerto conforme posicionamento do paciente em relação à posição gravada pelo fisioterapeuta. A seção 3.4.4 apresenta o teste de validação do percentual de acerto com usuários de diferentes estaturas. Por fim, a seção 3.4.5 detalha o teste de usabilidade feito na aplicação.

#### 3.4.1 Experimento 01: representação do esqueleto no ambiente virtual

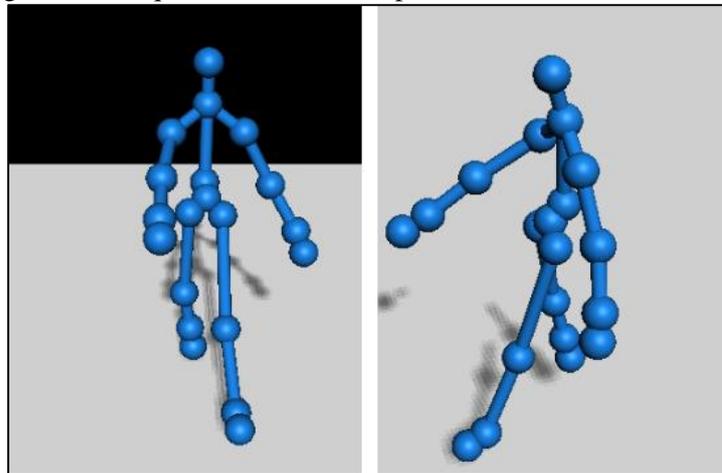
Este experimento tem como finalidade verificar se o posicionamento do esqueleto do usuário é representado corretamente no ambiente virtual. Verificou-se que o Kinect deve estar posicionado em um ângulo de 90 graus em relação ao usuário, em uma altura entre 1,20 e 1,50 metro e o usuário aproximadamente a 3 metros de distância. Para demonstrar a representação visual foram tiradas duas fotos com os mesmos ângulos que a aplicação representa o esqueleto no ambiente virtual. A Figura 26 mostra o posicionamento do usuário no ambiente real.

Figura 26 - Posicionamento usuário



A partir dos dados capturados pelo Kinect do usuário no ambiente real (Figura 26), a aplicação gera a representação do esqueleto do usuário no ambiente virtual, conforme pode ser visto na Figura 27.

Figura 27 - Esqueleto do usuário representado no ambiente virtual

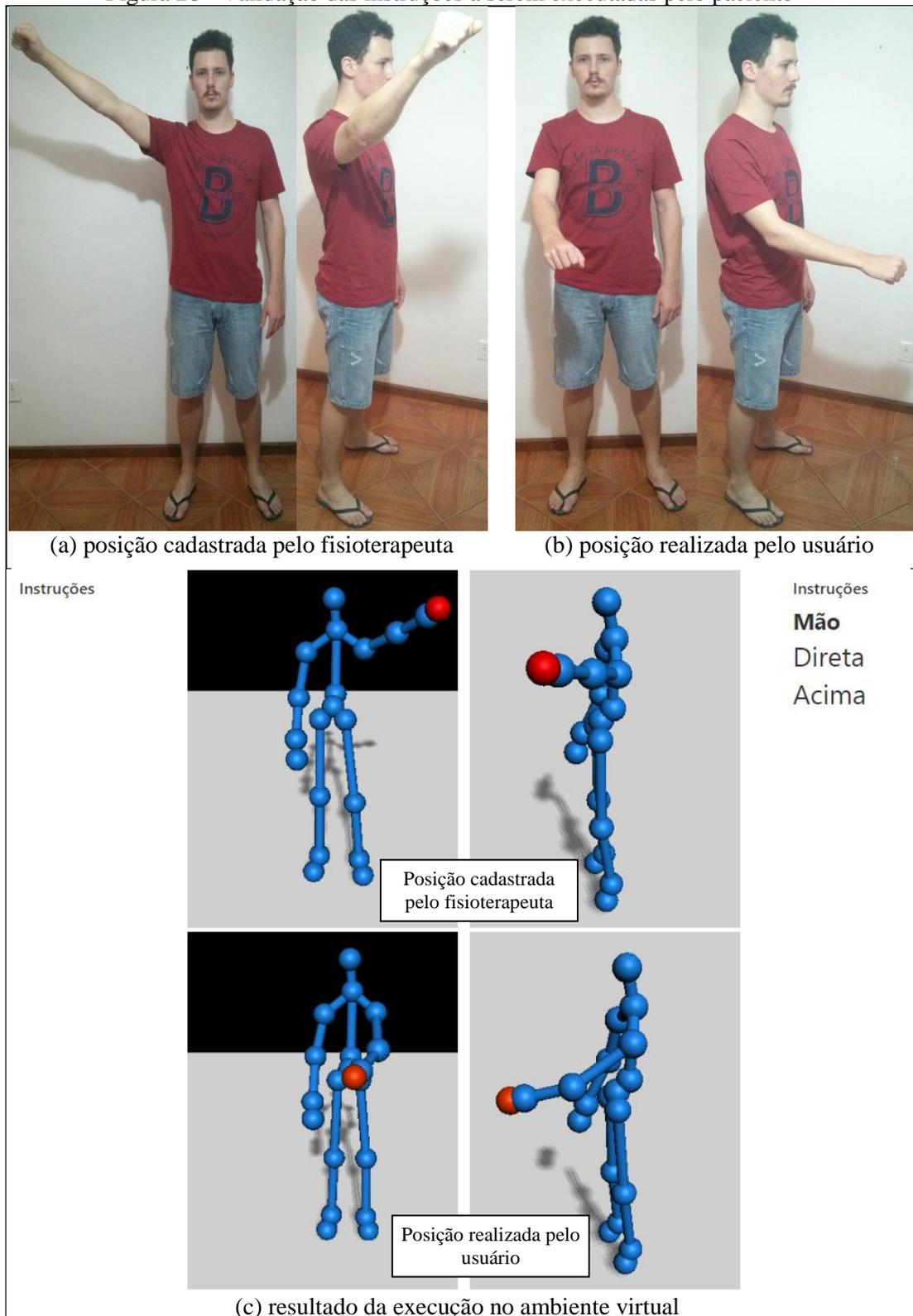


Observando o resultado alcançado na Figura 27, percebe-se que a aplicação representou o esqueleto do usuário corretamente conforme estava no ambiente real. O ambiente virtual está espelhado, para facilitar a interpretação e a execução do exercício pelo usuário.

#### 3.4.2 Experimento 02: validação das instruções dadas ao paciente

Este experimento tem como objetivo verificar se a aplicação estava orientando de forma correta os movimentos que usuário deve realizar para alcançar o posicionamento cadastrado pelo fisioterapeuta. O item (a) da Figura 28 mostra o posicionamento cadastrado pelo fisioterapeuta e, que o paciente deve atingir. No item (b), tem-se a posição realizada pelo usuário ao executar o exercício e, no lado direito do item (c) estão indicadas as instruções que o usuário deve realizar para alcançar o posicionamento correto.

