



CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS

Recredenciado pela Portaria Ministerial nº 1.162, de 13/10/16, D.O.U nº 198, de 14/10/2016
ASSOCIAÇÃO EDUCACIONAL LUTERANA DO BRASIL

Luiz Carlos Fernandes da Luz

FERRAMENTA DE ACOMPANHAMENTO DA EVOLUÇÃO DO PACIENTE EM UM
SISTEMA PARA AVALIAÇÃO DA AMPLITUDE DE MOVIMENTO (ADM)

Palmas - TO

2017

Luiz Carlos Fernandes da Luz
FERRAMENTA DE ACOMPANHAMENTO DA EVOLUÇÃO DO PACIENTE EM UM
SISTEMA PARA AVALIAÇÃO DA AMPLITUDE DE MOVIMENTO (ADM)

Trabalho de Conclusão de Curso (TCC) elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientador: Prof. M.e Fabiano Fagundes.

Palmas - TO

2017

Luiz Carlos Fernandes da Luz

FERRAMENTA DE ACOMPANHAMENTO DA EVOLUÇÃO DO PACIENTE EM UM
SISTEMA PARA AVALIAÇÃO DA AMPLITUDE DE MOVIMENTO (ADM)

Trabalho de Conclusão de Curso (TCC)
elaborado e apresentado como requisito parcial
para obtenção do título de bacharel em
Sistemas de Informação pelo Centro
Universitário Luterano de Palmas
(CEULP/ULBRA).

Orientador: Prof. M.e. Fabiano Fagundes.

Aprovada em: de 2017.

BANCA EXAMINADORA

Prof. M.e Fabiano Fagundes
Centro Universitário Luterano de Palmas

Prof. M.e Fernando Luiz de Oliveira
Centro Universitário Luterano de Palmas

Prof. M.e Madianita Bogo Marioti
Centro Universitário Luterano de Palmas

Palmas - TO

2017

AGRADECIMENTOS

Agradeço primeiramente a Deus, senhor de todas as coisas, que me concedeu a oportunidade de vida e que me proporciona uma vida cheia de aventuras e em nenhum momento me faz sentir abandonado da sua proteção de Pai. Aos meus pais que batalharam para a minha criação, formação e hoje contemplam comigo mais essa vitória, as minhas irmãs que sempre aceitaram minhas ausências nos eventos familiares. A Keydi, minha esposa e eterna parceira, que nunca me desanimou e sim só contribuiu para o meu crescimento pessoal e profissional, tens meu amor e minha eterna gratidão.

Ao mestre Fabiano, meu orientador, “pai” e pessoa de exímia sagacidade, pois soube explorar de mim capacidades intelectuais que até eu mesmo duvidava existir, muitíssimo obrigado por me mostrar e trilhar este caminho comigo. Aos professores das bancas de Estágio e TCC, Fernando, Madianita e Pierre que sempre acrescentaram informações relevantes para o meu aprimoramento acadêmico. Aos professores Cristina, Edeilson, Jackson e Parcilene pelas instruções e combinações do curso com as peripécias da vida e do mercado.

Também aos grandes amigos encontrados ao decorrer do curso, em especial aos “Koxinhas” grupo de muita divulgação de informações e de pôr as amizades em prova (risos), todas mantidas até então. E ainda aos amigos da vida que de forma indireta colaboraram para a conclusão do curso.

RESUMO

LUZ, Luiz Carlos Fernandes da. **Ferramenta de Acompanhamento da evolução do Paciente em um Sistema Para Avaliação da Amplitude de Movimento (ADM)**. 2017. 54 f. Trabalho de Conclusão de Curso (Graduação) – Curso de Sistemas de Informação, Centro Universitário Luterano de Palmas, Palmas-TO, 2017.

O presente trabalho apresentou uma melhoria no módulo de Amplitude de Movimento (ADM) da ferramenta FísioKinect desenvolvida pelo Grupo de Estudos e Pesquisa em Tecnologia, Saúde e Qualidade de Vida (GEPETS) do Centro Universitário Luterano de Palmas (CEULP). O sistema FísioKinect possui módulos que realizam avaliações do corpo como antropometria, hiperlordose etc. Neste trabalho é enfatizado o módulo ADM que realiza a avaliação de ADM com o sensor Kinect v1. Posteriormente foi desenvolvida uma adaptação do módulo para que utilizasse o Kinect v2, foram utilizados elementos da plataforma *Vitruvius* para o desenvolvimento do sistema, efetuando os cálculos dos ângulos nos movimentos exercidos das articulações do ombro e quadril, resultados necessários para o profissional obter um diagnóstico. Atualmente as avaliações do módulo ADM apresentam resultados em relatórios com os valores dos ângulos atingidos pelo paciente. Como melhoria, o sistema foi adaptado para seguir os padrões dos sistemas automatizados de avaliação que, além dos relatórios numéricos, exibem os ângulos obtidos através de imagens.

Palavras-chave: Amplitude de Movimento, FísioKinect, Melhoria e Kinect v2.

LISTA DE ABREVIATURAS

ADM	Amplitude de Movimento
CEULP	Centro Universitário Luterano de Palmas
GEPETS	Grupo de Estudos e Pesquisa em Tecnologia, Saúde e Qualidade de Vida
SAA	Sistemas Automatizados de Avaliação
SDK	Software Development Kit

LISTA DE FIGURAS

Figura 1 - Goniômetro Universal para análise da ADM	11
Figura 2 - Planos anatômicos.....	13
Figura 3 - Abdução e adução do ombro	14
Figura 4 - Padrão para abdução do quadril.....	14
Figura 5 - Tela de exame da ferramenta MediFisio.....	16
Figura 6 - Relatório por imagem - Ferramenta Physical Test	17
Figura 7 - Interface Inicial FisioKinect	18
Figura 8 - Relatório módulo ADM – FisioKinect	19
Figura 9 - Desenho do Estudo	20
Figura 10 - Caminho do Diretório da aplicação para gravação da imagem	21
Figura 11 - Troca de Background – Vitruvius.....	24
Figura 12 - Arquitetura Módulo ADM Atualizado	27
Figura 14 - Salvamento da imagem capturada	29
Figura 15 - Troca de background – FisioKinect ADM	30
Figura 16 - Interação do paciente com o background	31
Figura 17 - Relatório de resultados	32
Figura 18 - Comparativo de avaliações ADM.....	34

SUMÁRIO

1	INTRODUÇÃO	9
2	REFERENCIAL TEÓRICO	11
2.1.	Amplitude de Movimento	11
2.2.	Trabalhos Correlatos	15
2.2.1.	MediFisio.....	15
2.2.2.	<i>Physical Test</i>	16
2.3.	FisioKinect	18
3	MATERIAIS E MÉTODOS.....	20
3.1.	Desenho do Estudo	20
3.2.	Local e período de realização do estudo	21
3.3.	Hardware.....	22
3.3.1.	Kinect v2	22
3.4.	Softwares.....	22
3.4.1.	Vitruvius.....	23
3.4.2.	Desenvolvimento de Melhoria do Software	24
4	RESULTADOS E DISCUSSÕES	26
4.1.	Arquitetura	26
4.2.	Tela da aplicação que realiza o acompanhamento da evolução do paciente	27
4.3.	Codificação da captura de tela	28
4.4.	Plano de Fundo	29
4.5.	Configuração da interação com o <i>background</i>	31
4.6.	Relatório dos resultados da avaliação de ADM	31
4.7.	Protocolo de Utilização.....	33
5	CONSIDERAÇÕES FINAIS.....	34
	REFERÊNCIAS.....	36
	APÊNDICES	38
	APÊNDICE A – Código fonte da tela principal – FisioKinect.....	39
	APÊNDICE B – Código fonte da tela inicial Módulo ADM.....	43
	APÊNDICE C – Código fonte da classe de Background.....	52

1 INTRODUÇÃO

Conforme Pereira Junior e Petró (2010), o sensor Kinect, acessório do console Xbox 360, foi um grande marco na história dos games, pois possibilitou aos seus usuários jogarem sem a interferência de um *joystick*, diferente do que ocorre na maioria dos consoles, o que deu maior liberdade de movimentos ao jogador. O sensor do Kinect permitiu, também, que surgissem experimentos fora do ramo dos games como controle de robôs, de aeronave, dentre outros. Para auxiliar essas inovações, a *Microsoft* desenvolveu em 2012 o *Microsoft Kinect for Windows*, pacote de codificação própria para o Kinect, que conta com exemplos de códigos para facilitar a criação de novas aplicações.

Um dos campos desses novos projetos que foram beneficiados com a tecnologia do Kinect é o da saúde, onde já existem sistemas com alimentação de imagens como, por exemplo sistemas de imagens radiológicas e sistemas que auxiliam fisioterapeutas a executarem o tratamento com o seu paciente com ajuda do dispositivo. Esses tipos de ferramentas têm como intuito a reabilitação dos pacientes na fisioterapia com a captura de movimentos pela câmera do sensor.

Um exemplo de um desses sistemas é o FisioKinect, aplicação desenvolvida pelo Grupo de Estudos e Pesquisa em Tecnologia, Saúde e Qualidade de Vida (GEPETS) do Centro Universitário Luterano de Palmas (CEULP), que possui vários módulos que realizam diferentes formas de avaliação do corpo humano com auxílio da câmera do Kinect.

Um de seus módulos é o Amplitude de Movimento (ADM), que realiza o cálculo dos ângulos de ADM das articulações de ombro e quadril do paciente frente a câmera do Kinect v1 desenvolvido pelo egresso Thayso Henrique Capuchinho Camargo do Curso de Ciências da Computação como conclusão de curso em 2015. Segundo Luz (2016), o módulo ADM foi atualizado em 2016 para que examinasse o paciente utilizando o sensor Kinect v2 em conjunto com elementos necessários da plataforma *Vitruvius*, que realiza a simplificação da codificação necessária para realização da análise de ADM.

Mesmo com a utilização da tecnologia do Kinect v2, o módulo ADM não modificou os relatórios na maneira de exibição dos resultados dos ângulos atingidos na ADM, expressados em valores numéricos, não gerando motivações para o paciente acompanhar os resultados do tratamento contribuindo na desistência do paciente durante o tratamento. Assim, este trabalho buscou responder ao seguinte problema de pesquisa: como armazenar e exibir ao paciente, que tem sua ADM das articulações do ombro e quadril avaliada pelo *Sensor Microsoft Kinect v2*, além dos valores numéricos, imagens da sua última avaliação para servir de acompanhamento de evolução do tratamento feito pelo profissional?

A hipótese levantada é de que, como o sensor Kinect v2 realiza a captura de tela desejada, é possível que a imagem capturada do paciente durante a avaliação de ADM torne-se o *background* das avaliações de ADM posteriores. Portanto, o objetivo geral deste trabalho foi desenvolver uma adaptação no módulo ADM da ferramenta FizioKinect permitindo um acompanhamento mais nítido dos resultados das avaliações de ADM através de imagens que apresentam a evolução do trabalho realizado. Neste sentido objetivou-se especificamente em:

- realizar a captura de imagens do paciente durante a avaliação de ADM das articulações de ombro e quadril, através do dispositivo Kinect v2;
- armazenar os resultados dos ângulos das articulações de ombro e quadril no exercício de ADM tanto em valores numéricos como em imagens, que se tornarão o limite mínimo a ser atingido pelo paciente pois já os alcançou em exames anteriores;
- exibir as imagens capturadas do paciente, de acordo com a articulação do ombro ou do quadril, selecionada pelo profissional para acompanhamento dos exercícios aplicados pelo profissional e efetuados pelo paciente nos relatórios ou no background da avaliação de ADM, como forma de motivação ou meta para as próximas avaliações.

Conforme Luz (2016), a melhoria no módulo ADM da ferramenta FizioKinect, passou a apresentar as evoluções do paciente por meio de suas imagens capturadas pelo sensor Kinect v2, tornou-se possível realizar um acompanhamento mais nítido dos resultados das avaliações de ADM fornecendo uma supervisão dos resultados obtidos e a percepção do que deve ser conseguido para superar a avaliação anterior.

O trabalho está estruturado da seguinte forma: o capítulo 2 apresenta o referencial teórico dividido entre Amplitude de Movimento, Trabalhos Correlatos que apresenta softwares que exibem os resultados de avaliações por meio de imagens e a ferramenta FizioKinect. No capítulo 3 são apresentados a metodologia e os materiais utilizados no desenvolvimento deste trabalho. O capítulo 4 apresenta os resultados obtidos na melhoria aplicada ao módulo ADM da FizioKinect. No capítulo 5 são relacionadas as conclusões deste trabalho, e por fim as referências.

