



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

JAYANDERSON SOARES BONFIM

DIRETRIZES DE DESENVOLVIMENTO DE INTERFACE HOMEM- COMPUTADOR PARA O FISIOKINECT

Palmas – TO
2017

JAYANDERSON SOARES BONFIM

**DIRETRIZES DE DESENVOLVIMENTO DE INTERFACE HOMEM-
COMPUTADOR PARA O FISIOKINECT**

Trabalho de Conclusão de Curso (TCC) II 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. Me. Fabiano Fagundes.

JAYANDERSON SOARES BONFIM
**DIRETRIZES DE DESENVOLVIMENTO DE INTERFACE HOMEM-
COMPUTADOR PARA O FISIOKINECT**

Trabalho de Conclusão de Curso (TCC) II 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. Me. Fabiano Fagundes

Aprovado em: ____/____/____

BANCA EXAMINADORA

Prof. Me. Fabiano Fagundes

Orientador

Centro Universitário Luterano de Palmas – CEULP

Prof. Me. Fernando Luiz de Oliveira

Centro Universitário Luterano de Palmas – CEULP

Prof. Me. Pierre Soares Brandão

Centro Universitário Luterano de Palmas – CEULP

Palmas – TO
2017

Dedico este trabalho a todos que me apoiaram
e auxiliaram nas minhas decisões, em especial
minha família e amigos.

AGRADECIMENTOS

Agradeço primeiramente aos meus familiares que sempre estiveram presentes em minha vida, auxiliando em toda a minha trajetória acadêmica e ajudando a alcançar meus objetivos profissionais e pessoais. Com carinho, respeito e bons exemplos eles me ensinaram valores que formaram o ser humano que sou hoje.

De maneira geral, agradeço aos professores do curso de Sistemas de Informação do CEULP/ULBRA, pelo apoio desde o início da faculdade, por acreditarem na capacidade de seus alunos e por nos incentivarem a alcançarmos nossas metas acadêmicas.

Ao meu orientador, Fabiano Fagundes, pelo apoio, paciência, pelos ensinamentos, correções, cobranças e incentivos que possibilitaram o sucesso na conclusão deste trabalho.

Muito obrigado!

*“A tarefa não é tanto ver aquilo que ninguém viu,
mas pensar o que ninguém ainda pensou sobre
aquilo que todo mundo vê.”*

Arthur Schopenhauer

RESUMO

BONFIM, Jayanderson Soares. **Diretrizes de desenvolvimento de Interface Homem-Computador para o FISIOKINECT**. 2017. 56 f. Trabalho de Conclusão de Curso (Graduação) – Sistemas de Informação, Centro Universitário Luterano de Palmas, Palmas/TO, 2017.

O presente trabalho discute o projeto de criação de diretrizes de desenvolvimento de *interface* para a plataforma Fisiokinect, tendo por objetivo principal a padronização e facilitação do trabalho colaborativo de produção e desenvolvimento de *softwares* para a plataforma. Para chegar à forma mais adequada de *interface*, foi realizado um estudo documental e bibliográfico sobre o tema. Por fim, o produto final do trabalho é a criação de uma *guideline* para a plataforma onde será proposta um conjunto de boas práticas para o desenvolvimento da *interface* do sistema. Além de ser um projeto colaborativo e complexo, atualmente não existe um guia de *interface* para os novos desenvolvedores da ferramenta. por isso a importância deste trabalho. O Fisiokinect possui uma *interface* consistente, mas alguns pontos necessitam ser melhorados, como alinhamento, tipos de mensagens, o *feedback* com o usuário etc. No final deste estudo serão apresentadas as diretrizes que ajudarão aos novos colaboradores do sistema.

Palavras-chave: *design* de interface; Fisiokinect; *guideline*; *interface* homem-computador, *Kinect*.

LISTA DE FIGURAS

Figura 1 - Evolução da Interação homem-máquina.....	14
Figura 2 - <i>Command Line Interface</i>	15
Figura 3 - Exemplo de tecnologia <i>touchscreen</i>	17
Figura 4 - Abrangência da Usabilidade	22
Figura 5 - <i>Kinect Hardware</i>	26
Figura 6 - Resumo da arquitetura básica do <i>Kinect</i>	27
Figura 7 - Sensor <i>Kinect</i>	27
Figura 8 - Tela inicial FisioKinect.....	28
Figura 9 - Tela do módulo ADM.....	29
Figura 10 - Tela do módulo Antropometria	30
Figura 11 - Desenho de Estudo	31
Figura 12 - Exemplo de <i>layout</i> desalinhado	38
Figura 13 - Exemplo de <i>layout</i> alinhado	39
Figura 14 - Exemplo de <i>layout</i>	40
Figura 15 - Exemplo de mensagem de aviso	45
Figura 16 - Exemplo de mensagem de confirmação	46
Figura 17 - Cores primárias visão normal	48
Figura 18 - Cores primárias visão Protan	48
Figura 19 - Cores primárias visão Deuteranopia	49
Figura 20 - Cores primárias visão Tritanopia.....	49

LISTA DE TABELAS

Tabela 1 - Atributos de um bom e de um mau <i>software</i>	34
Tabela 2 - Controles	35
Tabela 3 - Comandos de Menus	41
Tabela 4 - Padrões de Escrita	42
Tabela 5 - Pontos Mensagens de Erro	44
Tabela 6 - Tipos de Deficiência.....	47