Figura 28 - Validação das instruções a serem executadas pelo paciente

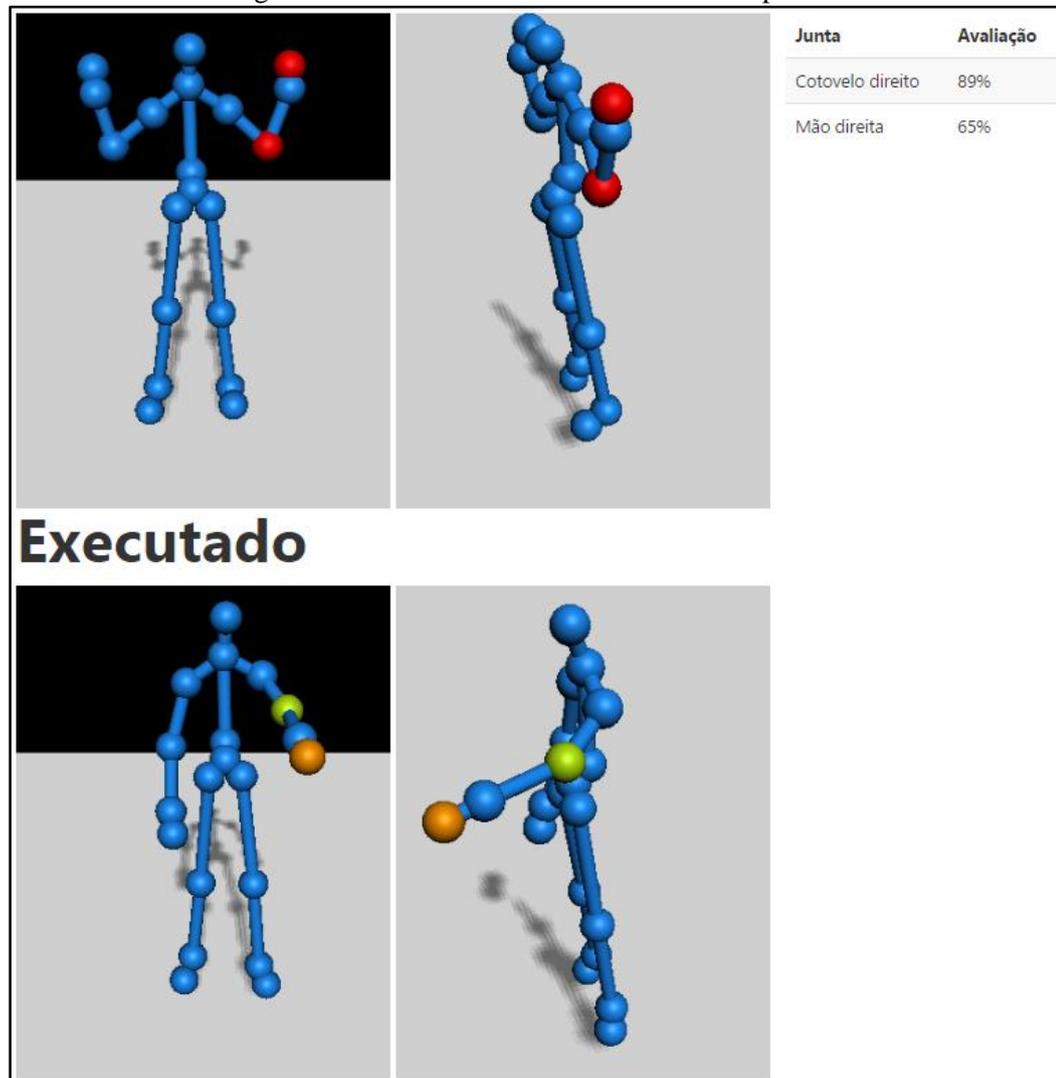


Considerando o efeito de espelhamento, pode-se observar que a aplicação instrui corretamente o paciente através das instruções Direita e Acima, pois caso o paciente movimente sua mão direita para acima e à direita atingirá a posição correta do exercício.

### 3.4.3 Experimento 03: validação do percentual de acerto

Este experimento tem por objetivo validar se o percentual de acerto indicado pela aplicação é condizente com a posição desempenhada pelo paciente em relação à posição gravada pelo fisioterapeuta. A Figura 29 mostra a avaliação entre duas posições.

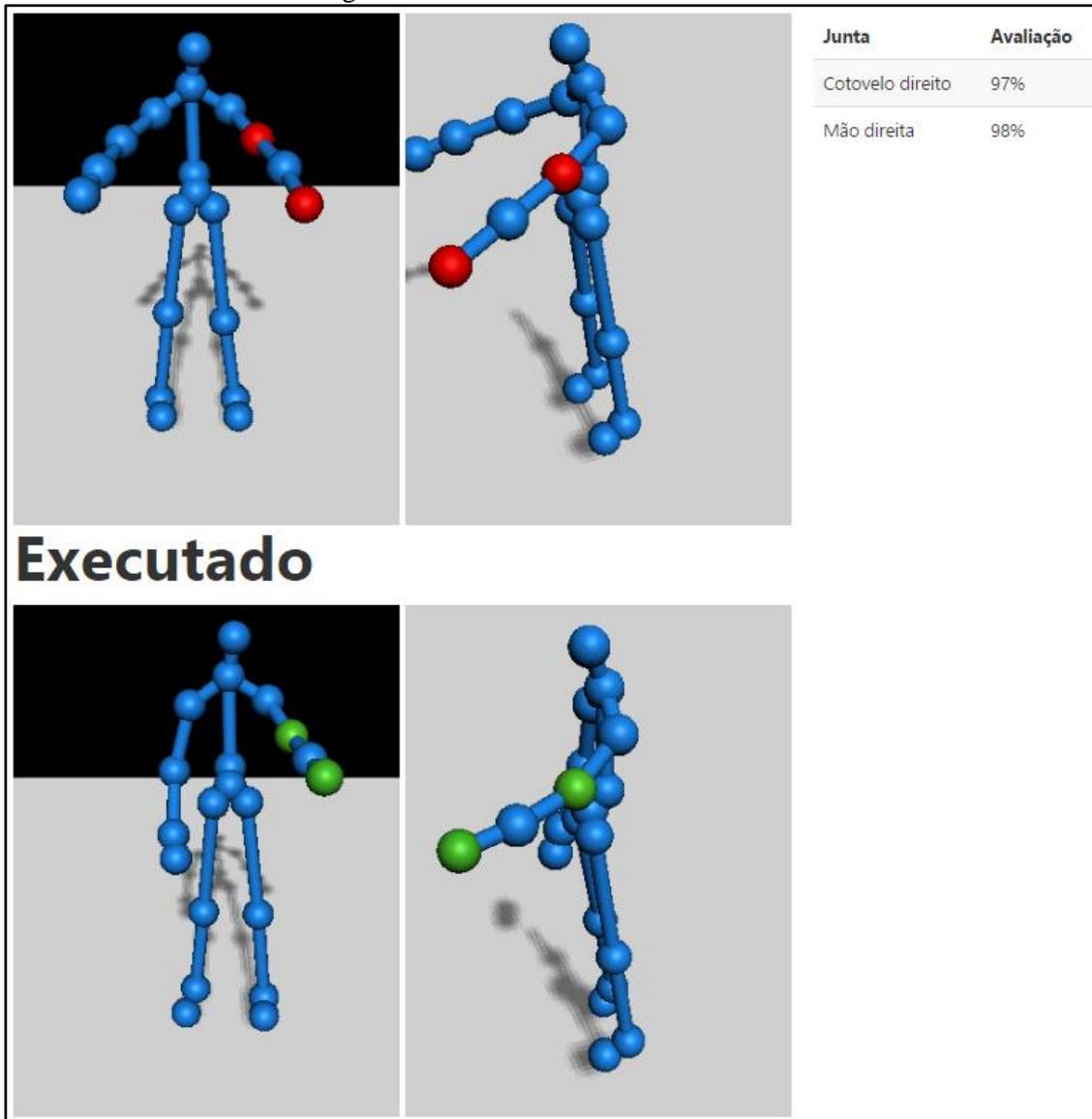
Figura 29 - Percentual de acerto abaixo do esperado



A partir da Figura 29, pode-se perceber que o cotovelo direito teve 89% de acerto, enquanto a mão direita teve 65%. O cotovelo direito do paciente está um pouco à frente em relação ao que o fisioterapeuta cadastrou no exercício. A altura está correta (verde), logo teve um percentual de acerto muito perto dos 100%. A mão direita do paciente está muito à frente e abaixo em relação à do fisioterapeuta (laranja), por isso teve um percentual de 65%.

A Figura 30 mostra uma avaliação entre duas posições, onde o usuário atingiu a posição correta, aumentando o percentual de acerto/repetição.

Figura 30 – Percentual de acerto elevado



Na Figura 30 pode-se perceber que o cotovelo e a mão direita tiveram 97% e 98% de acerto respectivamente. Pode-se observar que o braço não está totalmente estendido, fazendo com que o cotovelo fique um pouco mais para fora do que deveria em relação à posição do fisioterapeuta, obtendo assim um percentual menor do que à mão a qual está corretamente posicionada. A aplicação não irá mostrar 100% de acerto devido a instabilidade do Kinect em relação ao sensor que captura o esqueleto, pois o mesmo sempre sofre pequenas alterações nas coordenadas, fazendo com que o cálculo nunca chegue a 100%.