2 REFERENCIAL TEÓRICO

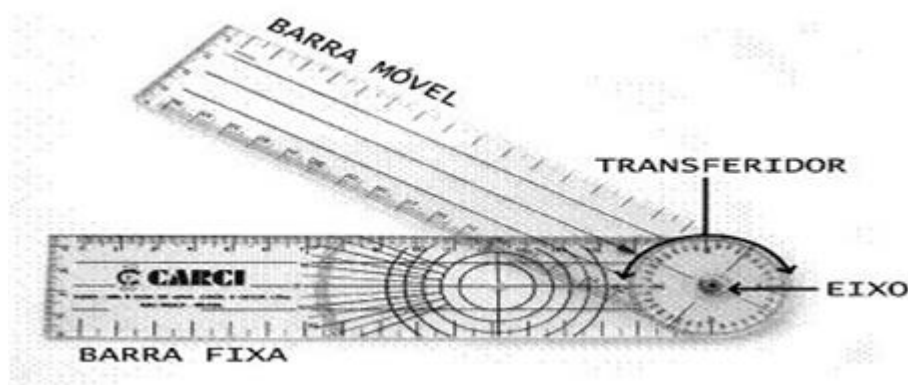
Nesta seção são abordados os conceitos necessários para o desenvolvimento e entendimento deste trabalho. A princípio são apresentados os fundamentos da Amplitude de Movimento (ADM), como ela foi utilizada para o cálculo dos ângulos das articulações e o quanto pode ser benéfica para o tratamento de deficiências motoras. São contextualizados também os Sistemas Automatizados de Avaliação (SAA), na forma em que esses sistemas representam os dados obtidos nessas automações de avaliação e de como é proveitoso o acompanhamento por imagem dos resultados por parte do paciente.

2.1. Amplitude de Movimento

“A ADM é a quantidade de movimento de uma articulação. A posição inicial para se medir a ADM de todas as articulações, com exceção dos movimentos de rotação, é a posição anatômica” (MARQUES, 2014, p. 20). Qualquer movimento exercido por um segmento do corpo humano (braço direito, braço esquerdo, perna direita e perna esquerda) permite a avaliação de ADM executada por um profissional, que analisa a medida do ângulo do membro distanciado do eixo corporal baseando-se na posição anatômica.

O cálculo das medidas de ADM das articulações do corpo humano é orientado pela goniometria, técnica para se obter as medidas dos ângulos, sendo o goniômetro universal o instrumento mais utilizado para se calcular a ADM. Os goniômetros “têm um corpo com dois braços: um móvel e outro fixo. No corpo do goniômetro, estão as escalas, podendo ser um círculo completo (0 a 360°) ou meio círculo (0 a 180°)” (MARQUES, 2014, p. 14). Na figura 1 é exibido o goniômetro, aparelho mais utilizado entre os profissionais para avaliar a ADM.

Figura 1 - Goniômetro Universal para análise da ADM



Fonte: Adaptação de Rosely Saldanha (2011, online).

O padrão básico do instrumento é um corpo com dois braços: um móvel e o outro fixo. Através dessa ferramenta o profissional consegue mensurar os graus dos ângulos corporais nos movimentos exercidos pelo paciente. O eixo do instrumento é posicionado sobre o ponto anatômico do paciente a ser observado pelo profissional.

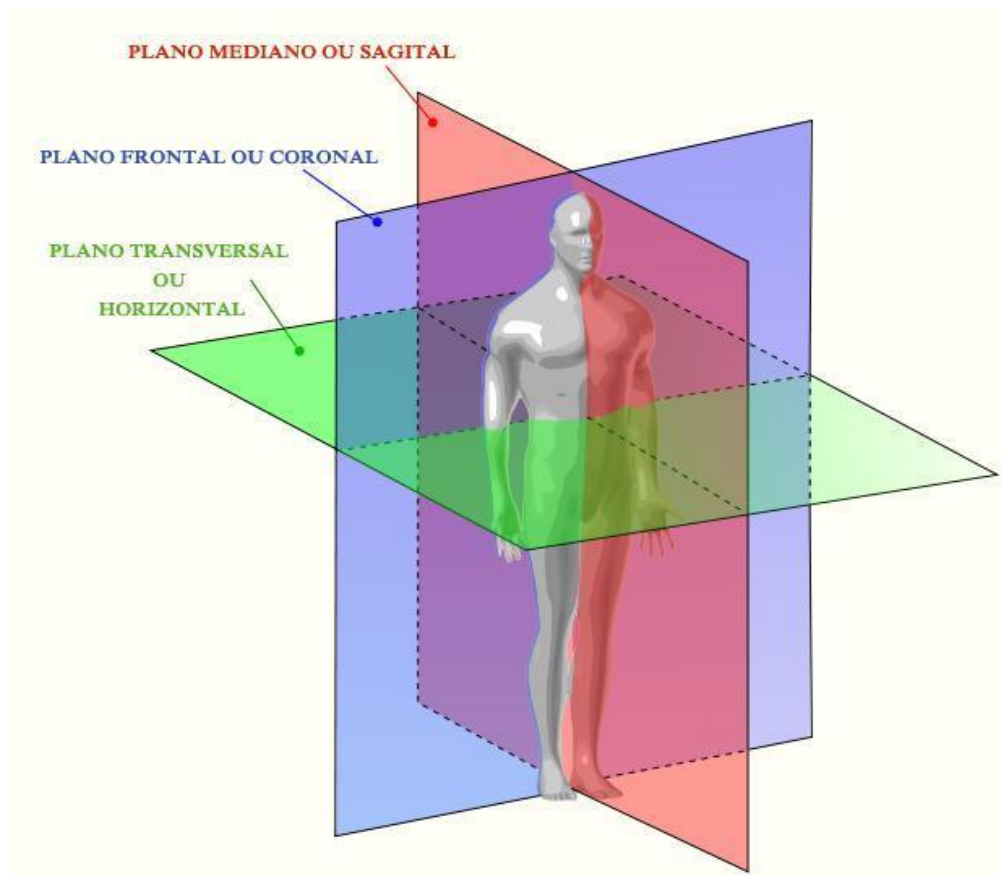
Conforme Marques (2003), os movimentos articulares ocorrem em três planos: frontal, sagital e transverso. Para avaliações do corpo é necessário entender sobre os planos anatômicos que delimitam a anatomia corporal:

- Plano Sagital: divide o corpo humano em duas porções, lado direito e lado esquerdo, possuindo dois tipos de movimentos para as articulações que são flexão e extensão;
- Plano Coronal ou Frontal: a sua divisão imaginária ocorre como se o corpo estivesse de lado e o dividisse em duas partes, anterior que é a parte da frente e posterior que a parte das costas, os movimentos executados nesse plano são abdução e adução;
- Plano Transversal: o corpo humano é dividido nas porções superior e inferior ou cranial e caudal, o movimento das articulações admitido nesse plano é a rotação.

Para este trabalho a ADM teve enfoque no plano coronal ou frontal para as articulações do ombro e do quadril. Conforme Rosa Filho (2002), os movimentos analisados pelo profissional são denominados pela fisioterapia como abdução, afastamento da articulação em relação ao corpo, e adução, retorno da articulação em relação ao corpo, que são avaliados na ADM. As literaturas do ramo fisioterapêutico definem para as articulações os padrões ideais para o movimento ser atingido e uma posição de referência em que o paciente deve-se posicionar.

Para simetria dos planos anatômicos existem os eixos cartesianos que ligam as divisões das seções imaginárias, que representam os planos denominados como: eixo sagital, que junta o plano coronal na visão anterior e posterior; o eixo transversal, que liga a porção superior e inferior do plano; e o eixo transversal, que conecta as divisões direita e esquerda do plano sagital. A figura 2 apresenta as seções imaginárias que delimitam os planos e eixos anatômicos.

Figura 2 - Planos anatômicos



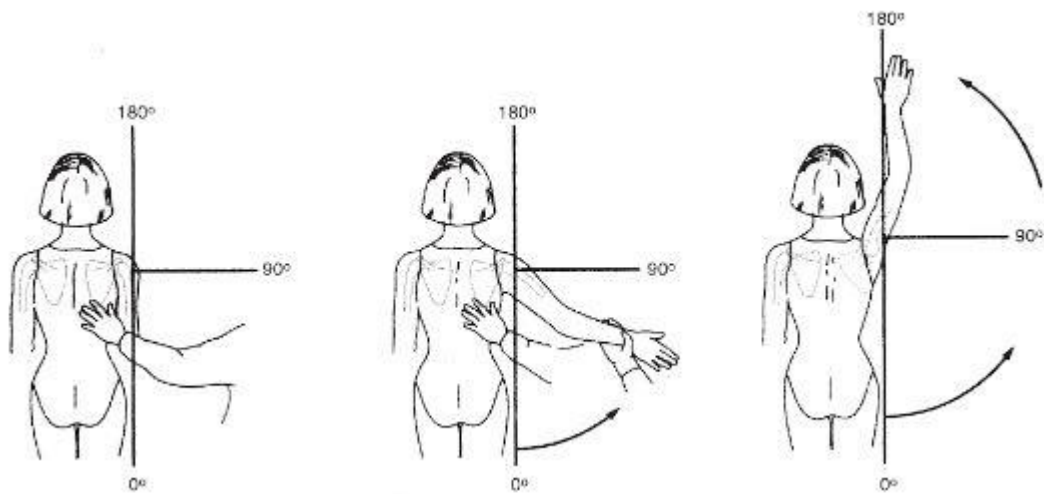
Fonte: Adaptação de Nick Zuccarello (2014, online).

A goniometria fornece a verificação de disfunção ou não da articulação e permite estabelecer diagnósticos, metas para tratamento, acompanhar a melhoria nos exercícios realizados e direcionamento para fabricação de órteses¹.

Para reconhecer a posição anatômica da articulação é necessário que o paciente esteja na orientação de referência adotada pelos profissionais da saúde, o indivíduo deve se posicionar de forma ereta (em pé), com a face para a frente, braços e antebraços próximos ao tronco com as palmas das mãos viradas para a frente e pernas e pés juntos e para frente.

A figura 3 demonstra a posição de referência dos movimentos para a articulação do ombro.

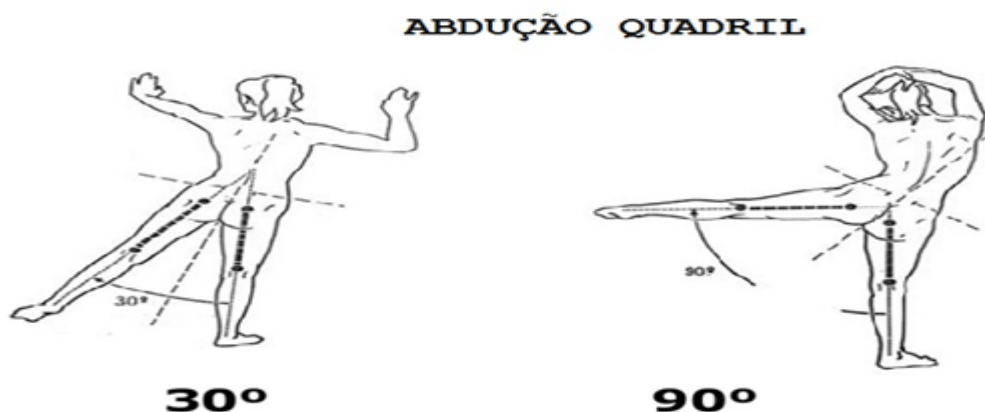
¹ Órtese é todo dispositivo permanente ou transitório, utilizado para auxiliar as funções de um membro, órgão ou tecido, evitando deformidades ou sua progressão e/ou compensando insuficiências funcionais. (BRITO, 2015).

Figura 3 - Abdução e adução do ombro

Fonte: Dutton, 2009, p.515

Conforme Dutton (2009), os movimentos possuem três valores de base a serem atingidos, que são o ângulo 0° , quando não há nenhum tipo de deslocamento do ombro; o ângulo de 90° , quando o ombro se encontra na posição de ângulo reto; e 180° , que é o ápice do movimento ao ser exercido. A figura 3 apresenta os ângulos de 0° que representam a adução e os ângulos de 90° e 180° que são movimentos de abduções.

Segundo Duarte (2011), a posição de referência para a articulação do quadril atinge a satisfação máxima quando o paciente consegue realizar abdução de 90° em relação ao corpo, na figura 4 são exibidas as posições anatômicas ideais para análise da ADM da articulação do quadril.

Figura 4 - Padrão para abdução do quadril

Fonte: Adaptação de Duarte (2011, online).

O paciente deverá se sobrepor na perna direita para a abdução do quadril esquerdo e se sobrepor na perna esquerda para abduzir o quadril direito, possibilitando o movimento das articulações a serem mensurados, o ângulo de 30° representa o ângulo médio e o ângulo de 90° refere-se ao valor máximo a ser obtido, ambos representam o movimento de abdução.