LISTA DE ABREVIATURAS E SIGLAS

AMD - Avaliação de Amplitude de Movimento em Plano Coronal Anterior

IHC - Interação Homem-Computador

CLI - *Command Line Interface*

GUI- *Graphical User Interface*

NUI - *Natural User Interface*

CRT - Tubos de Raios Catódicos

WIMP - *Windows, Icon, Menu e Pointer*

RBG - *Color Codes Chart*

LTS - Laboratório de Tecnologia em Saúde

LABMÍDIA - Laboratório de Multimídia

SUMÁRIO

1. INTRODUÇÃO	12
2. REFERENCIAL TEÓRICO	14
2.1. <i>INTERFACE</i> HOMEM-COMPUTADOR.....	14
2.2. <i>COMMAND LINE INTERFACE</i> (CLI)	15
2.3. <i>GRAPHICAL USER INTERFACE</i> (GUI)	16
2.4. <i>NATURAL USER INTERFACE</i> (NUI)	17
2.5. USABILIDADE.....	19
2.6. ACESSIBILIDADE.....	23
2.7. <i>GUIDELINES</i> PARA O DESENVOLVIMENTO DE <i>INTERFACES</i>	24
2.8. <i>KINECT</i>	25
2.9. <i>FISIOKINECT</i>	28
3. MATERIAIS E MÉTODOS	31
3.1. DESENHO DE ESTUDO.....	31
3.2. MATERIAIS	32
3.2.1. Fontes bibliográficas	32
3.2.2. <i>Hardware</i>	32
3.2.3. <i>Software</i>	32
3.2.4. Local de desenvolvimento	32
4. RESULTADOS E DISCUSSÃO	34
4.1. A GUIDELINE	34
4.1.1. Princípios básicos da aplicação	34
4.1.2. Controles	35
4.1.3. Disposição visual.....	37
4.1.4. Comandos	40
4.1.5. Estilo de escrita	42
4.1.6. Mensagens	44
4.1.7. Ícones	46
4.1.8. Interação.....	46
4.1.9. Cor	47
4.1.10 <i>Interface Natural</i>	49
5. CONSIDERAÇÕES FINAIS	52
REFERÊNCIAS	53

1. INTRODUÇÃO

As *interfaces* gráficas permitem a interação do usuário com a máquina por meio de elementos gráficos e dispositivos de entrada de dados (JORGE, 2016). Segundo o dicionário Aurélio, *interface* significa: "dispositivo (material e lógico) graças ao qual se efetuam trocas de informações entre dois sistemas. Limite comum a dois sistemas ou duas unidades que permite troca de informações." Com efeito, o dicionário Dicio ensina que *Interface* é: "o modo através do qual o usuário consegue, usando um computador, interagir com um programa ou com um sistema operacional. Aquilo que causa uma união física ou lógica entre dois sistemas que, diretamente, não poderiam estar conectados".

Segundo Preece (2005), o significado de *interface* está agregado a uma linguagem de entrada de dados para o usuário, uma saída de dados para a máquina e um protocolo de interação. E mais que um projeto de *interface*, a interação homem-computador se preocupa com as interações entre usuários e computadores.

Desde o primeiro computador até os dias atuais as interfaces por onde ocorre a comunicação entre o homem e a máquina estão em constante evolução. Como exemplo, no início essa interação entre homem-máquina era feita por meio de cartões perfurados, depois passou a ser por linha de comando, seguido por interface gráfica e, atualmente, já está sendo utilizada a interface natural do usuário, onde o homem comunica-se com a máquina de forma natural utilizando seus órgãos sensoriais e motores.

O *Kinect* é uma ferramenta que utiliza os órgãos sensoriais e motores do homem para se comunicar com a máquina e foi introduzido pela *Microsoft* no mercado em novembro de 2010 como um acessório para o console Xbox 360.

Esta inovação no mercado de jogos permitiu que o jogador interagisse com o *game* utilizando apenas os movimentos naturais do corpo, sem a necessidade de qualquer tipo de controle. Consiste em um acessório de reconhecimento de usuário, permitindo, dentre outros, mapear o ambiente em três dimensões, filmar e receber comandos de áudio com cancelamento de ruído e detecção da fonte através de seus microfones (FLORES; SOUZA, 2013). Segundo Camargo (2015), o *microsoft kinect* inicialmente possuía o codinome Projeto Natal, referenciando a capital do Rio Grande do Norte, e tem como seu idealizador o brasileiro Alex Kipman.

Por sua vez, o *FisioKinect* é uma plataforma computacional, ou seja, ele contém *softwares* destinados à área da saúde, desenvolvidos pelos alunos dos cursos de Ciência da Computação e Sistemas de Informação do Centro Universitário Luterano de Palmas (CEULP) em conjunto com o Laboratório de Tecnologia em Saúde (LTS) do CEULP ULBRA. A

plataforma apresenta um ambiente pré-existente e projetado para desenvolvimento de soluções para o campo de fisioterapia e conta com uma arquitetura de *hardware* baseada no *kinect* utilizando a *interface* gráfica e a *interface* natural para interagir com o usuário. Como exemplo dessas soluções, pode-se apresentar a Avaliação de Amplitude de Movimento em Plano Coronal Anterior (ADM) utilizando o sensor *kinect*, que se trata de uma ferramenta que utiliza o sensor *microsoft kinect for windows* na captura de ângulos corporais da ADM (CAMARGO, 2015).