#### 3.4.4 Experimento 04: validação do percentual de acerto entre diferentes estaturas

Este experimento tem por objetivo validar se a aplicação verifica corretamente o percentual de acerto independentemente da diferença de estatura entre usuários, pois paciente e fisioterapeuta podem ter estaturas diferentes. Foram utilizados dois usuários, onde a usuária

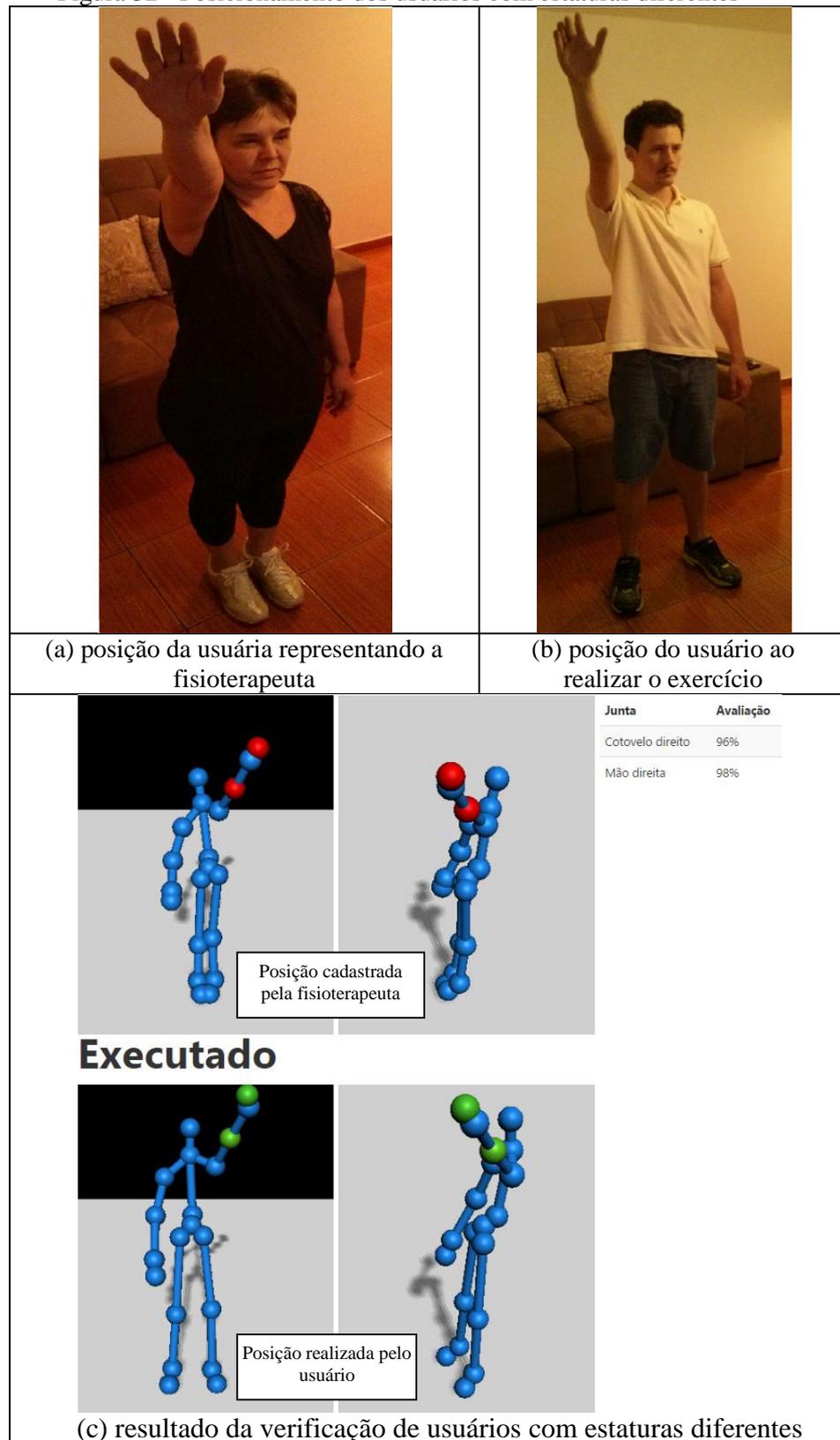
da esquerda possui a altura de 1,53 metro (um metro e cinquenta e três centímetros) e o usuário da direita possui 1,90 metro (um metro e noventa centímetros), conforme mostra a Figura 31.

Figura 31 - Usuários de diferentes estaturas



Para realizar o teste, foi cadastrado um exercício onde apenas a mão e cotovelo direito foram marcados para validação, e então, gravou-se apenas uma posição com base na usuária da esquerda (representando a fisioterapeuta - Figura 32 item (a)), sendo repetida posteriormente pelo usuário da direita (Figura 32 item (b)). O resultado considerando as diferentes estaturas pode ser visto na Figura 32 item (c).

Figura 32 - Posicionamento dos usuários com estaturas diferentes



Conforme pode ser visto na Figura 32, a aplicação validou corretamente a repetição da posição, obtendo um percentual de acerto de 96% e 98% para cotovelo e mão direita, respectivamente.

### 3.4.5 Experimento 05: usabilidade da aplicação

O experimento de usabilidade foi realizado com um fisioterapeuta e oito pacientes para avaliar eficiência e facilidade de uso da aplicação.

#### 3.4.5.1 Metodologia

Os experimentos foram realizados durante o mês de novembro por meio de testes individuais. Para o usuário fisioterapeuta foi disponibilizado um questionário de perfil, uma lista de tarefas a serem executadas e um questionário de usabilidade com questões referentes ao papel do fisioterapeuta da aplicação. Para usuários comuns, foram disponibilizados questionários de perfil e um questionário de usabilidade referente à usabilidade do papel paciente da aplicação. Os questionários disponibilizados aos usuários estão disponíveis no Apêndice D.

#### 3.4.5.2 Aplicação teste com o fisioterapeuta

Primeiramente o usuário fisioterapeuta foi orientado a preencher o questionário de perfil. Após o preenchimento do questionário de perfil, ele foi orientado sobre o objetivo do teste, da aplicação e de suas funcionalidades. A lista de tarefas foi composta de cinco tarefas que teve por objetivo contemplar todos objetivos e funcionalidades da aplicação. Ao finalizar cada tarefa, foi solicitado que o fisioterapeuta informasse se esta foi executada, caso tenha sido identificado algum problema ou se havia alguma observação a ser feita.

O questionário de usabilidade do fisioterapeuta foi composto por quinze perguntas, sendo quatorze fechadas e uma aberta. Para uma questão das questões fechadas foi disponibilizada a opção de informar uma observação. As perguntas tiveram por objetivo obter as impressões do fisioterapeuta sobre a usabilidade em geral da aplicação e as funcionalidades disponibilizadas ao papel fisioterapeuta. O perfil do fisioterapeuta foi contabilizado junto com os demais usuários e são apresentados na seção 3.4.5.3.1. Os resultados referentes à lista de tarefas desempenhadas e a usabilidade do fisioterapeuta são respectivamente apresentados nas seções 3.4.5.2.1 e 3.4.5.2.2.

##### 3.4.5.2.1 Análise dos resultados da lista de tarefas do fisioterapeuta

No Quadro 25 são apresentados os resultados quanto às questões envolvendo os processos desempenhados pelo fisioterapeuta na aplicação.

Quadro 25 - Respostas quanto à lista de tarefas

Perguntas/Respostas	Executou?
Cadastrar paciente	Sim
Cadastrar exercício	Sim
Marcar articulações	Sim
Gravar posições	Sim
Visualizar execução paciente	Sim

Considerando os resultados das questões da lista de tarefas exibidas no Quadro 25, percebeu-se que todas tarefas foram desempenhadas, o que indica a simplicidade e facilidade de uso da aplicação. Isto também indica que é possível executar todas funcionalidades no experimento. Não houve nenhuma observação para perguntas do questionário. O profissional fisioterapeuta observou fora das perguntas do questionário que, conforme sua experiência de trabalho, grande parte dos pacientes irá necessitar de uma pessoa para comandar a aplicação e orientar em seu uso. Este ainda ressaltou da importância de que a aplicação permite auxiliar fisicamente o paciente durante a execução dos exercícios, sem este ser adicionado à cena da aplicação. A aplicação apenas mostra o esqueleto do primeiro usuário capturado pelo Kinect, para que caso necessário o fisioterapeuta possa auxiliar o paciente durante a execução dos exercícios, desde que este não obstrua o ângulo de captura do Kinect.