2.2. Trabalhos Correlatos

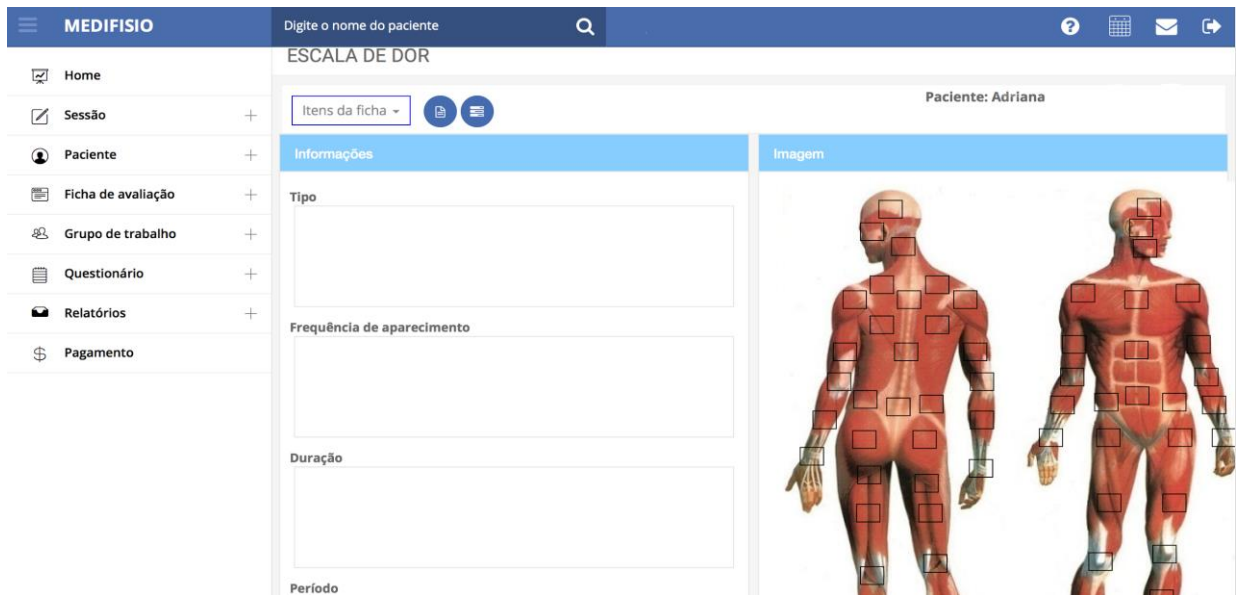
2.2.1. MediFisio

Segundo Pereira (2013), a automação de sistemas é a otimização de processos que permite que operações manuais possam ser executadas pelo sistema sem interferência humana. A automatização de procedimentos está abrangendo todas as áreas do conhecimento humano, inclusive o setor da saúde, e é gradualmente mais comum encontrar sistemas automatizando rotinas com o intuito de garantir uma padronização de resultados.

Cada vez mais os processos de avaliações de patologias ou disfunções motoras estão sendo incrementados nos sistemas automatizados para que o profissional da saúde se dedique apenas em diagnosticar o estado de saúde do seu paciente. As avaliações automatizadas possuem o objetivo de auxiliar o profissional da saúde na execução de seus exames, agilizar o colhimento de análises, e ainda contribuir para uma melhor precisão nos resultados.

A fisioterapia já possui várias ferramentas que integralizam informações fisioterapêuticas e fornecem uma gama de funcionalidades para que a consulta do paciente seja a mais prática possível. É o caso do sistema MediFisio da empresa TIMAX Informática Ltda, especialista em softwares no ramo fisioterapêutico como reabilitação motora, hidroterapia (hidroterapeuta), prevenção e reabilitação de lesões e disfunções musculares ou músculo-esqueléticas (INFORMÁTICA, 2010). A figura 5 apresenta um exemplo da ferramenta Medifisio.

Figura 5 - Tela de exame da ferramenta MediFisio



Fonte: Informática (2010)

Na figura 5, o profissional tem as opções de acesso através de um menu à esquerda que permite o cadastro do paciente, a ficha de avaliação, questionário, relatórios e, no caso deste sistema que é voltado para clínicas particulares, a forma de pagamento.

Conforme a tela apresentada da ferramenta Medifisio, o profissional faz o preenchimento da ficha avaliativa para determinar uma escala de dor mediante as queixas do paciente, além da ficha, o profissional marca as articulações que estão causando desconforto para o paciente na imagem à direita, proporcionando um diagnóstico mais ágil do profissional.

O sistema Medifisio apresenta aspectos positivos para avaliações de sessões fisioterapêuticas como formulários com imagens auxiliares em que o profissional pode identificar as anormalidades do paciente mais rapidamente. Por outro lado, as informações dos formulários são inseridas manualmente o que torna as avaliações demoradas.

2.2.2. *Physical Test*





A ferramenta *Physical Test* é uma outra ferramenta de avaliação automatizada na área da fisioterapia, desenvolvida pela empresa Terrazul Tecnologia, especializada em sistemas para avaliação e preparo físico, assim como no acompanhamento da evolução fisiológica e nutricional (TERRAZUL, 1992).

A figura 6 exemplifica um tipo de relatório por imagem, obtido através da ferramenta *Physical Test*.

Figura 6 - Relatório por imagem - Ferramenta *Physical Test*

TERRAZUL INFORMÁTICA LTDA.
 Código de avaliação: 0000001 Nome: ALUNO EXEMPLO Data de avaliação: 15/08/2005 No.: 01

AValiação POSTURAL DIGITAL

Visão Anterior <input checked="" type="checkbox"/> Triângulo de Tales Assimétrico <input type="checkbox"/> Pé abduído <input type="checkbox"/> Pé aduto <input checked="" type="checkbox"/> Genu Valgo <input type="checkbox"/> Genu Varo Assimetria de Quadril: Ombros Assimétricos: Elevação Direita: 10, 11, 12, 13		Visão Posterior <input type="checkbox"/> Escoliose Cervical <input type="checkbox"/> Escoliose Torácica <input type="checkbox"/> Escoliose Lombar <input type="checkbox"/> Pé Varo <input type="checkbox"/> Pé Valgo <input type="checkbox"/> Encurtamento de Trapézio <input type="checkbox"/> Protração Escapular <input type="checkbox"/> Retração Escapular <input checked="" type="checkbox"/> Depressão Escapular Tipos de Escoliose: 10, 11, 12, 13	
Visão Lateral Direita <input type="checkbox"/> Retificação Cervical <input type="checkbox"/> Hiperlordose Cervical <input type="checkbox"/> Hipercifose Torácica <input checked="" type="checkbox"/> Hiperlordose Lombar <input type="checkbox"/> Genu Flexo <input checked="" type="checkbox"/> Genu Recurvado <input type="checkbox"/> Antversão de Quadril <input type="checkbox"/> Retroversão de Quadril <input type="checkbox"/> Pé Plano <input type="checkbox"/> Pé Cavo <input type="checkbox"/> Pé Calcâneo <input type="checkbox"/> Pé Equino <input type="checkbox"/> Protusão Abdominal <input type="checkbox"/> Rotação Interna de Ombros (R.I.O.)		Visão Lateral Esquerda <input type="checkbox"/> Retificação Cervical <input type="checkbox"/> Hiperlordose Cervical <input type="checkbox"/> Hipercifose Torácica <input checked="" type="checkbox"/> Hiperlordose Lombar <input type="checkbox"/> Genu Flexo <input checked="" type="checkbox"/> Genu Recurvado <input type="checkbox"/> Antversão de Quadril <input type="checkbox"/> Retroversão de Quadril <input type="checkbox"/> Pé Plano <input type="checkbox"/> Pé Cavo <input type="checkbox"/> Pé Calcâneo <input type="checkbox"/> Pé Equino <input type="checkbox"/> Protusão Abdominal <input type="checkbox"/> Rotação Interna de Ombros (R.I.O.)	
Observações gerais acerca da avaliação postural			

Terrazul Informática Ltda. • TeleSuporte: (0xx11) 3272-9305 Contato: 11 3272-9305

Fonte: Terrazul (1992)

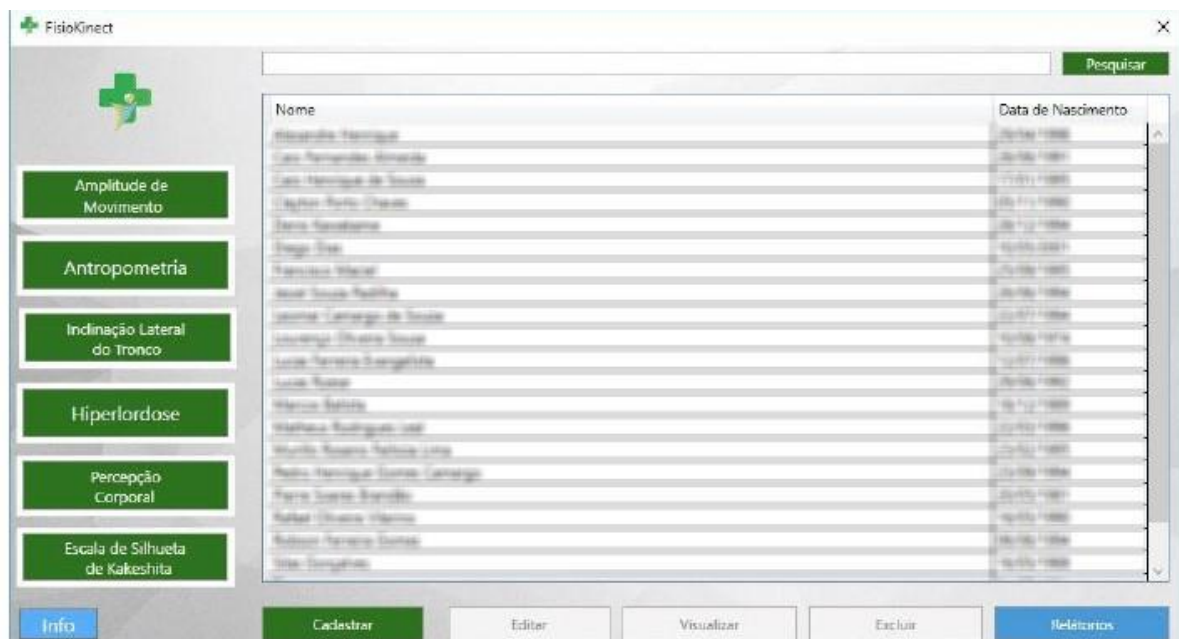
Na figura 6, demonstra a ferramenta fisioterapêutica *Physical Test* que permite a visualização dos resultados dos exames por imagem, neste exemplo o paciente é orientado a se posicionar para o profissional avaliar sua postura. Conforme Terrazul (1992), o sistema *Physical Test* fornece avaliações e relatórios dos diagnósticos com imagens que podem ser impressas ou enviadas por email para comparações que é muito utilizado pelos profissionais para apresentar ao paciente a melhora durante o tratamento.

O acompanhamento por imagem permite um entendimento dos resultados mais abrangível aos que o visualizarem, pois, o profissional pode se deparar com um paciente que não consegue ler os valores numéricos dos ângulos e pode gerar um desestímulo do tratamento. Com isso justifica a grande importância do acompanhamento através de mídias para a fisioterapia e principalmente para o indivíduo que está sendo avaliado.

2.3. FisioKinect

A ferramenta FisioKinect foi desenvolvida pelo GEPETS do CEULP, realiza análise das articulações do corpo humano através da câmera do dispositivo Kinect e diagnostica algum tipo de anomalia como hiperlordose ou disfunção motora pela execução de ADM. A figura 7 exibe a interface da ferramenta FisioKinect e as opções para selecionar um de seus módulos de acordo com a avaliação.

Figura 7 - Interface Inicial FisioKinect



Fonte: Souza, (2015)

A figura 7 apresenta a interface inicial do sistema FisioKinect, com os botões de acesso aos módulos da ferramenta, campo de pesquisa do paciente, botão de informações do sistema como versão e proprietários, botões que compõem o CRUD de uma aplicação (cadastrar, editar, visualizar e excluir) e o botão de relatório, que apresenta os dados levantados das avaliações que existem na ferramenta na forma de relatório. Este projeto enfatiza o módulo de ADM que realiza o exame de ADM com auxílio da câmera do Kinect v2 e recursos da plataforma *Vitruvius* que permitem a realização dos cálculos para obtenção dos ângulos de ADM das articulações de ombro e quadril e arcos de identificação sobre essas articulações.

Quando o profissional acessa o módulo ADM a câmera do dispositivo Kinect v2 é ativada, o paciente se posiciona a frente da câmera, o módulo ADM calcula os ângulos

relativos ao movimento das articulações dos ombros e quadris e projeta arcos de identificação, tornando os ângulos mais visíveis ao profissional.

Ao salvar a avaliação de ADM o sistema coleta os valores dos ângulos obtidos e armazena no banco de dados da ferramenta, permitindo a geração do relatório em que apresenta o nome do paciente e os segmentos avaliados e seus resultados de ADM no exame. A figura 8 exibe o modelo de relatório fornecido pelo módulo ADM.