Atualmente, a plataforma FísioKinect não possui um guia de *interface* a ser seguido, e, por ser um *software* colaborativo onde diferentes pessoas desenvolvem módulos para o sistema, ele acaba por sofrer diversas atualizações. Isso torna-o complexo, demonstrando a necessidade de um guia que contenha as diretrizes para o desenvolvimento da *interface* de forma que possa ser mantido um padrão capaz de atender às necessidades dos possíveis usuários.

Assim, o presente trabalho pretende realizar um estudo bibliográfico acerca dos dispositivos de *interface* natural de usuário e sobre a plataforma FísioKinect, de forma a analisar a *interface* da plataforma, propor diretrizes de desenvolvimento de *interface* e, por fim, propor a *guideline*.

Desta forma, a presente obra se justifica pelo fato de não existir um guia de *interface* para os novos desenvolvedores da ferramenta. E, sendo um projeto colaborativo e complexo, faz-se especialmente importante dispor de um conjunto de diretrizes para a plataforma FísioKinect, de forma que possibilite aos novos programadores do sistema ter uma *guideline* consistente para direcioná-los no desenvolvimento da *interface* da plataforma FísioKinect.

2. REFERENCIAL TEÓRICO

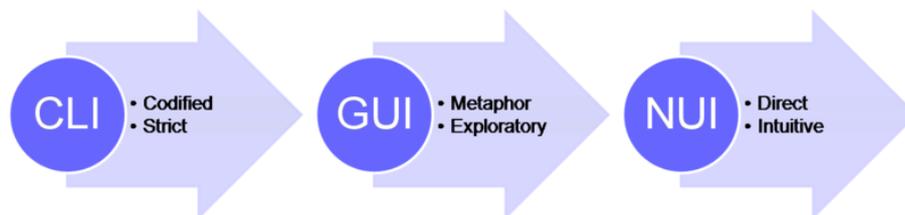
Neste tópico serão explanados os conceitos mais importantes sobre o assunto abordado, como: *Interface* Homem-Computador desde a sua história, conceito e características de Usabilidade e Acessibilidade, *Guidelines* para *interface* e sobre a plataforma FizioKinect.

2.1. INTERFACE HOMEM-COMPUTADOR

A Interação Homem-Computador (IHC) destaca-se pela seguinte evolução: *Batch interface*, *Command Line Interface (CLI)*, *Graphical User Interface (GUI)* e a *Natural User Interface (NUI)*. Segundo Garbin (2010), até cerca de 1960 os primeiros *mainframes* operavam somente em modo de processamento de dados por lotes e não possuíam *interface* interativa, trabalhando através de cartões perfurados e fitas magnéticas. Esse tipo de processo realizava a maioria dos cálculos necessários, mas o usuário não possuía uma comunicação interativa com o computador.

A seguir, a figura 1 representa a evolução da interação homem-máquina e posteriormente o tópico seguinte explora o CLI.

Figura 1 - Evolução da Interação homem-máquina



Fonte: Wigdor e Wixon (2011)

Pode-se observar que a evolução da IHC passou por três grandes fases: a primeira, CLI, que é a interação por linha de comando; posteriormente, a GUI, que consiste em uma interação gráfica com o usuário; e, por fim, está ocorrendo a transição e mesclagem entre a GUI e a NUI que é um tipo de interação natural que tende a utilizar movimentos naturais como forma de entradas, para que ocorra a interação entre o homem e a máquina.

2.2. *COMMAND LINE INTERFACE (CLI)*

Na interação *Command Line Interface (CLI)* ou interface por linha de comando, os usuários precisavam aprender a utilizar um meio artificial de entrada, o teclado, e uma série de entradas codificadas, que tinha uma gama limitada de respostas, onde a sintaxe desses comandos era rigorosa (CARDOSO; SCHMIDT, 2012; WIGDOR; WIXON, 2011).

Em 1970, surgem *interfaces* que permitiam interatividade com o usuário, os *mainframes* já estavam aceitando comando e operavam como computadores em sistema de tempo compartilhado (*time-sharing*), esse conceito chamava-se *Command-Line Interface*. O CLI originou-se de máquinas de escrever eletromecânicas, utilizadas desde os anos 1950 para transmissão de dados. Essas máquinas de escrever eram ligadas aos computadores, somado ao CRT (tubos de raios catódicos) para a visualização do *prompt* de comando a ser digitado (STEPHENSON, 1999). Até hoje alguns programas ou configurações que exijam maior conhecimento técnico, ainda utilizam a linha de comando. A figura 2 apresenta um exemplo do CLI.

Figura 2 - *Command Line Interface*



```

IBM Tivoli Storage Manager
Command Line Backup-Archive Client Interface
  Client Version 6, Release 4, Level 1.5
  Client date/time: 13-03-2014 14:45:26
(c) Copyright by IBM Corporation and other(s) 1990, 2013. All Rights Reserved.

Node Name: DEMO-WINDOWS8-ITSERU
Session established with server 0X_HFS_B1: AIX
  Server Version 6, Release 3, Level 4.2000
  Server date/time: 13-03-2014 14:45:26 Last access: 13-03-2014 14:14:57

tsm>

```

Fonte: Google Data¹

Observa-se na figura acima um exemplo atual da utilização da *interface* por linha de comando, visto que mesmo se baseando por linha de comando, é possível visualizar uma janela com ícones que também aceita interatividade, assunto da próxima seção. Assim, o CLI consiste em um computador que necessita de um mecanismo para que ocorra a interação do usuário e máquina através da digitação de comandos de texto para realizar tarefas (GARBIN 2010).

¹Disponível em: <<https://help.it.ox.ac.uk/hfs/help/dsmc>>. Acesso em 27 mar. 2017.