#### 3.4.5.2.2 Análise resultados usabilidade fisioterapeuta

Após análise dos resultados da lista de tarefas do fisioterapeuta, foram analisados os resultados obtidos do questionário de usabilidade. Os resultados constam no Quadro 26.

Quadro 26 - Respostas do fisioterapeuta em relação ao questionário de usabilidade

Perguntas	Respostas
De modo geral, você achou a aplicação intuitiva e de fácil utilização?	Sim
Como você avalia a representação gráfica do paciente dentro da aplicação?	Muito bom
Como você avalia a funcionalidade para informar quais articulações serão avaliadas pela aplicação?	Muito bom
Você teve dificuldade para cadastrar o exercício?	Não
Você considera importante a avaliação do percentual de acerto feita pela aplicação?	Sim
Você considera importante visualizar o desempenho do paciente após a realização do exercício?	Sim
Você precisaria aprender muitas coisas antes de usar o aplicativo?	Não
Você acha que seria necessário o apoio de uma pessoa técnica para utilizar a aplicação?	Não
Foi fácil lembrar como fazer as coisas no aplicativo?	Sim
Você utilizaria esta aplicação no seu trabalho?	Sim
A aplicação aumentaria a eficácia do seu trabalho?	Não
Você acha que os pacientes se sentiriam mais motivados utilizando a aplicação desenvolvida nas sessões de fisioterapia?	Sim
Você recomendaria este aplicativo para outras pessoas?	Sim

Qual é sua avaliação do protótipo?	Muito bom
Qual foi sua maior dificuldade utilizando o protótipo?	Nenhuma

O fisioterapeuta que realizou o teste, considerou a aplicação intuitiva e de fácil utilização, não tendo nenhuma dificuldade na utilização, visto que não precisa aprender muitas coisas para operacionaiiza-la. Respondeu também que não acha necessário o apoio de uma pessoa técnica para operar a aplicação, e que é fácil lembrar como efetuar as tarefas do fisioterapeuta nesta. Este não teve nenhuma dificuldade para cadastrar um novo paciente, assim como cadastrar um novo exercício e marcar as articulações a serem validadas pela aplicação. Também destacou como ótima e objetiva a forma de como o esqueleto do usuário é representado dentro da aplicação.

O fisioterapeuta considerou importante a avaliação do percentual de acerto feita pela aplicação, e que acha que os pacientes se sentiriam mais motivados utilizando a aplicação durante as sessões fisioterapêuticas. Ressaltou também sobre a importância da aplicação permitir tanto ao fisioterapeuta quanto ao paciente visualizar o desempenho após a realização dos exercícios.

Por fim, respondeu que recomendaria para outras pessoas, mas que este aplicativo não aumentaria a eficácia do seu trabalho. Mesmo assim, avaliou a aplicação como muito boa, ressaltando que esta realmente auxiliaria nos tratamentos fisioterapêuticos.

#### 3.4.5.3 Aplicação do teste de usabilidade com usuários comuns

Primeiramente os usuários foram orientados a preencherem o questionário de perfil, e, após o preenchimento, eles foram orientados sobre o objetivo do teste e do objetivo da aplicação. Todos usuários executaram o mesmo exercício, que continha por sua vez cinco posições. O questionário de usabilidade foi disponibilizado após a conclusão do exercício por parte do usuário.

O questionário de usabilidade dos usuários foi composto de quinze perguntas, sendo treze fechadas e duas abertas. As perguntas tiveram por objetivo obter as impressões dos usuários sobre a usabilidade e entendimento em geral da aplicação. Os perfis dos usuários são apresentados na seção 3.4.5.3.1 e os resultados do questionário de usabilidade dos usuários são apresentados na seção 3.4.5.3.2.

##### 3.4.5.3.1 Análise e interpretação dos dados coletados dos questionários do usuário

A primeira análise realizada foi feita com base nos dados do questionário de perfil de usuário. No Quadro 27 são exibidos os perfis dos usuários envolvidos nos testes de usabilidade.

Quadro 27 - Perfis dos usuários envolvidos nos testes de usabilidade

Sexo	62,5% Masculino 37,5% Feminino
Idade	12,5% 15 à 10 anos 25% 35 ou mais 62,5% 20 à 25 anos
Nível de escolaridade	25% Fundamental completo 75% Superior incompleto
Frequência uso computador	12,5% Nunca utilizei 12,5% Às vezes 75% Frequentemente
Familiaridade Kinect	12,5% Nunca ouvi falar 37,5 Conheço, mas nunca utilizei 50% Já utilizei

Com as informações de perfil dos usuários que se voluntariaram para os testes, percebe-se que grande maioria dos usuários utiliza o computador frequentemente, e também ou conhecia ou já utilizou o sensor Kinect, o que colaborou para o entendimento do funcionamento da aplicação e de como utilizá-la.

#### 3.4.5.3.2 Análise resultados usabilidade dos usuários

Após orientar os usuários sobre o objetivo da aplicação, estes foram submetidos à execução de um único exercício pré-definido, onde deveriam repeti-lo conforme orientações da aplicação. Após a execução do exercício cada usuário respondeu o questionário de usabilidade. Estes resultados constam no Quadro 28.

Quadro 28 - Respostas dos usuários em relação ao questionário de usabilidade

Perguntas	Respostas
Dos exercícios feitos, quais você conseguiu executar de forma correta?	75% Todos 25% Maioria
De modo geral, você achou a aplicação intuitiva e fácil utilização?	87,5% Sim 12,5% Não
Como você avalia a representação visual de cada posição que deve ser repetida?	75% Bom 25% Muito bom
Como você avalia a forma que a aplicação aponta se o exercício foi feito corretamente?	37,5% Bom 25% Regular 25% Muito bom 12,5% Insatisfatório
Você considera importante que a aplicação dê orientações de como executar o exercício?	100% Sim
Você conseguiu entender as orientações dadas para alcançar o objetivo do exercício?	100% Sim
Quanto as mensagens da aplicação?	62,5% poderia ser mais fácil – tive dúvidas em dados momentos 37,5% Fáceis de entender
Senti-me confortável usando este sistema	100% Sim
Às vezes eu não sei o que fazer com este aplicativo	75% Não 25% Sim
Você precisaria do apoio de uma pessoa para usar este aplicativo?	62,5% Sim 37,5% Não
O aplicativo em algum momento parou de funcionar inesperadamente?	75% Não 25% Sim
Você recomendaria este aplicativo para outras pessoas?	100% Sim
Qual é sua avaliação do protótipo?	50% Bom 37,5% Muito bom 12,5% Regular
Você gostou de utilizar o Kinect para realizar os exercícios?	Sim 100%

Mais de sessenta por cento dos usuários julgaram que as orientações dadas pela aplicação poderiam ser mais fáceis, e que tiveram dúvidas em alguns momentos durante a execução do exercício. A dificuldade de entender as orientações dadas pela aplicação foi muito comentada, tanto na questão fechada referente às mensagens quanto na questão aberta sobre qual a maior dificuldade que o usuário teve durante a utilização da aplicação. Outra dificuldade comentada por dois usuários, foram os comandos de voz necessários para utilizar a aplicação, os quais devem ser ditos no idioma inglês. Cerca de sessenta por cento dos usuários julgaram necessário o apoio de uma pessoa para utilizar a aplicação.

Embora a maioria dos usuários tenha tido dificuldades para entender as orientações dadas pela aplicação, 75% (setenta e cinco por cento) dos usuários conseguiram executar de forma correta o exercício dado. Isto ocorreu devido à representação visual de como o exercício deve ser executado, com dois diferentes ângulos de câmera. O *feedback* em tempo real, onde a

aplicação faz um degrade da cor verde para vermelha conforme o percentual de acerto, também auxiliou para que o usuário soubesse se estava posicionado de forma correta durante a repetição da posição do exercício. Com dois dos oito usuários, a aplicação parou de responder inesperadamente durante a execução do exercício, a qual foi reestabelecida rapidamente.

Todos os usuários sentiram-se confortáveis utilizando a aplicação e gostaram de utilizar o sensor Kinect. Além disso, recomendariam o uso da aplicação para outras pessoas. Quase noventa por cento dos usuários julgaram a aplicação como boa ou muito boa, e os 10% (dez) por cento restantes julgaram como regular.