Figura 8- Relatório módulo ADM – FisioKinect



Nome	Data Avaliação	Braço Direito	Braço Esquerdo	Perna Direita	Perna Esquerda
Luiz Carlos Fernandes da Luz	01/12/2016	69	61	122	-68
Luiz Carlos Fernandes da Luz	01/12/2016	82	38	139	-35
Luiz Carlos Fernandes da Luz	01/12/2016	68	52	-90	-64
Luiz Carlos Fernandes da Luz	01/12/2016	55	53	113	-5
Luiz Carlos Fernandes da Luz	05/12/2016	23	41	-5	4
Luiz Carlos Fernandes da Luz	08/12/2016	97	30	35	32

CEULP/ULBRA
CENTRO UNIVERSITÁRIO LUISIANO DE PALMAS
TECNOLOGIA, SAÚDE E QUALIDADE DE VIDA

Cada linha apresentada no relatório da figura 8 refere-se a uma avaliação na qual um determinado indivíduo executou os movimentos das articulações dos ombros ou quadris, conforme instruções do profissional. O módulo ADM realiza os cálculos para serem registrados na base de dados da ferramenta.

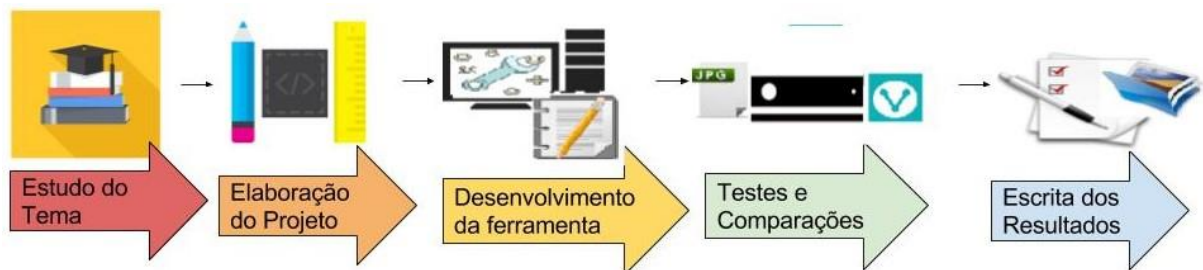
De forma geral os sistemas automatizados de avaliação para área da fisioterapia realizam análise do paciente com opções modernas e de rápida absorção pelo profissional, como identificação das articulações por imagem ou acompanhamento por vídeo, e não só durante o exame como também no acompanhamento por imagens.

3 MATERIAIS E MÉTODOS

3.1. Desenho do Estudo

Para concepção do trabalho proposto inicialmente, foi realizada uma pesquisa baseada no tema estudado, para a elaboração do projeto e suas possibilidades de resultados. A ferramenta foi desenvolvida por meio de parametrização de captura de imagem, função já existente no Kinect v2. As imagens capturadas são armazenadas no banco de dados da FisiKinect para apresentação em relatório ou como *background* da avaliação de ADM. Para troca do plano de fundo do exame avaliação de ADM, ou seja, tornar a imagem capturada como *background* são utilizados recursos do *Vitruvius*, para quando o profissional selecionar a articulação já avaliada em uma nova avaliação, a imagem capturada será exibida como *background*. Após fundamentação teórica para o desenvolvimento da aplicação, foi elaborada um desenho de estudo que representa a metodologia utilizada neste projeto. A figura 9 apresenta o desenho de estudo.

Figura 9 - Desenho do Estudo



O estudo do tema é a base do projeto onde direciona a etapa de elaboração do trabalho para se conseguir desenvolver a ferramenta proposta e com isso realizar os testes e comparações das avaliações ADM do sistema com as avaliações tradicionais feitas de forma manual com o goniômetro. Para aplicação dos testes foi necessário avaliações de um profissional com pacientes realizando exames de ADM para definirem se existem estímulos na avaliação por ter o acompanhamento das evoluções do paciente no tratamento. Os testes foram feitos de forma funcionais em cima das imagens capturadas pelo módulo ADM armazenadas no banco de dados da ferramenta FisiKinect, o profissional e o paciente fazem as comparações dos movimentos executados na nova avaliação de acordo com a articulação selecionada no início de um novo exame. Os testes são analisados por meio dos resultados dos

ângulos de ADM da ferramenta comparando com uma avaliação manual com o goniômetro e um *feedback* do paciente em relação ao acompanhamento por imagem.

3.2. Local e período de realização do estudo

O trabalho foi desenvolvido no Laboratório de Tecnologia em Saúde localizado no Complexo de Informática do CEULP. As informações dos resultados como os valores dos ângulos e o caminho das imagens armazenadas são registradas por meio de conexão ao banco de dados “pegasus” do servidor *sandbox.ulbra-to.br* da instituição.

Os ângulos são gravados diretamente na tabela *Avaliacao_ADM_Sistema* nas colunas respectivas (*TbAngulo1*, *TbAngulo2*, *TbAngulo3*, *TbAngulo4*), nas colunas (*ImgBracoD*, *ImgBracoE*, *ImgPernaD*, *ImgPernaE*) são registrados o caminho da imagem que estão salvas no diretório *ADMImagens* da aplicação. Na figura 10 é apresentado de que forma esse caminho da imagem é armazenado na tabela da avaliação de ADM.

Figura 10 - Caminho do Diretório da aplicação para gravação da imagem

The screenshot shows a SQL query result in a database client. The table has columns for patient ID (Id), age (Tb...), weight (TbAngu...), height (T...), and image paths for the right arm (ImgBracoD), left arm (ImgBracoE), right leg (ImgPernaD), and left leg (ImgPernaE). The paths are stored in the format C:\Users\Luiz Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BracoDireito_8034.jpg.

Id	Tb...	TbAngu...	T...	ImgBracoD	ImgBracoE	ImgPernaD	ImgPernaE
47	16	16		C:\Users\Luiz Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BracoDireito_8034.jpg	C:\Users\Luiz L...	C:\Users\Luiz L...	C:\Users\Luiz L...
33	3	12		C:\Users\Luiz Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BracoDireito_4029.jpg	C:\Users\Luiz L...	C:\Users\Luiz L...	C:\Users\Luiz L...
84	34	50		C:\Users\Luiz Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BracoDireito_8034.jpg	C:\Users\Luiz L...	C:\Users\Luiz L...	C:\Users\Luiz L...
	NULL	NULL	N...	NULL	NULL	NULL	NULL

Conforme exibido na figura 10 as relativas colunas referente as articulações do ombro e quadril e indexadas pelo *Id* do paciente que são avaliadas pela aplicação ao gravar a imagem através da captura da câmera do Kinect o caminho do diretório é inserido no banco de dados mantendo assim o desempenho da ferramenta ao retornar a imagem para a tela de avaliação.

3.3. Hardware

Para o desenvolvimento da ferramenta foi utilizado um computador com o processador Intel core i5 2.30 GHz de 64 bits, barramento USB 3.0 e 2.0, 12GB de RAM e placa de vídeo integrada compatível com o DirectX 11, e um sensor Kinect Xbox ONE com adaptador – responsável pela conexão entre o Kinect v2 e o computador - disponibilizado pelo GEPETS do CEULP. Para implementação das interfaces e códigos foi utilizado o sensor Microsoft Kinect, o software Visual Studio 2015, sendo utilizado para persistir os dados o SGBD SQL Server 2016. Por fim, para fazer a comunicação entre o software e a base de dados foi utilizada a tecnologia LINQ to SQL.

3.3.1. Kinect v2

Segundo Polygon (2014), foi lançado no dia 15 de julho de 2014 o novo sensor Kinect One ou Kinect v2 substituto do Kinect 360, o novo dispositivo oferece uma melhor visualização do corpo a frente do dispositivo. Pois apresentou um grande aumento na resolução da câmera colorida para 1080p (HD), além do aumento da distância mínima e máxima de profundidade, o que não obriga o usuário estar sempre à frente do dispositivo.

O Kinect v2 é a última versão do sensor lançada no mercado pela *Microsoft* que conta com microfones e uma câmera de alta definição, consegue identificar mais articulações do que a versão anterior além de detectar batimentos cardíacos. Por meio de seus microfones o usuário consegue ativar o dispositivo por comandos de voz como, ligar, desligar abrir o aplicativo de “loja” e capturar imagem da tela.

Este trabalho utilizou os recursos do sensor para identificação das articulações do ombro e quadril que já são analisadas no módulo ADM da ferramenta FisoKinect e também a câmera do dispositivo para utilização do *screenshot* (captura de tela) para ativação da captura de tela do paciente durante a avaliação de ADM.

3.4. Softwares

Esta seção explana sobre a linguagem de programação e os softwares que foram utilizados para o desenvolvimento da aplicação proposta.

- C# (CSharp) - é a linguagem de programação desenvolvida pela Microsoft para a plataforma .NET. Possui tipagem dinâmica e estática, além de ser orientada a objetos; A linguagem foi utilizada para manipulação do código e envio de instruções

necessárias para a aplicação;

- Microsoft Windows 10 - atualmente é última versão do sistema operacional da Microsoft, usado em grande parte dos computadores pessoais; O sistema operacional foi utilizado para instalação da SDK, IDE e a plataforma *Vitruvius*;
- Software Development Kit (SDK) Kinect para Windows 2.0 - biblioteca para utilização dos recursos do dispositivo Kinect;
- Visual Studio 2015 - é a IDE oficial da Microsoft para desenvolvimento de softwares na plataforma .NET. Também é usada no desenvolvimento de aplicações na plataforma ASP.NET. A IDE possibilitou a utilização da linguagem C# (CSharp) para o desenvolvimento da aplicação;
- *Vitruvius* – plataforma desenvolvida pela Agência *LigthBuzz* para simplificação de implementação dos códigos de manipulação do sensor Kinect v2. Foi utilizada para cálculo dos ângulos de ADM e manuseio das imagens salvas no banco de dados da FísioKinect, tornando-as em *background* nas avaliações posteriores.

3.4.1. Vitruvius

A agência *LigthBuzz* especializada em realidade virtual, realidade aumentada e aplicações com o Kinect desenvolveu a ferramenta *Vitruvius* que contém um pacote de métodos, extensões e controles que permitem a simplificação de classes necessárias no desenvolvimento de projetos que utilizam o sensor Kinect (LIGTHBUZZ, 2015).

Em virtude da plataforma *Vitruvius* intermediar a implementação dos códigos exigidos pela SDK 2.0 simplifica o desenvolvimento de aplicações que utilizam o Kinect. Com essa facilitação de códigos o desenvolvedor tem um foco maior nos objetivos de seus projetos, não perdendo tempo com a programação básica e tem mais chances de expandir suas implementações.

Na versão 1 do Kinect os desenvolvedores tinham muita dificuldade na criação de novas ferramentas pois tinham que partir diretamente para a implementação básica e o *Vitruvius* encapsula a codificação básica permitindo aplicações mais rápidas e diretas.

Os recursos da plataforma já são utilizados no módulo ADM para obtenção dos cálculos das articulações e projeção dos arcos de identificação, exibindo em qual posição a articulação está sendo avaliada. *Vitruvius* beneficiou ainda este projeto permitindo a troca de plano de fundo da avaliação de ADM buscando a imagem capturada na avaliação do paciente armazenada no banco de dados da FísioKinect. Por exemplo, conforme o profissional seleciona uma articulação seja do ombro ou do quadril, a ferramenta apresenta a imagem

capturada do paciente anteriormente na ADM da articulação selecionada pelo profissional. Esta passa a ser exibida como *background* da avaliação, tornando-a assim como uma meta mínima de ADM a ser atingida pelo paciente em avaliações posteriores. Na figura 11 é apresentado um exemplo de como essa troca de plano de fundo ocorre com os recursos da plataforma Vitruvius.

Figura 11 - Troca de Background – Vitruvius



Fonte: Adaptado de PTERNEAS, (2014).

A figura 11 exemplifica a manipulação do *background* pelo *Vitruvius* que permite o controle do plano de fundo da imagem projetada pela câmera do Kinect v2, o que torna primordial a utilização dos recursos da plataforma para o desenvolvimento do acompanhamento de evolução do paciente. Na figura 11 apresentada o *background* no lado esquerdo é de cor verde, já no lado direito foi inserido uma imagem de um estádio como plano de fundo, justificando a utilização de *Vitruvius* na aplicação. As imagens do paciente capturadas e armazenadas no banco de dados poderão ser inseridas na avaliação possibilitando o acompanhamento da evolução no tratamento.

3.4.2. Desenvolvimento de Melhoria do Software

A melhoria realizada no sistema consistiu em manter o padrão dos trabalhos do GEPETS e, por se tratar de uma evolução, o trabalho foi realizado a partir da codificação do módulo ADM que realiza o cálculo da avaliação de ADM com os recursos de *Vitruvius*. A lógica da evolução foi permitir que o sensor Kinect v2 realizasse um *screenshot* para que esta captura fosse armazenada no banco de dados da ferramenta e pudesse ser reutilizada nas

avaliações posteriores. E, ao ser realizada uma nova avaliação do mesmo paciente o profissional de saúde pode fazer a solicitação ao módulo para que a imagem salva torne-se o plano de fundo dessa nova avaliação. O *screenshot* teve suas dimensões padronizadas para que o Kinect v2 capturasse apenas o que compõe a avaliação e com isso possibilitar que a imagem capturada, ao se tornar *background*, não invadisse os *groupbox* dos menus que são utilizados para CRUD (acrônimo de expressão do idioma inglês - Create (Criação), Retrieve (Consulta), Update (Atualização) e Delete (Destruição) definição das quatro operações básicas utilizadas em banco de dados relacionais) na avaliação. Para solicitação da imagem do paciente armazenada no banco de dados foi adicionado um botão com o nome de Plano de Fundo e, quando o profissional da saúde o seleciona, a imagem salva anteriormente transforma-se no plano de fundo da avaliação.