O pesquisador e inventor Steve Mann, durante os anos de 1970 e 1990, desenvolveu uma série de estratégias de *interface*-usuário usando interação natural com o mundo real, como alternativa a CLI ou a GUI (ANNA; FERRONATO, 2010).

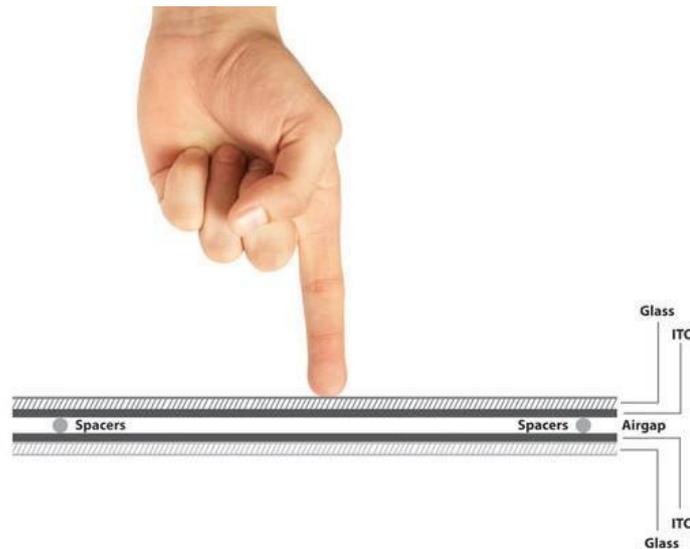
2.3. GRAPHICAL USER INTERFACE (GUI)

Medeiros Filho e Vieira (2013), afirmam que o verdadeiro início da interação homem-computador foi na CLI. Diferente desta, a *Graphical User Interface* (GUI) permite que o usuário interaja com o sistema, de forma a apontar, mover ou conectar representações de objetos na tela do computador. As *interfaces* gráficas mais comuns são as baseadas no contexto WIMP (*Window, Icon, Menu e Pointer*) que se caracterizam por utilizar o *mouse* ou outro dispositivo equivalente para manipular os elementos da tela que fazem com que a interação possa ser mais intuitiva.

Uma GUI Consiste na ação de mover objetos em uma tela e selecioná-los em menus. O usuário, portanto, modificaria um documento por meio de uma *interface* de maneira equivalente, ou parecido, a que faria na forma impressa. Esse tipo de interação é conhecida como *What You See Is What You Get* - WYSIWYG, em tradução livre, “o que você vê é o que você obtém”. Tal modelo de *interface* era mais voltado aos editores de texto, porém se tornou mais abrangente sendo adequada a outras necessidades de *interfaces*.

Ainda entre os anos de 1970 e 1980, foram registradas patentes de outro modelo de interatividade, como as voltadas às telas sensíveis ao toque (*Touchscreens*), onde são utilizados os dedos ou caneta específica como cursores, ao invés do *mouse*, principalmente em dispositivos portáteis (GARBIN, 2010). Esse tipo de *interface* permite ao usuário interagir com controles e aplicações de forma intuitiva, regra em *smartphones* e *tablets*. A figura 3 a seguir apresenta um exemplo da tecnologia *touchscreen*.

Figura 3 - Exemplo de tecnologia *touchscreen*



Fonte: Tech Explainer²

A tecnologia *touchscreen* trabalha com a GUI e a NUI, utilizando-se os dedos como se fosse o *mouse* em uma *interface* próxima à realidade. Com isso, ela não perde as características de uma GUI, porém possui comandos realizados com movimentos naturais em substituição ao *mouse*. Na próxima seção será explanado sobre as *interfaces* naturais.

2.4. NATURAL USER INTERFACE (NUI)

A *Natural User Interface* (NUI) tende a ser bastante intuitiva e de fácil aprendizado por se basear em movimentos naturais do usuário. Os dispositivos que operam com a concepção de *interface* natural do usuário são sistemas de interação homem-máquina em que o usuário opera por meio de ações intuitivas relacionadas ao comportamento humano natural e do cotidiano (CARDOSO; SCHMIDT, 2012).

A tendência percebida no mercado é de que métodos de interação através de menus e caixa de diálogo com o uso de teclado e *mouses* sejam substituídos em partes por técnicas de reconhecimento gestual e, escrita manual e reconhecimento de fala. Em relação a comandos de voz, compreender o que o usuário deseja, pode ser difícil, considerando-se as formas de falar, sotaque, idioma e a capacidade do *software* de absorver e compreender apenas o necessário.

Para uma boa experiência a *interface* natural do usuário necessita de *softwares* capazes de identificar as ações do usuário de maneira mais natural e interpretar o seu significado como

² Disponível em: <<https://techexplainer.wordpress.com/2012/04/02/resistive-vs-capacitive-touchscreen/>>. Acesso em: 01 abr. 2017.

dispositivo de entrada (GARBIN, 2010). Segundo LIU (2010), as *Interfaces Naturais* apresentam as seguintes características:

- **Centrado no Usuário:** esse tipo de *interface* se preocupa com as necessidades, desejos e limitações dos usuários. O processo de criação deste tipo de *interface* requer a análise e previsão de como os usuários a utilizarão, bem como testes para validação de premissas com relação ao comportamento do usuário.
- **Multi-Channel:** utiliza habilidades sensoriais e motoras em conjunto para aumentar a naturalidade da interação entre homem-computador. As habilidades sensoriais humanas trabalhadas são visão, audição, tato, olfato e equilíbrio, quanto que para as habilidades motoras são as mãos, boca, olhos, cabeça, pés, corpo e etc.
- **Inexata:** uma *interface* exata deve explicar completamente qual a intenção do usuário como, por exemplo, ao usar um teclado e mouse, cujas entradas não se modificam e são sempre as mesmas, não cabe outra interpretação. As ações e pensamentos das pessoas não são muito precisos, ou seja, são de forma inexata, logo, a *interface* natural deve entender a solicitação do usuário e até corrigir alguns erros no processo de comunicação para compreender as verdadeiras intenções do usuário.
- **Grande largura de banda:** *interface* natural capta grande quantidade de informações quando comparada, ao teclado, por exemplo, onde a entrada de dados é muito baixa. Já com a interação natural o usuário pode utilizar voz, gestos e expressões faciais simultaneamente, elevando a quantidade de informações para se capturar.

Com efeito, existem dispositivos de reconhecimento de gestos que rastreiam movimentos de usuários traduzindo-os em instruções e, normalmente, são utilizados em sistemas de consoles de *games* como o *PlayStation Move* da *Sony* e o *Nintendo Wii*, que utilizam acelerômetros e giroscópios para detectar inclinação, rotação e aceleração (CARDOSO; SCHMIDT, 2012).

O tipo mais sofisticado de dispositivo de leitura de movimentos é equipado com uma câmera e *software* no dispositivo que reconhece gestos específicos e os traduz para ações como é o caso do *kinect*, sensor de movimento para o console de jogos Xbox 360, permitindo aos usuários interagir através de movimentos corporais, gestos e comandos de voz. De acordo com Trevisan et al ([s.d.], p. 73) “é possível interagir em ambientes holográficos com o uso de dispositivos para reconhecimento de gestos que empregam sensores de profundidade, entre eles *Microsoft Kinect*, *ASUS Xtion* e *Prime Senses Prime Sensor*”.

Em 2010, Daniel Wigdor e Dennis Wixon forneceram as bases teóricas para uma operacionalização de construção de *interfaces* de usuário naturais em seu livro, onde

cuidadosamente diferenciaram entre *interfaces* naturais de usuário, as tecnologias utilizadas para alcançá-las e *interface* de usuário baseada na realidade. Os autores ainda alertam para os erros que os novos desenvolvedores de *interfaces* naturais podem cometer. Entre esses erros está a possibilidade do *design* se basear em conceitos e abordagens que funcionariam em outro tipo de *interface*, em casos mais extremos, o desenvolvedor faria a nova aplicação (NUI) ser uma cópia da *interface* antiga, neste caso o NUI se tornaria uma GUI com entradas naturais, probabilidade de isso não funcionar muito bem é enorme.

2.5. USABILIDADE

De forma sucinta, usabilidade é a capacidade de algo ser utilizado pelo ser humano de forma fácil e eficaz (SHACKEL, 2009). Diversos princípios de *design* foram definidos com o intuito de guiar o planejamento do *design* de interação. Entre eles existem dois conceitos principais: a caracterização e a análise do usuário, e a identificação de onde o produto será utilizado. A utilização desses conceitos é de suma importância, pois permite que os sistemas sejam desenvolvidos de forma personalizada para um tipo específico de usuários, fazendo com que o usuário possa ter uma melhor experiência de interação, além de permitir que a tecnologia se ajuste às restrições presentes na sua vida (HONG, 2011).

Segundo Rocha e Baranauskas (2003), mesmo que possa parecer inviável considerar essa quantidade de usuários que possam utilizar o sistema durante a implementação de uma *interface*, quando essas diferenças são consideradas logo no início do desenvolvimento do projeto, a implementação torna-se simples e mais ágil, além de diminuir os custos de manutenção.

Além de identificar o usuário e o local onde a aplicação será utilizada, a usabilidade deve ser usada como uma métrica para definir o quanto um produto pode ser utilizado por vários usuários dentro dos objetivos da eficiência, eficácia e a satisfação. O desenvolvedor da *interface* deve ser guiado por essas métricas de usabilidade para que a interação se desenvolva de forma natural. De acordo com Rogers *et al.* (2007), a usabilidade é geralmente aceita como garantia de que os produtos interativos são fáceis e agradáveis de serem utilizados pelo usuário.

A norma ISO/IEC 9126-2011 contém diretrizes de usabilidade sobre requisitos de qualidade de sistemas e *softwares*. Por se tratar de norma geral para o desenvolvimento de *softwares* com qualidade, seus escopos possuem, além de recomendações de usabilidade, diretrizes para critérios: de adequação das funcionalidades, performance, compatibilidade, segurança e portabilidade. Essa norma ainda possui diretrizes para eficácia, eficiência,

satisfação, gerenciamento de riscos e cobertura do contexto, demonstrando que a preocupação com a experiência final do usuário com o produto final (MARTINS, 2016). São diretrizes de usabilidade e de qualidade de uso presentes na norma (ABNT, 2011):

Diretrizes de usabilidade

1. Reconhecimento: permitir que os usuários possam reconhecer se um sistema é apropriado para suas necessidades.
2. Aprendizagem: permitir que um sistema possa ser usado por usuários específicos para alcançar objetivos específicos de aprendizagem em um determinado contexto de uso.
3. Operacionalidade: permitir que um sistema tenha atributos que o tornam fácil de operar e controlar.
4. Erros: permitir que um sistema proteja os usuários contra a ocorrência de erros.
5. Estética da *Interface*: permitir que uma *interface* de um sistema possua uma interação agradável e satisfatória para o usuário.
6. Acessibilidade: permitir que um sistema possa ser usado por usuários com a maior variedade de características e capacidades possíveis para alcançar um objetivo especificado em um determinado contexto de uso.