## 4 CONCLUSÕES

O principal objetivo do trabalho, que era fazer uma reformulação do trabalho feito por Pilon (2013) foi alcançado. Pilon (2013) teve como objetivo além de desenvolver a aplicação para auxiliar nos tratamentos fisioterapêuticos, validar se esta seria viável nos tratamentos de reabilitação fisioterapêutica.

O objetivo de desenvolver uma aplicação *web* foi atendido, bastando ao usuário acessar a aplicação com um navegador recente que suporte os recursos do HTML5. Através da lista de tarefas e do questionário de usabilidade aplicados ao fisioterapeuta, é possível afirmar que o objetivo de disponibilizar um mecanismo para que este pudesse cadastrar exercícios específicos utilizando o Microsoft Kinect foi alcançado e que é intuitivo e de fácil utilização.

Pela lista de tarefas e questionário de usabilidade aplicados ao fisioterapeuta, também é possível afirmar que os objetivos de disponibilizar um mecanismo que valide de forma visual se os exercícios foram executados corretamente e o objetivo de armazenar os dados gerados nas sessões fisioterapêuticas e disponibilizá-los para acompanhamento do paciente e fisioterapeuta foram alcançados. A aplicação também calcula o percentual de acerto para cada articulação marcada pelo fisioterapeuta, e exibe esta avaliação tanto textual quanto visual na tela de consulta do progresso.

O objetivo de orientar o paciente de como executar corretamente o exercício também foi atendido, este pode ser visto pelo questionário de usabilidade dos usuários. Ainda através deste questionário, é possível verificar que a aplicação efetua uma avaliação do percentual de acerto durante a execução do exercício, de modo que o paciente verifique de forma visual seu desempenho.

Vale ressaltar que a representação gráfica do esqueleto do usuário dentro da aplicação foi aprimorada, onde anteriormente esta era representada num modelo de duas dimensões, o que não dava ao paciente o entendimento correto de como este deveria executar o exercício, por não ter a percepção da posição exata que este deveria fazer no espaço. Para aprimorar a representação, montou-se dois ambientes em três dimensões, sendo que um possui um ângulo de câmera frontal e outro na diagonal superior direita. Desta forma fisioterapeuta e paciente podem visualizar mais precisamente como a posição foi ou deve ser executada.

O mecanismo de cadastro de um novo exercício também foi aprimorado. Anteriormente no trabalho de Pilon (2013) era necessário falar, uma a uma, os nomes das articulações a serem marcadas, e posteriormente posicionar no espaço, uma a uma, a localização desta. Na aplicação

desenvolvida o fisioterapeuta marca quais articulações deseja validar no ato do cadastro de um novo exercício, e posteriormente grava várias posições de todas articulações de uma só vez.

Tanto usuários quanto o fisioterapeuta elogiaram a fácil utilização e entendimento da aplicação, com destaque ao *feedback* visual. Por fim, considerando os testes aplicados utilizando usuários comuns e um fisioterapeuta, considerou-se que a ferramenta é intuitiva e de fácil utilização. O sensor Kinect mostrou-se hábil na captura de movimentos para representá-los no ambiente virtual assim como efetuar cálculos de avaliações sob os dados obtidos a partir deste. Este ainda mostra algumas limitações quanto à imprecisão e variação de retorno dos dados conforme seu posicionamento.

#### 4.1 EXTENSÕES

Algumas das possíveis extensões para este trabalho são:

- a) criar *plugin* ou aplicação para permitir a utilização remota: atualmente para utilizar a aplicação é necessário ao usuário ter a aplicação e todo conjunto de bibliotecas e SDK's instalados em seu computador. Com a criação de um *plugin* ou outra aplicação bastaria este possuir somente um Kinect, seu respectivo *driver* e conexão com à internet;
- b) criar novas métricas de avaliação: já que atualmente a aplicação avalia posições estáticas. Uma opção por exemplo seria esta permitir o cadastro e avaliação de movimentos;
- c) permitir aos usuários navegar complementemente ou o máximo possível por comandos de voz: pois atualmente parte da operacionalidade da aplicação é feita através de teclado e mouse, como o cadastros, navegação, carregamento exercícios, entre outros;
- d) permitir comandos de voz em português: conforme testes de usabilidade, os comandos em inglês confundem os usuários e tornam a iteração com a aplicação menos natural;
- e) orientação de execução dos exercícios: conforme testes de usabilidade dos usuários, a orientação textual confundiu e não colaborou tanto quanto deveria. Um sugestão seria inserir instruções dentro da cena do esqueleto do paciente com as direções de cada articulação;
- f) validar usuários com membros amputados e/ou com outros tipos de deficiência: adaptar a aplicação para usuários com membros amputados ou portadores de alguma deficiência física ou mental.

## REFERÊNCIAS

- ARAUJO, Regina Borges de; KIRNER, Cláudio. Especificação e análise de um sistema distribuído de realidade virtual. In: XI SIMPÓSIO DE REDES DE COMPUTADORES, 14., 1996, Recife. **SBRC 96**. Recife: Sbc, 2012. p. 357 - 374.
- BIGGS, John. **Microsoft Launches Kinect SDK For Windows**. 2011. Disponível em: <<http://techcrunch.com/2011/06/16/microsoft-launches-kinect-windows-sdk/>>. Acesso em: 20 nov. 2015.
- BURDEA, Grigore C.; COIFFET, Philippe. **Virtual Reality Technology**. 2. ed. Nova Jersey, Us: John Wiley & Sons, 2003. 464 p.
- CAMPOS, Guilherme Pires. **Sistema para fisioterapia baseado na plataforma Kinect**. 2013. 68 f. Dissertação (Mestrado) - Curso de Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, Engenharia Eletrotécnica e de Computadores, Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2013.
- CATUHE, David. **Programming with the Kinect for Windows Software Development Kit**. Redmond, Washington: Microsoft Press, 2012. 207 p.
- CHANG, Yao-jen; CHEN, Shu-fang; HUANG, Jun-da. A Kinect-based system for physical rehabilitation: A pilot study for young adults with motor disabilities. **Research In Developmental Disabilities**, [s.l.], v. 32, n. 6, p.2566-2570, nov. 2011. Elsevier BV. DOI: 10.1016/j.ridd.2011.07.002. Disponível em: <<http://api.elsevier.com/content/article/PII:S0891422211002587?httpAccept=text/xml>>. Acesso em: 20 nov. 2015.
- COFFITO. **Definição de Fisioterapia e Áreas de Atuação**. 2012. Disponível em: <<http://www.coffito.org.br/site/index.php/fisioterapia/definicao.html>>. Acesso em: 23 nov. 2015.
- CRAWFORD, Stephanie. **How Microsoft Kinect Works**. 2010. Disponível em: <<http://electronics.howstuffworks.com/microsoft-kinect.htm>>. Acesso em: 13 jul. 2010.
- DANCHILLA, Brian. **Beginning WebGL for HTML5**. New York: Apress, 2012. 329 p.
- DIRKSEN, Jos. **Learning Three.js – the JavaScript 3D Library for WebGL: Create stunning 3D graphics in your browser using the Three.js JavaScript library**. 2. ed. Birmingham, United Kingdom: Packt Publishing Ltd, 2015. 397 p.
- GARDINER, M. Dena. **Manual de terapia por exercícios**. 4. ed. Santos: Santos, 1995.
- HEILIG, Morton. **Morton Heilig**. 1995. Disponível em: <<http://www.mortonheilig.com/InventorVR.html>>. Acesso em: 21 set. 2015.
- JINTRONIX. **Kinect Rehabilitation Software**. 2015. Disponível em: <<http://www.jintronix.com>>. Acesso em: 2 abr. 2015.
- KHOSHELHAM, Kourosh; ELBERINK, Sander Oude. Accuracy and Resolution of Kinect Depth Data for Indoor Mapping Applications. **Sensors**, Enschede, v. 12, n. 12, p.1437-1454, 1 fev. 2012. MDPI AG. DOI: 10.3390/s120201437. Disponível em: <<http://www.mdpi.com/1424-8220/12/2/1437>>. Acesso em: 5 nov. 2015.
- LOWENSOHN, Josh. **Timeline: A look back at Kinect's history**. 2011. Disponível em: <<http://www.cnet.com/news/timeline-a-look-back-at-kinects-history/>>. Acesso em: 5 nov. 2015.