4 RESULTADOS E DISCUSSÕES

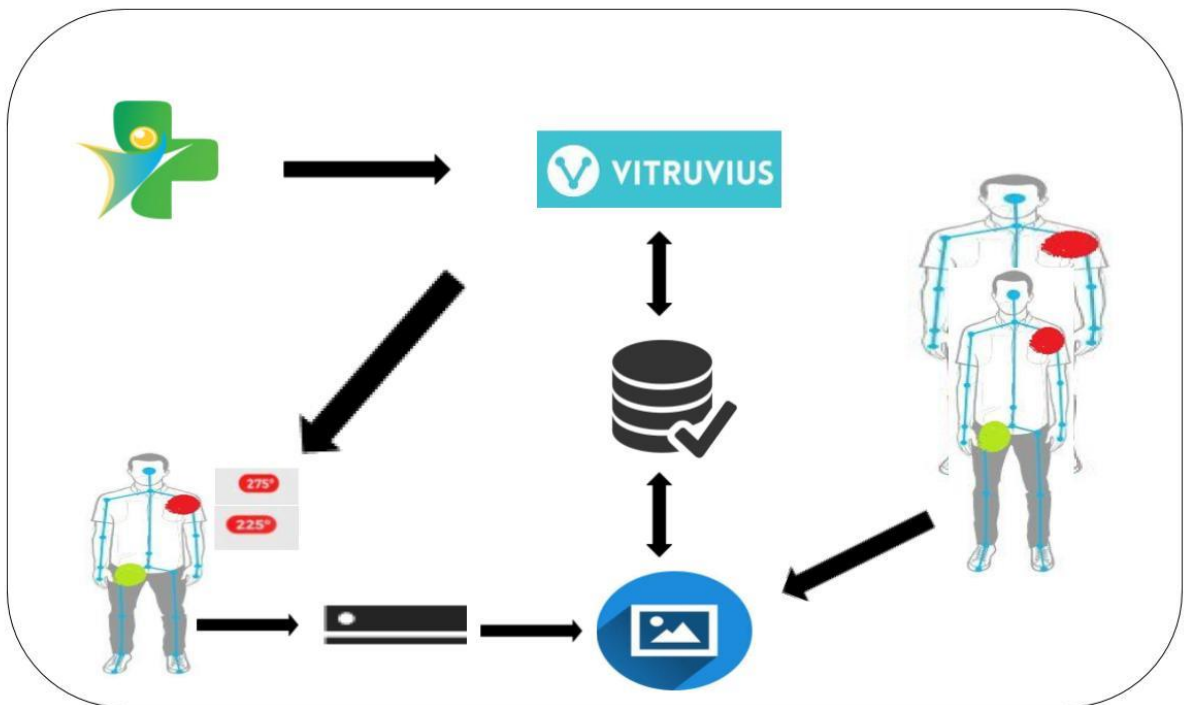
Nesta seção exibe os resultados obtidos no decorrer deste trabalho. A princípio será apresentada uma visão geral da ferramenta desenvolvida, bem como a arquitetura e funcionamento da mesma. Posteriormente, será apresentado a arquitetura do software e a codificação das principais funcionalidades da aplicação e por último o protocolo para a utilização da aplicação.

A aplicação desenvolvida tem como resultados: a avaliação de ADM com auxílio da câmera do dispositivo Kinect v2, por meio dos recursos da plataforma *Vitruvius*, os cálculos dos ângulos atingidos no movimento das articulações do ombro e quadril e a identificação dessas articulações durante o exame. E ainda a manipulação da imagem do paciente que fora capturada pela câmera do Kinect e armazenada no banco de dados da ferramenta FísioKinect, possibilitando que essa imagem seja utilizada na próxima avaliação do paciente como plano de fundo e cria-se uma interação entre o paciente e seus resultados.

4.1. Arquitetura

A arquitetura apresentada na figura 12 fornece uma visão geral do funcionamento do módulo ADM da FísioKinect e a melhoria de alteração de *background* proposta neste trabalho. Com isso, evidenciam-se as relações entre os processos de aquisição das imagens e a projeção das imagens como plano de fundo nas avaliações futuras.

Figura 12 - Arquitetura Módulo ADM Atualizado

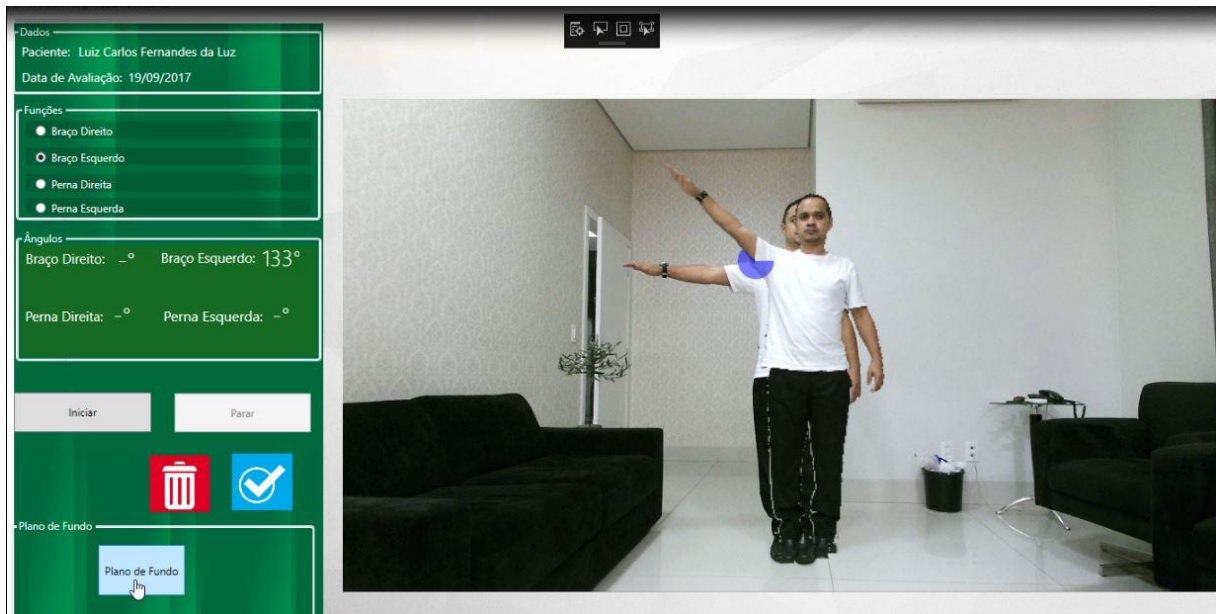


Conforme apresentado na figura 12 a arquitetura da melhoria no módulo ADM da FísioKinect faz a interligação do módulo com os recursos do Vitruvius gerando os *joints* com os arcos de identificação e também o cálculo dos ângulos visualizados pela câmera do sensor Kinect que realiza a captura da imagem e a armazena no banco de dados da aplicação. O módulo ainda utiliza os recursos de *Vitruvius* para utilização da imagem do paciente armazenada no banco de dados como *background* assim que solicitada por meio da aplicação, o que faz o paciente se projetar em sua própria imagem que está sendo exibida como plano de fundo.

4.2. Tela da aplicação que realiza o acompanhamento da evolução do paciente

O software trabalha com a manipulação de elementos do *XAML*, tendo sido elaborada a implementação para realizar o *screenshot* na execução da avaliação permitindo que esta captura possa ser utilizada em tempo real na avaliação. A interação do paciente com o plano de fundo organizado pela imagem contida no banco de dados utilizou recursos do *Vitruvius* para que, em tempo real, o paciente pudesse sobrepor sua própria silhueta. A figura 13 exibe como a sobreposição ou interação do paciente ocorre.

Figura 13 - Background Articulação do Ombro Esquerdo



A figura 13 apresenta a captura feita pela aplicação, ao ser selecionado o segmento do braço esquerdo no *groupbox* Funções. Ao clicar no botão plano de fundo dentro do *groupbox* Plano de Fundo, a ferramenta busca a imagem salva e a traz como *background* em tempo de execução da avaliação de ADM. Esse acompanhamento em tempo real proporciona motivação tanto para paciente quanto para o profissional da saúde aumentando as possibilidades de conclusão do tratamento.

4.3. Codificação da captura de tela

Para que o sensor Kinect capturasse a imagem por meio de sua câmera e a armazenasse no diretório da aplicação foi elaborada codificação que permite que, quando o usuário exercer o movimento de uma das articulações do ombro ou quadril, a imagem seja gravada na pasta da aplicação por meio do caminho informado no método de salvamento de imagem. A figura 14 exhibe o trecho do código do método que faz o salvamento da imagem capturada pelo dispositivo.

Figura 13 - Salvamento da imagem capturada

```
313 #region Captura de Tela.  
1 reference  
314 private void SaveBD()  
315 {  
316     String directoryName = new FileInfo(Assembly.GetExecutingAssembly().Location).DirectoryName;  
317  
318     string path = System.IO.Path.Combine(@"C:\Users\Luiz Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BraçoDireito.jpg");  
319  
320     (viewer.Image as WriteableBitmap).Save(path);  
321 }
```

Conforme exibido na figura 14 é demonstrado o código que realiza a armazenagem da articulação do braço direito, na linha 316 é criada uma variável do tipo `String` e de nome `directoryName` e é gerado um novo arquivo do tipo `FileInfo` que retorna o arquivo em execução por meio da classe `Assembly`. Na linha 318 foi criada uma variável `path` do tipo `string` contendo o caminho do diretório da aplicação que recebe a imagem capturada e por fim, na linha 320 o elemento do *XAML* `viewer` que é onde está posicionada a câmera do Kinect v2 e que, com isso, realiza a escrita da imagem por meio da classe `WriteableBitmap` no diretório informado.

4.4. Plano de Fundo

O plano de fundo é a transformação da imagem armazenada no diretório da aplicação como *background* da avaliação que é acionada por meio do clique ao botão Plano de Fundo. O código para que realize a troca de imagens do plano de fundo aconteça é apresentado na figura 15.

Figura 14 - Troca de *background* – FisioKinect ADM

```

378 |
379 | #region Troca de Plano de Fundo
    | 1 reference
380 | public void btnPlanodeFundo_Click(object sender, RoutedEventArgs e)
381 | {
382 |     btnPlanodeFundo_Clicado = true;
383 |
384 |     if (btnPlanodeFundo_Clicado)
385 |     {
386 |         _backgroundRemovalTool = new BackgroundRemovalTool(sensorKinect.CoordinateMapper);
387 |
388 |         remocao = sensorKinect.OpenMultiSourceFrameReader(FrameSourceTypes.Color | FrameSourceTypes.Depth | FrameSourceTypes.BodyIndex);
389 |         remocao.MultiSourceFrameArrived += Reader_RemoveMultiSourceFrameArrived;
390 |
391 |         playersController = new PlayersController();
392 |         playersController.BodyEntered += UserReporter_BodyEntered;
393 |         playersController.BodyLeft += UserReporter_BodyLeft;
394 |         playersController.Start();
395 |     }
396 |     if (radioButtonBD.IsChecked == true)
397 |     {
398 |         ImageBrush imgB = new ImageBrush();
399 |         BitmapImage bmpImg = new BitmapImage();
400 |         bmpImg.BeginInit();
401 |         bmpImg.UriSource = new Uri(@"C:\Users\Luiz Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BraçoDireito.jpg");
402 |         imgB.ImageSource = bmpImg;
403 |         Plano_Fundo.Background = imgB;
404 |         bmpImg.EndInit();
405 |     }

```

Conforme demonstrado na figura 15 quando o botão Plano de Fundo é clicado a variável global `btnPlanodeFundo_Clicado` na linha 382 recebe o valor `true` e entra na condição na linha 384 que ativa a variável `_backgroundRemovalTool` do tipo `BackgroundRemovalTool` na linha 386, classe esta que é recurso da plataforma *Vitruvius*. Na linha 388 e 389 os leitores de cor, profundidade e índices do corpo ativam o método `Reader_RemoveMultiSourceFrameArrived`.

Da linha 396 à linha 404 o condicional `if` faz a verificação se o segmento braço direito está selecionado, caso seja verdadeiro a aplicação manipula o elemento *XAML* `ImageBrush` para que através do parâmetro `UriSource` na linha 401 busque a imagem salva no diretório e a insira no canvas `Plano_Fundo` em tempo de execução.

4.5. Configuração da interação com o *background*

A interação com o *background* ocorre quando o paciente pode realizar a avaliação sobrepondo a sua silhueta do plano de fundo. Na figura 16 é exibido o trecho de código que permite esta interação.