Diretrizes de Qualidade de uso

1. Eficácia: permitir aos usuários atingir metas específicas com exatidão e integridade.
2. Eficiência: permitir aos usuários atingir metas específicas utilizando o mínimo de recursos possível.
3. Satisfação: permitir ao sistema atender as necessidades dos usuários de forma satisfatória quando utilizado em um determinado contexto de uso.
4. Inexistência de Riscos: permitir que um sistema reduza o risco potencial de estado econômico, a vida humana, a saúde ou o meio ambiente.
5. Cobertura do Contexto: permitir que um sistema possa ser utilizado em contextos específicos que vão além dos explicitados nestas diretrizes.

Existem diversos princípios e diretrizes que orientam *designers* o caminho que devem seguir a fim de desenvolver uma *interface* usual sem, entretanto, desconsiderar que a avaliação de uma *interface* não deve ocorrer apenas durante sua implementação, mas continuamente.

Assim, no desenvolvimento de um projeto, não se deve considerar apenas os princípios de usabilidade, sendo necessário que o desenvolvedor realize avaliações das interações do usuário real com o sistema. Para se realizar essas avaliações existem testes, métricas e heurísticas de usabilidade (ROCHA; BARANAUSKAS, 2003; NIELSEN, 1993).

De acordo com Nielsen (1994), um conjunto pequeno de especialistas em *design* avalia separadamente uma *interface* confrontando-a com regras, conhecidas também como “heurísticas”, que servem para identificar erros na *interface* que podem atrapalhar a usabilidade. Olibário (2013) diz que, o método de avaliação de usabilidade de *interface* proposto por Nielsen em 1994 é um dos mais conhecidos entre os especialistas e é composto por um conjunto de heurísticas:

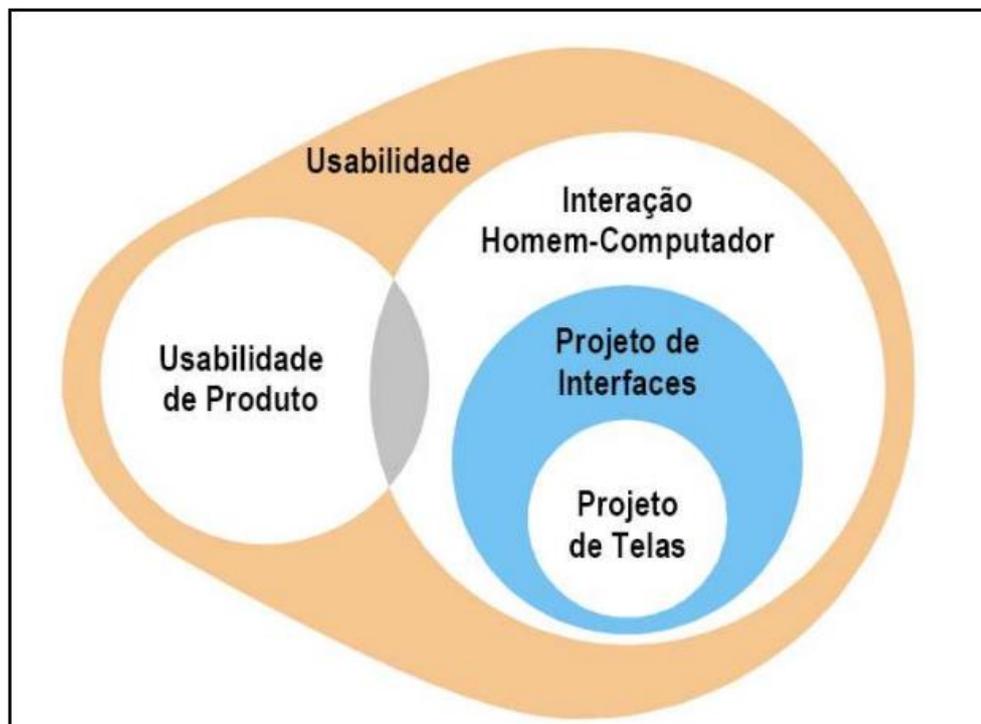
- Visibilidade do status do sistema: o usuário deve estar completamente informado do que está acontecendo, por meio de *feedback* imediato da *interface*;
- Compatibilidade do sistema com o mundo real: a terminologia deve ser adequada à linguagem do usuário e não orientado ao sistema. As informações devem ser organizadas de acordo com o modelo mental do usuário;
- Controle e liberdade do usuário: disponibilizar “saídas de emergência” ao usuário, para que ele possa desfazer ou refazer ações, a fim de que ele se situe em um ponto recente da interação;
- Consistência e padrões: nunca identificar uma mesma ação por ícones ou metáforas diferentes. Elementos similares devem ser usados para propósitos semelhantes, assim como funcionalidades semelhantes devem possuir uma sequência de ações semelhantes;
- Prevenção de erros: idealmente, *interfaces* não precisam de mensagens de erro por serem capazes de prevenir que erros ocorram. Ações definitivas podem ter um tratamento anterior para que os usuários as confirmem por meio de *checkboxes*, por exemplo;
- Reconhecimento ao invés de lembrança: o usuário não deve precisar memorizar o que está realizando. Permitir que a *interface* atue como um meio de dialogar com o usuário, em tempo de execução;
- Flexibilidade e eficiência de uso: o sistema deve ser fácil de ser operado por usuários novatos, mas também robusto o suficiente para permitir a eficiência de uso a usuários avançados. Teclas de atalhos e de comando por voz podem ser alternativas para tornarem a

interface flexível;