- LUTZ, Justin. **Beyond The Game - 5 Innovative Uses For Kinect**. 2014. Disponível em: <<http://www.theprimacy.com/blog/beyond-the-game-5-innovative-uses-for-kinect>>. Acesso em: 5 nov. 2015.
- MACHADO, Liliane dos Santos et al. Serious games baseados em realidade virtual para educação médica. **Rev. Bras. Educ. Med.**, Rio de Janeiro, v. 35, n. 2, p.254-262, jun. 2011. FapUNIFESP (SciELO). DOI: 10.1590/s0100-55022011000200015. Disponível em: <[http://www.scielo.br/scielo.php?pid=S0100-55022011000200015&script=sci\\_arttext](http://www.scielo.br/scielo.php?pid=S0100-55022011000200015&script=sci_arttext)>. Acesso em: 20 nov. 2015.
- MELANI, Rodolfo Francisco Haltenhoff; SILVA, Ricarda Duarte da. A relação profissional-paciente: o entendimento e implicações legais que se estabelecem durante o tratamento ortodôntico. **Rev. Dent. Press Ortodon. Ortop. Facial**, Maringá, Brasil, v. 11, n. 6, p.104-113, 2006. FapUNIFESP (SciELO). DOI: 10.1590/s1415-54192006000600013. Disponível em: <[http://www.scielo.br/scielo.php?script=sci\\_arttext&pid=S1415-54192006000600013&lng=en&tlng=pt](http://www.scielo.br/scielo.php?script=sci_arttext&pid=S1415-54192006000600013&lng=en&tlng=pt)>. Acesso em: 5 nov. 2015.
- MILES, Rob. **Start Here! Learn the Kinect API**. Sebastopol, California: O'reilly Media Inc., 2012. 250 p.
- MONTEIRO, Carlos Bandeira de Mello. **Realidade Virtual na Paralisia Cerebral**. São Paulo: Plêiade, 2011. 220 p.
- PARTRIDGE, Cecilly. **Fisioterapia Neurológica**. São Paulo: Santos, 2006. 240 p.
- PILON, Felipe. **Ferramenta de Auxílio a Tratamentos Fisioterapêutico utilizando o Kinect**. 2013. 71 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Regional de Blumenau, Blumenau, 2013. Disponível em: <<http://dsc.inf.furb.br/tcc/index.php?cd=11&tcc=1519>>. Acesso em: 5 nov. 2015.
- RODRIGUES, Gessica Palhares; PORTO, Cristiane de Magalhães. Realidade Virtual: conceitos, evolução, dispositivos e aplicações. **Interfaces Científicas - Educação**, Aracaju, v. 1, n. 3, p.97-109, 25 jun. 2013. Universidade Tiradentes. DOI: 10.17564/2316-3828.2013v1n3p97-109.
- SÁ, Jobert G. P.. **Construindo uma DSL para reconhecimento de gestos utilizando Kinect**. 2011. 76 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade Federal de Pernambuco, Recife, 2011.
- SILVA, Luiz José Schirmer et al. Sistema de Reabilitação Fisioterapêutica baseado em Jogos com Interfaces Naturais. In: XI SIMPÓSIO BRASILEIRO DE JOGOS E ENTRETENIMENTO DIGITAL, 6., 2012, Brasília. **SBC - Proceedings of SBGames 2012**. Brasília: Sbc, 2012. p. 61 - 64.
- TANG, Jin et al. 2.5D Multi-View Gait Recognition Based on Point Cloud Registration. **Sensors**, [s.l.], v. 14, n. 4, p.6124-6143, 28 mar. 2014. MDPI AG. DOI: 10.3390/s140406124.
- TEIXEIRA, Ingrid Carvalho. **Jogo virtual controlado pelos sinais mioelétricos na recuperação de pacientes com lesão muscular nos membros superiores e/ou inferiores**. 2013. 81 f. TCC (Graduação) - Curso de Engenharia de Computação, Engenharia de Computação, Centro Universitário de Brasília, Brasília, 2013.
- VALLI, Alessandro. **Notes on Natural Interaction**. 2004. 88 f. Tese (Doutorado) - Curso de Computer Engineering, Mathematics And Computer Science, University Of Florence, Firenze, Italia, 2004. Disponível em: <<http://www.micc.unifi.it/publications/2004/Val04/>>. Acesso em: 20 nov. 2015.

## APÊNDICE A – Detalhamento dos casos de uso

Nesta seção são apresentados os detalhamentos dos casos de uso, com descrição, ator(es), pré-condições e cenário.

No caso de uso UC01: *Cadastrar médicos* o administrador pode cadastrar novos médicos para que estes tenham acesso à aplicação, podendo assim exercer seu papel. Detalhes sobre o caso de uso estão descritos no Quadro 29.

Quadro 29 - UC01 Cadastrar médicos

Número	01
Caso de Uso	Cadastrar médicos
Ator	Administrador
Pré-condições	
Cenário Principal	<ol style="list-style-type: none"> <li>1. Seleccionar aba Médicos;</li> <li>2. Clicar no botão Inserir;</li> <li>3. Informar nome e usuário de acesso;</li> <li>4. Clicar no botão Salvar;</li> <li>5. Redirecionamento para aba Médicos.</li> </ol>

O caso de uso UC02: *Cadastrar pacientes*, o qual é desempenhado tanto pelo ator Administrador quanto pelo papel Fisioterapeuta, permite cadastrar novos pacientes para que este possa ter acesso à aplicação e exercer seu papel. Detalhes sobre o caso de uso estão descritos no Quadro 30.

Quadro 30 - UC02 Cadastrar pacientes

Número	02
Caso de Uso	Cadastrar pacientes
Atores	Administrador e Fisioterapeuta
Pré-condições	
Cenário Principal	<ol style="list-style-type: none"> <li>1. Seleccionar aba Pacientes;</li> <li>2. Clicar no botão Inserir;</li> <li>3. Informar nome e usuário de acesso;</li> <li>4. Clicar no botão Salvar;</li> <li>5. Redirecionamento para aba Pacientes.</li> </ol>

O caso de uso UC03: *Manter exercícios*, o qual é desempenhado exclusivamente pelo Fisioterapeuta, permite inserir, editar e excluir exercícios. Detalhes sobre o caso de uso estão descritos no Quadro 31.

Quadro 31 - UC03 Manter exercícios

Número	03
Caso de Uso	Manter exercícios
Ator	Fisioterapeuta
Pré-condições	
Cenário Principal	<ol style="list-style-type: none"> <li>1. Selecionar aba Exercícios;</li> <li>2. Clicar no botão Inserir;</li> <li>3. Informar nome, paciente, observações e selecionar as juntas a serem avaliadas;</li> <li>4. Clicar no botão Salvar;</li> <li>5. Redirecionamento para aba Exercícios;</li> <li>6. Clicar no botão Editar ao lado do exercício anteriormente cadastrado;</li> <li>7. Alterar as informações desejadas;</li> <li>8. Clicar no botão Salvar;</li> <li>9. Redirecionamento para aba Exercícios;</li> <li>10. Clicar no botão Editar ao lado do exercício anteriormente cadastrado;</li> <li>11. Clicar no botão Excluir;</li> <li>12. Confirmar exclusão;</li> <li>13. Redirecionamento para aba Exercícios.</li> </ol>

O caso de uso UC04: Gravar exercícios permite ao Fisioterapeuta cadastrar as posições do exercício, as quais serão posteriormente executadas pelo Paciente. Detalhes sobre o caso de uso estão descritos no Quadro 32.

Quadro 32 - UC04 Gravar exercícios

Número	04
Caso de Uso	Gravar exercícios
Ator	Fisioterapeuta
Pré-condições	<ol style="list-style-type: none"> <li>1. Ter um exercício cadastrado;</li> <li>2. Estar com o Kinect conectado e posicionado corretamente;</li> <li>3. Estar posicionado corretamente em frente ao Kinect.</li> </ol>
Cenário Principal	<ol style="list-style-type: none"> <li>1. Selecionar o exercício desejado na aba Exercícios;</li> <li>2. Clicar no botão Gravar;</li> <li>3. Falar a palavra <i>SAVE</i> para salvar uma nova posição;</li> <li>4. Salvar quantas posições forem necessárias;</li> <li>5. Falar a palavra <i>FINISH</i> para encerrar o cadastro.</li> </ol>

O caso de uso UC05: Visualizar progresso exercício, desempenhado pelo Fisioterapeuta e Paciente permite a ambos visualizarem o progresso da execução do exercício, após este ter sido executado pelo Paciente. Detalhes sobre o caso de uso estão descritos no Quadro 33.