Figura 15 - Interação do paciente com o *background*

```

75 void Reader_RemoveMultiSourceFrameArrived(object sender, MultiSourceFrameArrivedEventArgs e)
76 {
77     var reference = e.FrameReference.AcquireFrame();
78
79     using (var colorFrame = reference.ColorFrameReference.AcquireFrame())
80     using (var depthFrame = reference.DepthFrameReference.AcquireFrame())
81     using (var bodyIndexFrame = reference.BodyIndexFrameReference.AcquireFrame())
82     {
83         if (colorFrame != null && depthFrame != null && bodyIndexFrame != null)
84         {
85             // 3) Update the image source.
86             camera.Source = _backgroundRemovalTool.GreenScreen(colorFrame, depthFrame, bodyIndexFrame);
87         }
88     }
89 }
90

```

Fonte: Adaptado de PTERNEAS, (2014).

Na linha 83 é feita uma comparação dos leitores de cor, profundidade e corpo para verificar se não são nulos, ou seja, é necessário ter os leitores já inicializados para que a interação ocorra. Na linha 86 o elemento *XAML* *camera* recebe a variável *_backgroundRemovalTool* com o parâmetro *GreenScreen* que permite a interação do paciente no *background* da avaliação de ADM.

4.6. Relatório dos resultados da avaliação de ADM

A ferramenta disponibiliza os resultados dos ângulos e das imagens no formato de relatório para um acompanhamento mais detalhado das avaliações. O profissional da saúde e o paciente têm a opção de verificar as ADM por meio do relatório com os valores dos ângulos e com as imagens capturadas. A figura 17 apresenta o relatório que é fornecido pela aplicação

para verificação dos resultados.

Figura 16 - Relatório de resultados



Nome	Data Avaliação	Braço Direito	Imagem Braço Direito	Braço Esquerdo	Imagem Braço Esquerdo	Perna Direita	Imagem Perna Direita	Perna Esquerda	Imagem Perna Esquerda
Pedro Rodrigues Cardoso	04/12/2017	57		47		16		16	
Luiz Carlos Fernandes da Luz	04/12/2017	74		33		3		12	
Pedro Rodrigues Cardoso	04/12/2017	83		84		34		50	

O relatório de resultados apresentado na figura 17 traz os resultados do paciente obtido contendo tanto os valores dos ângulos atingidos quanto as imagens capturadas que estão armazenadas em diretório da aplicação sendo que apenas o caminho da imagem é armazenado no banco de dados da ferramenta. Mostra ainda o nome do paciente, a data da avaliação e o valor do ângulo seguidamente da imagem recolhida durante a avaliação de ADM, para que o paciente e o profissional da saúde possam aprimorar os exercícios para evolução do tratamento.

Estes novos tipos de resultados adquiridos nessa melhoria da ferramenta abrangem maior compreensão, pois os dados numéricos não são tão entendíveis quanto os dados visuais, com isso se permite um acompanhamento mais amplo do que a versão anterior da aplicação.

4.7. Protocolo de Utilização

Para utilizar o sistema deve-se seguir algumas premissas básicas. Conforme Microsoft (2012), o sensor deve estar numa altura máxima de 1,80m e mínima de 60cm do chão, e a uma distância mínima de 1,40m do usuário. O espaço de jogo deve ser de no mínimo 1,80m, mas não pode ter mais de 3,60m. O campo de visão do novo Kinect é 60% maior do que seu antecessor.

Durante a avaliação é necessário que o paciente esteja a uma distância mínima de 1,40m para que a silhueta do paciente não se perca do campo de visão do Kinect v2 e com isso a avaliação seja comprometida. É aconselhável que haja uma marca no chão para que o paciente posicione-se sempre no mesmo local para facilitar a sobreposição de imagens, o profissional da saúde orienta os movimentos de abdução e adução das articulações do ombro e quadril a serem exercidos pelo paciente para que a ferramenta calcule o ângulo de ADM.

No período de avaliação o profissional pode obter a captura da imagem do paciente ao selecionar a próxima articulação a ser avaliada marcando um dos segmentos. Na avaliação seguinte quando o profissional da saúde selecionar a articulação desejada para comparativo e clicar no botão Plano de Fundo a imagem salva na aplicação torna-se o *background* da avaliação servindo de conferência com o ângulo atingido na avaliação anterior.

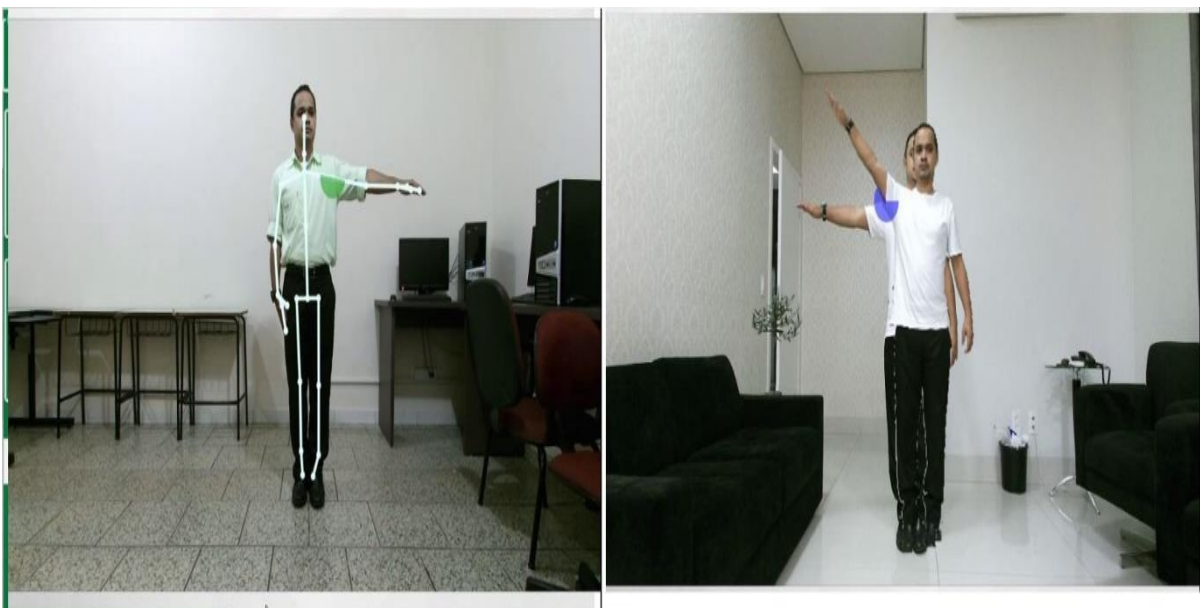
5 CONSIDERAÇÕES FINAIS

O presente trabalho teve como objetivo realizar uma melhoria no módulo de ADM da ferramenta FisioKinect tornando a própria avaliação do paciente uma motivação para conclusão do tratamento, bem como o seu acompanhamento tanto pelo profissional da saúde quanto do paciente, com intuito de incentivo e fazer com que o paciente também participasse de forma ativa do tratamento. Os trabalhos correlatos fundamentaram o objetivo de utilizar imagens para exibição de resultados, os quais são importantes para compreensão dos pacientes, em que os resultados da versão anterior era apenas em valores numéricos e em alguns casos eram entendidos apenas pelo profissional da saúde.

Por meio da utilização da câmera do dispositivo Kinect v2 e recursos da plataforma *Vitruvius*, foi possível fazer com que a imagem do paciente fosse capturada durante a avaliação de ADM e esta imagem se tornasse o *background* das avaliações posteriores permitindo, com isso, obter comparativos dos ângulos atingidos durante o tratamento. Com isso pode-se afirmar que o *Vitruvius* foi indispensável para idealização deste projeto e para a sua conclusão.

Através de um comparativo das versões das avaliações de ADM foi possível identificar uma melhoria significativa para a avaliação de ADM da ferramenta FisioKinect e o acompanhamento da evolução. Na figura 18 exibe este comparativo.

Figura 17 - Comparativo de avaliações ADM



Na figura à esquerda a avaliação demonstra o cálculo da ADM e a articulação é identificada por meio de arcos de identificação e é projetada o esqueleto de *joints* sobre o corpo do paciente. Na figura do lado direito exibe o novo tipo de avaliação que permite o acompanhamento por meio de imagens sobrepostas de avaliações anteriores.

O paciente também pode contribuir no seu tratamento a partir do acompanhamento via imagem, o que pode colaborar com a diminuição de desistências no decurso do tratamento. A aplicação desenvolvida estreita a comunicação entre o profissional da saúde e o paciente de forma que os dois busquem o êxito a partir da observação em comum das imagens capturadas da ferramenta. Projetos futuros que terão como base este trabalho poderão trabalhar a utilização dos microfones do dispositivo Kinect para o reconhecimento de voz e assim realizar a captura de imagem por meio do comando de voz, otimizando ainda mais o processo de avaliação de ADM.

Assim, conclui-se que é possível gerar uma motivação e acompanhamento da avaliação de ADM, por meio da captura de imagem do paciente durante a execução do exame e esta interação é proporcionada pela própria imagem capturada como *background* da avaliação seguinte.

REFERÊNCIAS

BABOO. **Versão Final do Kinect SDK 2.0 para Windows já está disponível.** Disponível em: <<https://www.baboo.com.br/software/versao-final-do-kinect-sdk-2-0-para-windows/>>. Acesso em: 29 jan. 2017.

DUARTE, Débora. **Biomecânica do quadril.** Disponível em: <<http://massoterapiadeboraduarte.blogspot.com.br/2011/04/biomecanica-do-quadril.html>>. Acesso em: 10 nov. 2016.

INFORMÁTICA, Timax. **Medifisio.** 2010. Disponível em: <<http://medifisio.com.br/>>. Acesso em: 14 jun. 2017.

LIGHTBUZZ. **Vitruvius.** 2015. Disponível em: <<http://vitruviuskinect.com/>>. Acesso em: 30 jan. 2017.

MARQUES, Amelia Pasqual. **Manual de Goniometria.** 3. ed. Barueri: Manole Ltda, 2014. 136 p. Disponível em: <<http://ulbra.bv3.digitalpages.com.br/users/publications/9788520438978/pages/136>>. Acesso em: 01 mar. 2017.

MICROSOFT. **Kinect for Xbox One.** 2017. Disponível em: <<http://www.xbox.com/en-US/xbox-one/accessories/kinect>>. Acesso em: 28 dez. 2016.

MICROSOFT. **Kinect for Windows SDK.** Disponível em: <<https://msdn.microsoft.com/pt-br/library/dn799271.aspx>>. Acesso em: 20 mar. 2017.

POLYGON. **Kinect for Windows v2 to launch july 15.** Disponível em: <<http://www.polygon.com/2014/7/7/5876283/report-kinect-for-windows-v2-to-launching-july-15>>. (2014). Acesso em 10 de Novembro de 2016.

PEREIRA, Sergio Luiz. **Processos produtivos necessidade de padronização entre as culturas acadêmica e empresarial.** São Paulo: Leitura, 2013. Color. Disponível em: <https://edisciplinas.usp.br/pluginfile.php/122832/mod_resource/content/1/PDF_PEA_2450_Automacao_2013.pdf>. Acesso em: 04 maio 2017.

PTERNEAS, Vangos. **Background Removal Using Kinect 2 (Green Screen Effect).** 2014. Disponível em: <<http://pterneas.com/2014/04/11/kinect-background-removal/>>. Acesso em: 11 abr. 2017.

ROSA FILHO, Blair Jose. **Movimentos de Abdução e Adução.** 2002. Disponível em: <http://www.wgate.com.br/conteudo/medicinaesaude/fisioterapia/cinesio/abducao_aducao.htm>. Acesso em: 03 abr. 2017.

SALDANHA, Rosely. **Membros Superiores – Cintura Escapular.** 2011. Disponível em: <<https://roselysaldanha.wordpress.com/category/membros-superiores-cintura-escapular/>>. Acesso em: 10 mar. 2017.

SANTOS, Maria. **Planos e eixos.** 2012. Disponível em:

<<http://biomecanicaeoserhumano.blogspot.com.br/2012/10/planos-e-eixos.html>>. Acesso em: 06 mar. 2017.

SEBASTIÃO, Silvana. **Planos Anatômicos**. 2012. Disponível em: <<http://www.institutodehipnose.com/2012/04/planos-anatomicos.html>>. Acesso em: 15 mar. 2017.

CAMARGO, Thayso Henrique Capuchinho. **AVALIAÇÃO DA AMPLITUDE DE MOVIMENTO (ADM) EM PLANO CORONAL ANTERIOR UTILIZANDO O SENSOR KINECT: articulações do ombro e do quadril**. 2015. 85 f. Monografia (Especialização) - Curso de Ciências da Computação, Centro Universitário Luterano de Palmas, Palmas, 2015.

SOUZA, Leomar Camargo de. **Avaliação antropométrica através da delimitação da silhueta corporal utilizando o Kinect**. 2015. 85 f. TCC (Graduação) - Curso de Ciências da Computação, Centro Universitário Luterano de Palmas, Palmas, 2015.