- Estética e design minimalista: as informações devem ser sucintas e não devem informar mais do que os usuários necessitam para realizar a funcionalidade corretamente. Os diálogos do sistema precisam ser diretos e naturais e devem aparecer nos momentos adequados;
- Ajuda para o usuário identificar, diagnosticar e corrigir erros: as mensagens de erro devem ser claras e simples e não podem intimidar o usuário. Ao contrário, devem estimulá-lo ao oferecer formas de corrigir o erro;
- Ajudas e documentação: um bom *design* evita que um usuário tenha que usar opções de ajuda com frequência. Entretanto, é fundamental que o sistema possua telas específicas de ajuda, para orientar o usuário em caso de dúvidas.

Segundo Gonçalves (2008), a usabilidade pode ser trabalhada em todos os tipos de produtos, como embalagens, roupas, móveis, meios de transporte, manuais, bulas de remédios e sistemas computacionais, concluindo-se que a usabilidade pode ser trabalhada onde houver *interface* com o homem. A figura 4 mostra a abrangência da usabilidade.

Figura 4 - Abrangência da Usabilidade



Fonte: Gonçalves (2008)

Observa-se na figura que o projeto de *interfaces* pertence à IHC, que por sua vez apresenta uma intersecção com usabilidade do produto e que todas essas áreas fazem parte da

usabilidade como um todo. Não se pode esquecer que a usabilidade de uma *interface* é um dos fatores determinantes para o sucesso de uma aplicação, considerando-se, ainda, a acessibilidade.

2.6. ACESSIBILIDADE

Atualmente a vida cotidiana da sociedade está caracterizada pela informação, onde o avanço das tecnologias difunde-se rapidamente. Ocorre que pessoas com deficiência, analfabetos e idosos tendem a sofrer a infoexclusão devido às dificuldades de acesso e problemas de acessibilidade nos ambientes atuais.

Acessibilidade significa estreitar limitações ou anular qualquer tipo de obstáculo para qualquer pessoa, mesmo as não deficientes ou que possuem limitações temporárias (QUEIROZ, 2015). Para Nicácio (2014), a acessibilidade é a capacidade de qualquer pessoa, independentemente de sua necessidade, ter facilidade em adentrar, aproximar, utilizar, entre outras tarefas.

A Lei 13.146, de 06 de julho de 2015, define acessibilidade como a possibilidade e condição de alcance para utilização, com segurança e autonomia, de espaços, mobiliários, equipamentos urbanos, edificações, transportes, informação e comunicação, inclusive seus sistemas e tecnologias, onde se encaixa a *interface*, bem como de outros serviços e instalações abertos ao público, de uso público ou privados de uso coletivo, tanto na zona urbana quanto na rural, por pessoa com deficiência ou com mobilidade reduzida.

Segundo Andréa Sonza (2008) os sites necessitam ser adaptados para que todos os usuários possam ter acesso, inclusive os que possuem algum tipo de deficiência ou limitações. Não obstante existirem recursos disponíveis para o desenvolvimento de sites acessíveis, a grande maioria permanece inacessível, fazendo com que os níveis de acessibilidade sejam ainda muito limitados.

O W3C (*World Wide Web Consortium*) procurando assegurar a utilização de padrões na web divulgou, em maio de 1999, a Cartilha de Diretrizes para Acessibilidade do Conteúdo Web 1.0 (*Web Content Accessibility Guidelines – WCAG 1.0*), que hoje está em sua versão 2.0 (W3C, 2012), sendo, até os dias atuais, a principal referência em torno de acessibilidade na web no mundo (DIAS, 2014).

A cartilha está dividida em quatro princípios que orientam e permitem o acesso, de qualquer indivíduo ao conteúdo da web (W3C, 2012). O primeiro, Princípio da percepção:

todos os dados e os conteúdos devem ser perceptíveis ao usuário; Princípio da operação: as *interfaces* devem ser operáveis por qualquer usuário; Princípio da compreensão: Conteúdos e controles devem ser compreensíveis pelo usuário; Princípio da robustez: o conteúdo deve ser forte o suficiente para trabalhar com tecnologias atuais e futuras.

O uso de inovações tecnológicas abre novas perspectivas de concepção e integração social, assegurando o acesso à informação de modo a não excluir as pessoas com necessidades especiais ou com percepções limitadas. *Interfaces* acessíveis e usáveis suprem limitações e garantem a inclusão.

2.7. GUIDELINES PARA O DESENVOLVIMENTO DE *INTERFACES*

Segundo Gasparetto, Pedrozo e Oliveira (2016), *design de interface* é um processo lógico e criativo de projeto que se baseia em três pilares: o usuário, tecnologias e métodos utilizados para a realização das tarefas e o contexto. Já Rocha e Baranauskas (2003, p. 17), acreditam que "os objetivos da IHC são os de produzir sistemas usáveis, seguros e funcionais". Observa-se que os autores enfatizam o fato de terem que ser baseados no usuário final, ser funcional e usável, pois não adianta ter um sistema perfeito em se tratando de código se a sua *interface* não for usual e de fácil aprendizado para o usuário que se deseja alcançar.