Quadro 33 - UC05 Visualizar progresso

Número	05
Caso de Uso	Visualizar progresso
Atores	Fisioterapeuta e Paciente
Pré-condições	<ol style="list-style-type: none"> <li>1. Ter um exercício cadastrado;</li> <li>2. Exercício já executado pelo paciente;</li> </ol>
Cenário Principal	<ol style="list-style-type: none"> <li>1. Clicar no botão Progresso do exercício desejado na aba Exercícios;</li> <li>2. Clicar nas posições ao lado esquerdo para visualizar a posição, execução do paciente e seu respectivo desempenho.</li> </ol>

O caso de uso UC06: Executar exercícios é desempenhado exclusivamente pelo Paciente, o qual permite ao mesmo carregar as posições a serem executadas, executá-las, e gravar os dados da execução no exercício. Detalhes sobre o caso de uso estão descritos no Quadro 34.

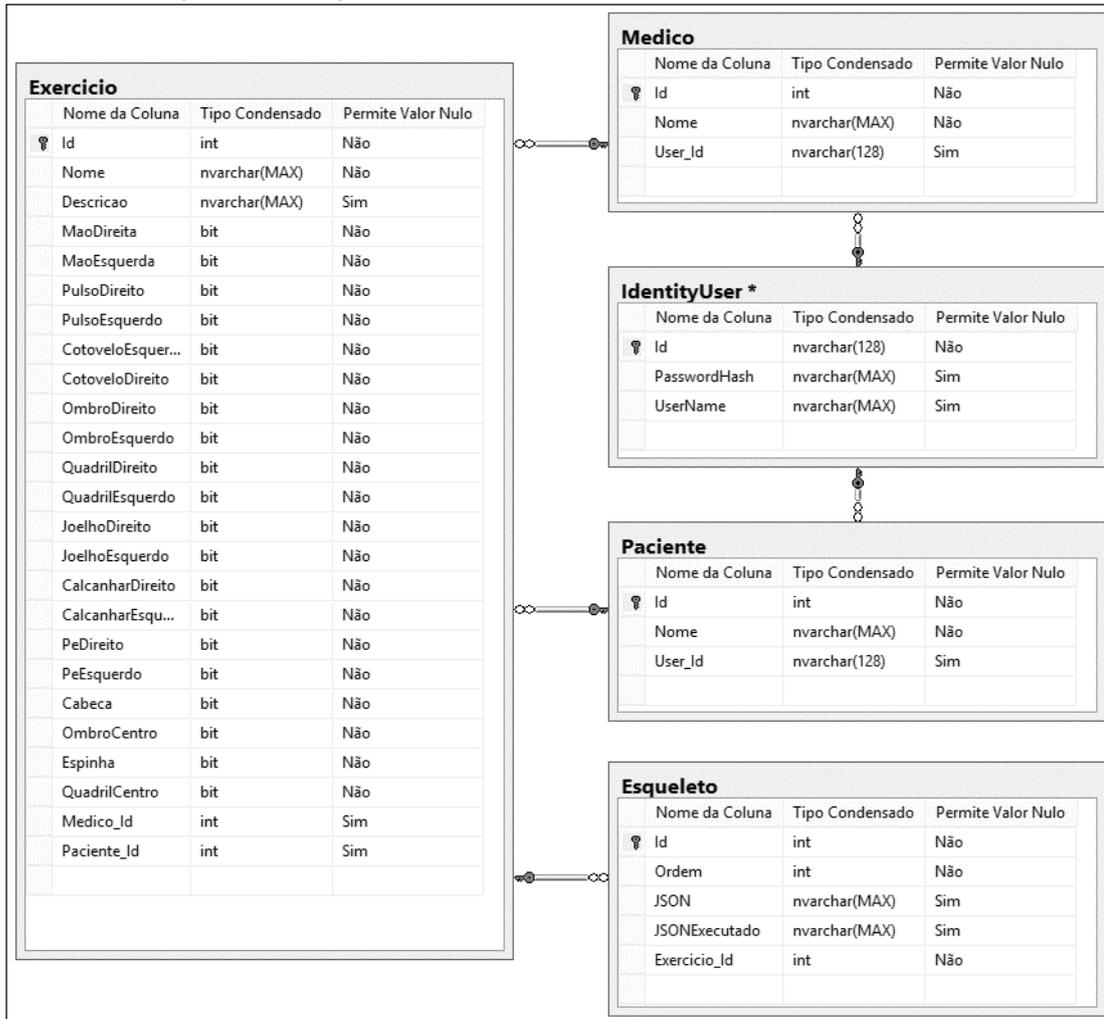
Quadro 34 - UC06 Executar exercícios

Número	06
Caso de Uso	Executar exercícios
Ator	Paciente
Pré-condições	<ol style="list-style-type: none"> <li>1. Ter um exercício com as posições gravadas pelo Fisioterapeuta;</li> <li>2. Estar com o Kinect conectado e posicionado corretamente;</li> <li>3. Estar posicionado corretamente em frente ao Kinect.</li> </ol>
Cenário Principal	<ol style="list-style-type: none"> <li>1. Clicar no botão Executar do exercício desejado na aba Exercícios;</li> <li>2. Reproduzir com maior similaridade possível a posição cadastrada pelo Fisioterapeuta atentando-se às orientações exibidas pela aplicação;</li> <li>3. Falar a palavra <i>SAVE</i> para salvar a posição atual e posicionar na próxima posição se houver;</li> <li>4. Repetir a operação 3 até chegar na última posição;</li> <li>5. Após salvar a última posição a aplicação exibirá a mensagem de conclusão do exercício.</li> </ol>

## APÊNDICE B – Modelo de entidade Relacionamento

Para ilustrar a base de dados da aplicação, utilizou-se do Modelo de Entidade Relacionamento (MER). Na Figura 33 podem ser vistos os atributos, tipos e chaves primárias e estrangeiras de todas as tabelas da aplicação.

Figura 33 - Diagrama de entidade e relacionamento da base de dados



Conforme mostra a Figura 33, tanto a entidade `Medico` quanto a entidade `Paciente` possuem uma ligação para entidade `IdentityUser` a qual contém as informações que são utilizadas e gerenciadas pelo mecanismo de autenticação do ASP.NET MVC.

Já na entidade `Exercicio` observa-se as informações básicas do exercício e seu relacionamento com as entidades `Medico` e `Paciente` através das chaves estrangeiras `Medico_Id` e `Paciente_Id` respectivamente. Os demais campos que controlam quais as juntas devem ser avaliadas são campos do tipo `bit`. A entidade `Esqueleto` por sua vez armazena as posições dos exercícios, sendo que a posição que deve ser executada e a maneira como foi executada são armazenadas nos atributos `JSON` e `JSONExecutado` respectivamente, os quais são do tipo `varchar` e armazenam todas as juntas do corpo no formato *JavaScript Object Notation* (JSON).

## APÊNDICE C – Cadastros iniciais da aplicação

A Figura 34 mostra como cadastrar ou editar as informações do usuário médico, onde na aba **Médicos** clica-se no botão  para editar e para cadastrar um novo clica-se no botão .

Figura 34 - Cadastro de médicos



Nome
Médico 01

A Figura 35 mostra a tela de cadastro de médicos, onde deve-se informar um nome e usuário de acesso.