LUZ, Luiz Carlos Fernandes da. **ADAPTAÇÃO DO SISTEMA DE AVALIAÇÃO DA AMPLITUDE DE MOVIMENTO (ADM) EM PLANO CORONAL ANTERIOR PARA ARTICULAÇÕES DE OMBRO E QUADRIL UTILIZANDO O SENSOR KINECT V2**. Palmas: Ceulp, 2016.

TERRAZUL INFORMÁTICA. **Physical Test**. 1992. Disponível em: <<http://www.terrazul.com.br/site/produtos/16/Software-de-Avaliação-Física/Physical-Test.>>. Acesso em: 18 mar. 2017.

APÊNDICES

APÊNDICE A – Código fonte da tela principal – Fisiokinect

```

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Windows;
using System.Windows.Controls;
using System.Windows.Data;
using System.Windows.Documents;
using System.Windows.Input;
using System.Windows.Media;
using System.Windows.Media.Imaging;
using System.Windows.Navigation;
using System.Windows.Shapes;
using Fisiok.Manager;
using Fisiok.WindowManager;
using Fisiok.Relatorios;
using System.Windows.Forms;
using KeyEventArgs = System.Windows.Input.KeyEventArgs;
using MessageBox = System.Windows.MessageBox;

namespace Fisiok
{
    /// <summary>
    /// Interaction logic for MainWindow.xaml
    /// </summary>
    public partial class MainWindow : Window
    {
        AppInfo app = new AppInfo();
        public MainWindow()
        {
            InitializeComponent();
        }

        private void Window_Loaded(object sender, RoutedEventArgs e)
        {
            //criarPasta();
            CarregarItens();
        }

        private void CarregarItens()
        {
            try
            {
                dataGridPacientes.Items.Clear();
                List<Paciente> listaP = ManagerPaciente.Listar();
                foreach (Paciente p in listaP)
                {
                    this.dataGridPacientes.Items.Add(p);
                }
            }
            catch (Exception ex)
            {
                System.Windows.Forms.MessageBox.Show(ex.Message, @"Fisiokinect",
                MessageBoxButton.OK, MessageBoxIcon.Warning);
                Close();
            }
        }

        private void CarregarItens(string val)
        {
            dataGridPacientes.Items.Clear();

```

```

        List<Paciente> listaP = ManagerPaciente.Listar(val);
        foreach (Paciente p in listaP)
        {
            this.dataGridPacientes.Items.Add(p);
        }
    }

    private void AmplitudeMovimento_Click(object sender, RoutedEventArgs e)
    {
        if (dataGridPacientes.SelectedItem == null)
        {
            MessageBox.Show("Selecione um paciente para entrar no módulo!",
                app.Titulo, MessageBoxButton.OK, MessageBoxImage.Information);
            return;
        }
        Paciente paciente = (Paciente)dataGridPacientes.SelectedItem;
        WindowAmplitudemovimento adm = new WindowAmplitudemovimento(paciente);
        adm.ShowDialog();
    }

    private void Antropometria_Click(object sender, RoutedEventArgs e)
    {
    }

    private void InclinaoLateralTronco_Click(object sender, RoutedEventArgs e)
    {
    }

    private void Hiperlordose_Click(object sender, RoutedEventArgs e)
    {
    }

    private void Percepcao_Click(object sender, RoutedEventArgs e)
    {
    }

    private void EscalaSilhueta_OnClick(object sender, RoutedEventArgs e)
    {
    }

    private void Info_Click(object sender, RoutedEventArgs e)
    {
        new WindowInfoSistema().ShowDialog();
    }

    private void ButtonCadastrar_Click(object sender, RoutedEventArgs e)
    {
        new WindowPaciente(null).ShowDialog();
        CarregarItens();
    }

    private void ButtonRelatorio_Click(object sender, RoutedEventArgs e)
    {
        new WindowRelatorio().ShowDialog();
    }

    private void ButtonPesquisar_Click(object sender, RoutedEventArgs e)
    {
        CarregarItens(txtPesquisar.Text);
    }

```



```

}

private void ButtonEditar_Click(object sender, RoutedEventArgs e)
{
    Paciente paciente = (Paciente)dataGridPacientes.SelectedItem;
    new WindowPaciente(paciente).ShowDialog();
    BtnEditar.IsEnabled = false;
    BtnVisualizar.IsEnabled = false;
    BtnExcluir.IsEnabled = false;
    CarregarItens();
}

private void ButtoVisualizar_Click(object sender, RoutedEventArgs e)
{
    Paciente paciente = (Paciente)dataGridPacientes.SelectedItem;
    new WindowVisualizar(paciente).ShowDialog();
    BtnEditar.IsEnabled = false;
    BtnVisualizar.IsEnabled = false;
    BtnExcluir.IsEnabled = false;
    CarregarItens();
}

private void ButtonExcluir_Click(object sender, RoutedEventArgs e)
{
    try
    {
        if (dataGridPacientes.SelectedItem == null)
        {
            MessageBox.Show("Selecione o Paciente a ser excluído!",
app.Titulo, MessageBoxButton.OK, MessageBoxImage.Information);
            return;
        }
        MessageBoxResult result = MessageBox.Show("Tem certeza que deseja
excluir o paciente?\nEsta ação acarretará na exclusão de todos os registros do
paciente.", app.Titulo, MessageBoxButton.YesNo, MessageBoxImage.Question);
        if (result == MessageBoxResult.Yes)
        {
            Paciente paciente = (Paciente)dataGridPacientes.SelectedItem;
            ManagerPaciente.Excluir(paciente.Id);
            CarregarItens();
            BtnEditar.IsEnabled = false;
            BtnVisualizar.IsEnabled = false;
            BtnExcluir.IsEnabled = false;
        }
        MessageBox.Show("Paciente excluído com sucesso!", app.Titulo,
MessageBoxButton.OK, MessageBoxImage.Information);
    }
    catch (Exception ex)
    {
        MessageBox.Show(ex.Message, app.Titulo, MessageBoxButton.OK,
MessageBoxImage.Error);
    }
}

private void DataGrid_MouseLeftButtonUp(object sender, MouseButtonEventArgs e)
{
    if (dataGridPacientes.SelectedItem != null)
    {
        BtnEditar.IsEnabled = true;
        BtnVisualizar.IsEnabled = true;
        BtnExcluir.IsEnabled = true;
        Paciente paciente = (Paciente)dataGridPacientes.SelectedItem;
    }
}

```

```
    }  
  
    private void txtPesquisar_KeyDown(object sender, KeyEventArgs e)  
    {  
        if (e.Key == Key.Enter)  
        {  
            CarregarItens(txtPesquisar.Text);  
        }  
    }  
  
    private void Window_Closed(object sender, EventArgs e)  
    {  
        Close();  
    }  
} }  
}
```

APÊNDICE B – Código fonte da tela inicial Módulo ADM

```

using LightBuzz.Vitruvius;
using Microsoft.Kinect;
using System.Windows;
using FisioK.Manager;
using System;
using System.Windows.Controls;
using System.Windows.Media.Imaging;
using System.IO;
using System.Reflection;
using System.Data;
using System.Data.SqlClient;
using System.Windows.Media;
using KinectBackgroundRemoval;
using System.Web;
using System.Configuration;
using System.Drawing;

namespace FisioK
{
    public partial class WindowAmplitudemovimento : Window
    {
        #region Variáveis
        AppInfo app = new AppInfo();
        Paciente paciente;

        string ImgBracoD;
        string ImgBracoE;
        string ImgPernaD;
        string ImgPernaE;

        KinectSensor sensorKinect;
        MultiSourceFrameReader leitura;
        MultiSourceFrameReader remocao;
        PlayersController playersController;
        BackgroundRemovalTool _backgroundRemovalTool;

        private bool btnPlanodeFundo_Clicado = false;

        JointType inicio1 = JointType.ElbowRight;
        JointType centro1 = JointType.ShoulderRight;
        JointType termino1 = JointType.SpineShoulder;

        JointType inicio2 = JointType.SpineShoulder;
        JointType centro2 = JointType.ShoulderLeft;
        JointType termino2 = JointType.ElbowLeft;

        JointType inicio3 = JointType.KneeRight;
        JointType centro3 = JointType.HipRight;
        JointType termino3 = JointType.SpineBase;

        JointType inicio4 = JointType.SpineBase;
        JointType centro4 = JointType.HipLeft;
        JointType termino4 = JointType.KneeLeft;

        #endregion

        public WindowAmplitudemovimento(Paciente paciente)
        {
            InitializeComponent();
            IniciarKinect();
        }
    }
}

```

```

        this.paciente = paciente;
        dataAvaliacao.Content = string.Format("{0:dd/MM/yyyy}", DateTime.Now);
        nomePaciente.Content = paciente.Nome;
    }

#region Inicializador Kinect

    public void IniciarKinect()
    {
        sensorKinect = KinectSensor.Default();

        if (sensorKinect != null)
        {
            sensorKinect.Open();

            leitura =
            sensorKinect.OpenMultiSourceFrameReader(FrameSourceTypes.Color
            | FrameSourceTypes.Depth | FrameSourceTypes.Infrared |
            FrameSourceTypes.Body);
            leitura.MultiSourceFrameArrived += Reader_MultiSourceFrameArrived;

            playersController = new PlayersController();
            playersController.BodyEntered += UserReporter_BodyEntered;
            playersController.BodyLeft += UserReporter_BodyLeft;
            playersController.Start();

        }
    }

    void Reader_RemoveMultiSourceFrameArrived(object sender,
    MultiSourceFrameArrivedEventArgs e)
    {
        var reference = e.FrameReference.AcquireFrame();

        using (var colorFrame = reference.ColorFrameReference.AcquireFrame())
        using (var depthFrame = reference.DepthFrameReference.AcquireFrame())
        using (var bodyIndexFrame =
        reference.BodyIndexFrameReference.AcquireFrame())
        {
            if (colorFrame != null && depthFrame != null && bodyIndexFrame !=
            null)
            {
                // 3) Update the image source.
                camera.Source = _backgroundRemovalTool.GreenScreen(colorFrame,
                depthFrame, bodyIndexFrame);
            }
        }
    }
}
#endregion

#region Leitores ADM
    void Reader_MultiSourceFrameArrived(object sender,
    MultiSourceFrameArrivedEventArgs e)
    {
        var reference = e.FrameReference.AcquireFrame();

        // Cor
        using (var frame = reference.ColorFrameReference.AcquireFrame())
        {
            if (frame != null)
            {
                if (viewer.Visualization == Visualization.Color)
                {
                    viewer.Image = frame.ToBitmap();
                }
            }
        }
    }
}

```

```

    }
  }
}

//Corpo
using (var frame = reference.BodyFrameReference.AcquireFrame())
{
  if (frame != null)
  {
    var bodies = frame.Bodies();
    playersController.Update(bodies);

    Body body = bodies.Closest();

    if (body != null)
    {
      viewer.DrawBody(body);

      if (radioButtonBD.IsChecked == true)
      {
        angulo1.Update(body.Joints[inicio1], body.Joints[centro1],
body.Joints[termino1], 40);
        tblAngulo1.Text = ((int)angulo1.Angle - 180 /
2).ToString();
      }
      else
      {
        if (radioButtonBE.IsChecked == true)
        {
          angulo2.Update(body.Joints[inicio2],
body.Joints[centro2], body.Joints[termino2], 40);
          tblAngulo2.Text = ((int)angulo2.Angle - 180 /
2).ToString();
        }
        else
        {
          if (radioButtonPD.IsChecked == true)
          {
            angulo3.Update(body.Joints[inicio3],
body.Joints[centro3], body.Joints[termino3], 40);
            tblAngulo3.Text = ((int)angulo3.Angle - 360 /
4).ToString();
          }
          else if (radioButtonPE.IsChecked == true)
          {
            angulo4.Update(body.Joints[inicio4],
body.Joints[centro4], body.Joints[termino4], 40);
            tblAngulo4.Text = ((int)angulo4.Angle - 360 /
4).ToString();
          }
        }
      }
    }
  }
}
}

}
#endregion

```

```

#region Campos Vitruvius
void UserReporter_BodyEntered(object sender, PlayersControllerEventArgs e)
{
}
/*Limpa os campos de ângulo*/
void UserReporter_BodyLeft(object sender, PlayersControllerEventArgs e)
{
    viewer.Clear();
    angulo1.Clear();
    angulo2.Clear();
    angulo3.Clear();
    angulo4.Clear();

    tblAngulo1.Text = "-";
    tblAngulo2.Text = "-";
    tblAngulo3.Text = "-";
    tblAngulo4.Text = "-";
}
#endregion

#region Iniciar
private void btnIniciar_Click(object sender, RoutedEventArgs e)
{
    if (radioButtonBD.IsChecked == true)
    {
        btnIniciar.IsEnabled = false;
        SaveBD();
        btnParar.IsEnabled = true;
    }
    if (radioButtonBE.IsChecked == true)
    {
        btnIniciar.IsEnabled = false;
        SaveBE();
        btnParar.IsEnabled = true;
    }
    if (radioButtonPD.IsChecked == true)
    {
        btnIniciar.IsEnabled = false;
        SavePD();
        btnParar.IsEnabled = true;
    }
    if (radioButtonPE.IsChecked == true)
    {
        btnIniciar.IsEnabled = false;
        SavePE();
        btnParar.IsEnabled = true;
    }

    //MessageBox.Show("Selecione uma articulação a ser feita a avaliação!",
    app.Titulo, MessageBoxButton.OK, MessageBoxImage.Information);
}
#endregion

#region Parar
private void btnParar_Click(object sender, RoutedEventArgs e)
{
    btnIniciar.IsEnabled = true;
    btnParar.IsEnabled = false;

    if (radioButtonBD.IsChecked == true)