Figura 35 - Cadastrar ou editar médico

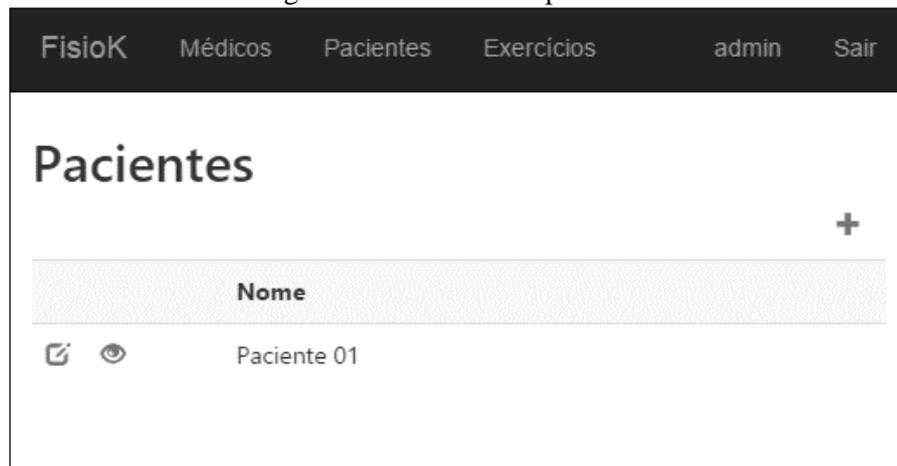


<b>Nome</b>	Médico 02
<b>Usuário</b>	medico.02

Salvar Voltar

Na Figura 36 abaixo é apresentado como deve-se cadastrar ou editar um paciente, onde na aba **Pacientes** clica-se no botão  para editar e para cadastrar um novo clica-se no botão .

Figura 36 - Cadastro de pacientes



A Figura 37 mostra a tela de cadastro de pacientes, onde deve-se informar um nome e usuário de acesso.

Figura 37 - Cadastrar ou editar paciente

**Nome**

**Usuário**

A Figura 38 mostra como cadastrar ou editar um exercício, onde na aba *Exercícios* clica-se no botão  para editar e para cadastrar um novo no botão . Para gravar os movimentos clica-se no botão  e para visualizar o progresso no botão .

Figura 38 - Cadastro de exercícios



	Nome	Médico	Paciente
   	Exercício 01	Médico 01	Paciente 01
   	Exercício 02	Médico 01	Paciente 01

**APÊNDICE D – Roteiro e questionário de perfil de usuário e usabilidade**

Neste apêndice constam os questionários de perfil de usuário, instruções do fisioterapeuta e os questionários de usabilidade do usuário fisioterapeuta e do usuário paciente. O Quadro 35 mostra o questionário aplicado a todos usuários. O Quadro 36 mostra o questionário de instruções passado ao usuário fisioterapeuta, o qual possui uma lista de tarefas que este deveria executar. No Quadro 37 e no

Quadro 38 consta o questionário de usabilidade da aplicação dado ao fisioterapeuta e paciente respectivamente.

Quadro 35 - Questionário de perfil de usuário

<p><b>PERFIL DE USUÁRIO</b></p> <p>Observação: as informações recebidas abaixo serão mantidas de forma confidencial.</p> <p><b>Sexo:</b> ( ) Masculino ( ) Feminino</p> <p>( ) Instrutor    <b>Área:</b> ( ) Fisioterapia    ( ) Outra _____</p> <p>( ) Paciente    ( ) Aluno</p>	
<p><b>Idade:</b></p> <p>( ) Tenho menos de 5 anos</p> <p>( ) Tenho entre 5 e 10 anos</p> <p>( ) Tenho entre 10 e 15 anos</p> <p>( ) Tenho entre 15 e 20 anos</p>	<p>( ) Tenho entre 20 e 25 anos</p> <p>( ) Tenho entre 25 e 30 anos</p> <p>( ) Tenho entre 30 e 35 anos</p> <p>( ) Tenho mais de 35 anos</p>
<p><b>Nível de escolaridade:</b></p> <p>( ) Ensino fundamental incompleto</p> <p>( ) Ensino fundamental completo – 1º grau</p> <p>( ) Ensino médio incompleto</p> <p>( ) Ensino médio completo – 2º grau</p> <p>( ) Ensino superior incompleto</p> <p>( ) Ensino superior completo</p> <p>Observação:</p> <p>_____</p> <p>_____</p>	
<p><b>Você utiliza o computador com qual frequência?</b></p> <p>( ) Nunca utilizei</p> <p>( ) Às vezes</p> <p>( ) Frequentemente</p>	
<p><b>Indique seu grau de familiaridade com o dispositivo Kinect:</b></p> <p>( ) Nunca ouvi falar</p> <p>( ) Conheço, mas nunca utilizei</p> <p>( ) Já utilizei</p> <p>Observação (utilizou para que):</p> <p>_____</p> <p>_____</p>	

Quadro 36 - Questionário de tarefas a serem realizadas pelo fisioterapeuta

<p>1. Realize o cadastro de um paciente, informando o nome e usuário de acesso.</p> <p>A tarefa foi executada? ( ) Sim ( ) Não</p> <p>Observação: _____</p> <p>_____</p>
<p>2. Realize o cadastro de um exercício, informando o nome e selecionando o paciente.</p> <p>A tarefa foi executada? ( ) Sim ( ) Não</p> <p>Observação: _____</p> <p>_____</p>

3. Realize o cadastro das juntas que serão avaliadas durante a realização do exercício pelo paciente.

A tarefa foi executada?  Sim  Não

Observação: \_\_\_\_\_

4. Grave as posições que o paciente deverá repetir durante a realização do exercício.

A tarefa foi executada?  Sim  Não

Observação: \_\_\_\_\_

5. Visualize/confira como o paciente executou cada exercício.

A tarefa foi executada?  Sim  Não

Observação: \_\_\_\_\_

Quadro 37 - Questionário de usabilidade aplicado no usuário fisioterapeuta

1. De modo geral, você achou a aplicação intuitiva e de fácil utilização?

Sim  Não

2. Como você avalia a representação gráfica do paciente dentro da aplicação?

Muito bom

Bom

Regular

Insatisfatório

3. Como você avalia a funcionalidade para informar quais juntas serão avaliadas pela aplicação

Muito bom

Bom

Regular

Insatisfatório

4. Você teve dificuldade para cadastrar o exercício

Sim  Não Qual? \_\_\_\_\_

5. Você considera importante a avaliação do percentual de acerto feita pela aplicação?

Sim  Não

6. Você considera importante visualizar o desempenho do paciente após a realização do exercício

Sim  Não

7. Você precisaria aprender muitas coisas antes de usar o aplicativo.

Sim  Não

8. Você acha que seria necessário o apoio de uma pessoa técnica para poder usar este aplicativo.

Sim  Não

9. Foi fácil lembrar como fazer as coisas no aplicativo

Sim  Não

10. Você utilizaria esta aplicação no seu trabalho?

Sim  Não

11. A aplicação aumentaria a eficácia do seu trabalho

Sim  Não

12. Você acha que os pacientes se sentiriam mais motivados utilizando a aplicação desenvolvida nas sessões de fisioterapia

Sim  Não

13. Você recomendaria este aplicativo para outras pessoas.

Sim  Não

14. Qual é sua avaliação do protótipo?

Muito bom

Bom

Regular

Insatisfatório

15. Qual foi sua maior dificuldade utilizando o protótipo?

---

---

Quadro 38 - Questionário de usabilidade aplicado ao usuário paciente

1. Dos exercícios feitos, quais você conseguiu executar de forma correta?  
 Todos  
 A maioria  
 Metade  
 Menos da metade  
 Nenhum
2. De modo geral, você achou a aplicação intuitiva e fácil utilização?  
 Sim  Não
3. Como você avalia a representação gráfica de cada posição a ser repetida?  
 Muito bom  
 Bom  
 Regular  
 Insatisfatório
4. Como você avalia a forma que a aplicação aponta se o exercício foi feito corretamente?  
 Muito bom  
 Bom  
 Regular  
 Insatisfatório
5. Você considera importante que a aplicação dê orientações de como executar o exercício?  
 Sim  Não
6. Você conseguiu entender as orientações dadas para alcançar o objetivo do exercício?  
 Sim  Não
7. Quanto às mensagens da aplicação?  
 Elas são fáceis de entender.  
 Poderia ser mais fácil – tive dúvida em alguns momentos.  
 São difíceis de entender.
8. Senti-me confortável usando este sistema.  
 Sim  Não
9. Às vezes eu não sei o que fazer com este aplicativo  
 Sim  Não
10. Você precisaria do apoio de uma pessoa para usar este aplicativo  
 Sim  Não
11. O aplicativo em algum momento parou inesperadamente  
 Sim  Não

12. Você recomendaria este aplicativo para outras pessoas.

Sim  Não

13. Qual é sua avaliação do protótipo?

Muito bom

Bom

Regular

Insatisfatório

14. Qual foi sua maior dificuldade utilizando o protótipo?

---

---

15. Você gostou de utilizar o Kinect para realizar os exercícios?

---

---