```

```

{
    tblAngulo1.Text = ((int)angulo1.Angle - 180 / 2).ToString();
}
else
{
    if (radioButtonBE.IsChecked == true)
    {
        tblAngulo2.Text = ((int)angulo2.Angle - 180 / 2).ToString();
    }
    else
    {
        if (radioButtonPD.IsChecked == true)
        {
            tblAngulo3.Text = ((int)angulo3.Angle - 360 / 4).ToString();
        }

        else if (radioButtonPE.IsChecked == true)
        {
            tblAngulo4.Text = ((int)angulo4.Angle - 360 / 4).ToString();
        }
    }
    else
    {
        MessageBox.Show("A avaliação foi concluída!", app.Titulo,
        MessageBoxButton.OK, MessageBoxImage.Information);
    }
}
}
}

}
#endregion

#region Cancelar
private void btnCancel_Click(object sender, RoutedEventArgs e)
{
    LimparCampos();
}
#endregion

#region Salvar
private void btnSalvar_Click(object sender, RoutedEventArgs e)
{
    try
    {
        string segmento = "";
        if (radioButtonBD.IsChecked == true)
        {
            segmento = "Braço Direito";
        }
        else if (radioButtonBE.IsChecked == true)
        {
            segmento = "Braço Esquerdo";
        }
        else if (radioButtonPD.IsChecked == true)
        {
            segmento = "Perna Direita";
        }
        else if (radioButtonPE.IsChecked == true)
        {
            segmento = "Perna Esquerda";
        }
    }
}
}

```

```

    }
    else
    {
        MessageBox.Show("selecione o segmento a ser salvo!", app.Titulo,
        MessageBoxButton.OK, MessageBoxImage.Information);
        return;
    }
    ManagerAvaliacaoADMSistema.Adicionar(DateTime.Now, segmento,
    paciente.Id, tblAngulo1.Text, tblAngulo2.Text, tblAngulo3.Text, tblAngulo4.Text,
    ImgBracoD, ImgBracoE, ImgPernaD, ImgPernaE);
    MessageBox.Show("Avaliação de Amplitude de movimento foi salva com
    sucesso!", app.Titulo, MessageBoxButton.OK, MessageBoxImage.Information);
    LimparCampos();
}
catch
{
    MessageBox.Show("Erro ao adicionar Avaliação de amplitude de
    movimento!", app.Titulo, MessageBoxButton.OK, MessageBoxImage.Information);
}
}
#endregion
#region Captura de Tela
private void SaveBD()
{
    string caminho;
    string path = System.IO.Path.Combine(@"C:\Users\Luiz
    Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BracoDireito_" +
    this.paciente.Id + ".jpg");

    (viewer.Image as WriteableBitmap).Save(path);

    caminho = @"C:\Users\Luiz
    Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BracoDireito_" +
    this.paciente.Id + ".jpg";
    this.ImgBracoD = caminho;
}
private void SaveBE()
{
    string caminho;
    string path = System.IO.Path.Combine(@"C:\Users\Luiz
    Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BracoEsquerdo_" +
    this.paciente.Id + ".jpg");

    (viewer.Image as WriteableBitmap).Save(path);

    caminho = @"C:\Users\Luiz
    Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BracoEsquerdo_" +
    this.paciente.Id + ".jpg";
    this.ImgBracoE = caminho;
}
private void SavePD()
{
    string caminho;
    string path = System.IO.Path.Combine(@"C:\Users\Luiz
    Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\PernaDireita_" +
    this.paciente.Id + ".jpg");

    (viewer.Image as WriteableBitmap).Save(path);

    caminho = @"C:\Users\Luiz
    Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\PernaDireita_" +
    this.paciente.Id + ".jpg";
}

```



```

        this.ImgPernaD = caminho;
    }
    private void SavePE()
    {
        string caminho;
        string path = System.IO.Path.Combine(@"C:\Users\Luiz
Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\PernaEsquerda_" +
this.paciente.Id + ".jpg");

        (viewer.Image as WriteableBitmap).Save(path);

        caminho = @"C:\Users\Luiz
Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\PernaEsquerda_" +
this.paciente.Id + ".jpg";
        this.ImgPernaE = caminho;
    }
}
#endregion

#region LimparCampos
private void LimparCampos()
{
    btnIniciar.IsEnabled = true;
    btnParar.IsEnabled = false;
    radioButtonBD.IsChecked = false;
    radioButtonBE.IsChecked = false;
    radioButtonPD.IsChecked = false;
    radioButtonPE.IsChecked = false;
    tblAngulo1.Text = "-";
    tblAngulo2.Text = "-";
    tblAngulo3.Text = "-";
    tblAngulo4.Text = "-";
}
#endregion

#region Fechamento
private void Window_Closed(object sender, EventArgs e)
{
    if (playersController != null)
    {
        playersController.Stop();
    }

    if (leitura != null)
    {
        leitura.Dispose();
    }

    if (sensorKinect != null)
    {
        sensorKinect.Close();
    }
}
#endregion

#region Troca de Plano de Fundo
public void btnPlanodeFundo_Click(object sender, RoutedEventArgs e)
{
    btnPlanodeFundo_Clicado = true;

    if (btnPlanodeFundo_Clicado)
    {

```

```

        _backgroundRemovalTool = new
BackgroundRemovalTool(sensorKinect.CoordinateMapper);

        remocao =
sensorKinect.OpenMultiSourceFrameReader(FrameSourceTypes.Color |
FrameSourceTypes.Depth | FrameSourceTypes.BodyIndex);
        remocao.MultiSourceFrameArrived +=
Reader_RemoveMultiSourceFrameArrived;

        playersController = new PlayersController();
        playersController.BodyEntered += UserReporter_BodyEntered;
        playersController.BodyLeft += UserReporter_BodyLeft;
        playersController.Start();
    }
    if (radioButtonBD.IsChecked == true)
    {
        ImageBrush imgB = new ImageBrush();
        BitmapImage btpImg = new BitmapImage();
        btpImg.BeginInit();
        btpImg.UriSource = new Uri(@"C:\Users\Luiz
Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BracoDireito_" +
this.paciente.Id + ".jpg");
        imgB.ImageSource = btpImg;
        Plano_Fundo.Background = imgB;
        btpImg.EndInit();
    }
    if (radioButtonBE.IsChecked == true)
    {
        ImageBrush imgB = new ImageBrush();
        BitmapImage btpImg = new BitmapImage();
        btpImg.BeginInit();
        btpImg.UriSource = new Uri(@"C:\Users\Luiz
Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\BracoEsquerdo_" +
this.paciente.Id + ".jpg");
        imgB.ImageSource = btpImg;
        Plano_Fundo.Background = imgB;
        btpImg.EndInit();
    }
    if (radioButtonPD.IsChecked == true)
    {
        ImageBrush imgB = new ImageBrush();
        BitmapImage btpImg = new BitmapImage();
        btpImg.BeginInit();
        btpImg.UriSource = new Uri(@"C:\Users\Luiz
Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\PernaDireita_" +
this.paciente.Id + ".jpg");
        imgB.ImageSource = btpImg;
        Plano_Fundo.Background = imgB;
        btpImg.EndInit();
    }
    if (radioButtonPE.IsChecked == true)
    {
        ImageBrush imgB = new ImageBrush();
        BitmapImage btpImg = new BitmapImage();
        btpImg.BeginInit();
        btpImg.UriSource = new Uri(@"C:\Users\Luiz
Luz\Downloads\FisioK\FisioK\FisioK\Imagens\ADMSistema\PernaEsquerda_" +
this.paciente.Id + ".jpg");
        imgB.ImageSource = btpImg;
        Plano_Fundo.Background = imgB;
        btpImg.EndInit();
    }
}
}

```

```
#endregion

private void Window_Loaded(object sender, RoutedEventArgs e)
{

}
}
```

APÊNDICE C – Código fonte da classe de Background

```

using Microsoft.Kinect;
using System;
using System.Collections.Generic;
using System.Linq;
using System.Runtime.InteropServices;
using System.Text;
using System.Windows;
using System.Windows.Media;
using System.Windows.Media.Imaging;

namespace KinectBackgroundRemoval
{
    public class BackgroundRemovalTool
    {
        #region Constants

        readonly double DPI = 96.0;

        readonly PixelFormat FORMAT = PixelFormats.Bgra32;

        readonly int BYTES_PER_PIXEL = (PixelFormat.Bgr32.BitsPerPixel + 7) / 8;

        #endregion

        #region Membros

        WriteableBitmap _bitmap = null;

        ushort[] _depthData = null;

        byte[] _bodyData = null;

        byte[] _colorData = null;

        byte[] _displayPixels = null;

        ColorSpacePoint[] _colorPoints = null;

        CoordinateMapper _coordinateMapper = null;

        #endregion

        #region Construtor

        public BackgroundRemovalTool(CoordinateMapper mapper)
        {
            _coordinateMapper = mapper;
        }

        #endregion

        #region Métodos

        public BitmapSource GreenScreen(ColorFrame colorFrame, DepthFrame depthFrame,
        BodyIndexFrame bodyIndexFrame)
        {
            int colorWidth = colorFrame.FrameDescription.Width;
            int colorHeight = colorFrame.FrameDescription.Height;

            int depthWidth = depthFrame.FrameDescription.Width;
            int depthHeight = depthFrame.FrameDescription.Height;

```

```

int bodyIndexWidth = bodyIndexFrame.FrameDescription.Width;
int bodyIndexHeight = bodyIndexFrame.FrameDescription.Height;

if (_displayPixels == null)
{
    _depthData = new ushort[depthWidth * depthHeight];
    _bodyData = new byte[depthWidth * depthHeight];
    _colorData = new byte[colorWidth * colorHeight * BYTES_PER_PIXEL];
    _displayPixels = new byte[depthWidth * depthHeight * BYTES_PER_PIXEL];
    _colorPoints = new ColorSpacePoint[depthWidth * depthHeight];
    _bitmap = new WriteableBitmap(depthWidth, depthHeight, DPI, DPI,
FORMAT, null);
}

if (((depthWidth * depthHeight) == _depthData.Length) && ((colorWidth *
colorHeight * BYTES_PER_PIXEL) == _colorData.Length) && ((bodyIndexWidth *
bodyIndexHeight) == _bodyData.Length))
{
    depthFrame.CopyFrameDataToArray(_depthData);

    if (colorFrame.RawColorImageFormat == ColorImageFormat.Bgra)
    {
        colorFrame.CopyRawFrameDataToArray(_colorData);
    }
    else
    {
        colorFrame.CopyConvertedFrameDataToArray(_colorData,
ColorImageFormat.Bgra);
    }

    bodyIndexFrame.CopyFrameDataToArray(_bodyData);

    _coordinateMapper.MapDepthFrameToColorSpace(_depthData, _colorPoints);

    Array.Clear(_displayPixels, 0, _displayPixels.Length);

    for (int y = 0; y < depthHeight; ++y)
    {
        for (int x = 0; x < depthWidth; ++x)
        {
            int depthIndex = (y * depthWidth) + x;

            byte player = _bodyData[depthIndex];

            if (player != 0xff)
            {
                ColorSpacePoint colorPoint = _colorPoints[depthIndex];

                int colorX = (int)Math.Floor(colorPoint.X + 0.5);
                int colorY = (int)Math.Floor(colorPoint.Y + 0.5);

                if ((colorX >= 0) && (colorX < colorWidth) && (colorY >=
0) && (colorY < colorHeight))
                {
                    int colorIndex = ((colorY * colorWidth) + colorX) *
BYTES_PER_PIXEL;
                    int displayIndex = depthIndex * BYTES_PER_PIXEL;

                    _displayPixels[displayIndex + 0] =
_colorData[colorIndex];
                    _displayPixels[displayIndex + 1] =
_colorData[colorIndex + 1];

```

```
        _colorData[colorIndex + 2];        _displayPixels[displayIndex + 2] =
        _displayPixels[displayIndex + 3] = 0xff;
    }
    }
}

_bitmap.Lock();

Marshal.Copy(_displayPixels, 0, _bitmap.BackBuffer,
_displayPixels.Length);
_bitmap.AddDirtyRect(new Int32Rect(0, 0, depthWidth, depthHeight));

_bitmap.Unlock();
}

return _bitmap;
}
#endregion
}
}
```