Nielsen (1993) ensina que *guidelines* ou diretrizes para desenvolvimento de IHC são recomendações utilizadas em avaliações heurísticas durante o desenvolvimento de uma IHC. Essa avaliação é feita por um grupo de profissionais observando e analisando a *interface* com o intuito de se identificar problemas e verificar a aplicação de heurísticas de usabilidade para solucioná-los (NIELSEN, 1993). A *guideline* pode e deve ser usada como apoio a projetos de uma IHC. É importante o *designer* fazer uso dessas diretrizes na construção de uma *interface* homem-computador, quando ele a tem disponível.

Shneiderman e Plaisant (2009) afirmam que essas diretrizes são como regras que representam uma linguagem comum para facilitar o trabalho em equipe, uniformizando as *interfaces* que vão sendo criadas. De acordo com Machado Neto (2013), acredita-se que as primeiras diretrizes foram propostas por Mullet e Sano em 1995. Posteriormente, em 2001, Lynch e Horton também criaram um conjunto de regras para *interface* que foi atualizado por Galitz quatro anos depois. Shneiderman (2009) diz que as *guidelines* são um dos quatro pilares que sustentam um projeto de IHC; os outros são testes de usabilidade, de ferramentas de projeto de IHC e de requisitos de IHC.

Os objetivos principais dessas diretrizes são aumentar a eficiência, reduzir a variação e poupar recursos financeiros, ajudando novos desenvolvedores menos experientes, fornecendo detalhes aos mais experientes e promovendo uma consistência ao projeto a ser implementado. Segundo Lima Junior (2016), a Apple faz com que terceiros que desenvolvem aplicações para seus dispositivos em sua *App Store*, sigam as diretrizes propostas pela empresa no guia *Human Interface Guidelines iOS*. Esse guia apresenta conceitos básicos de *UI Design*, estratégias de *design*, tecnologias do Sistema iOS, elementos de UI e etc, fazendo com que os novos *app* sejam submetidos a um processo rigoroso para aprovação.

O processo foi criado para que a empresa possa manter o *status* de qualidade alcançada com seus clientes. Ainda de acordo com Lima Junior (2016), estudos mostram que a preocupação com o desenvolvimento de *interfaces* possibilita que um projeto de *software* esteja mais próximo do sucesso.

O Google também possui uma *guideline* para desenvolvedores *android*, denominada *Android Design*, que apresenta princípios básicos e materiais para *design*, dispositivos, estilos, padrões, entre outros. No entanto, diferente da Apple, a empresa não impõe aos desenvolvedores de *softwares* para o *Play Store* que sigam à risca as diretrizes propostas por ela, pois acreditam que os desenvolvedores devam ter liberdade de criatividade para pensar e criar novos modelos. Ocorre que isso possibilita que vários aplicativos com padrão baixo estejam disponíveis no *Play Store*.

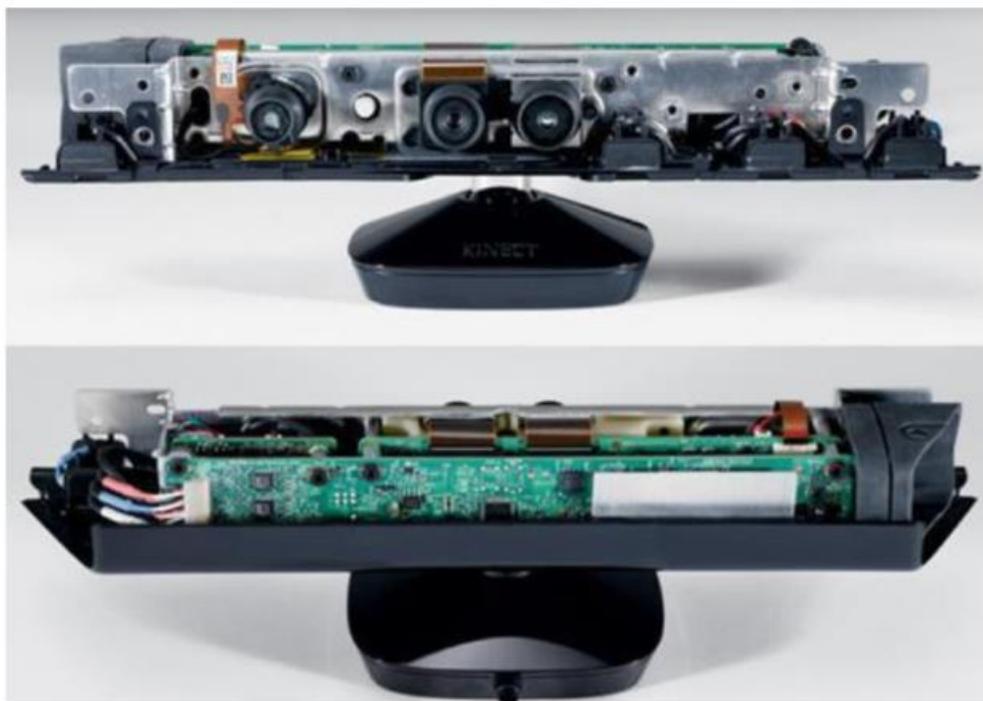
Essas diretrizes de desenvolvimento não servem para fazer com que o *design* não seja criativo, elas são criadas para ajuda-los a não cometer erros que venham comprometer seriamente todo o sistema. Por exemplo, na seção seguinte será explanado um pouco sobre o *kinect*, dispositivo que é utilizado no FisioKinect. Caso o desenvolvedor não saiba dos parâmetros mínimos de distância necessários para que o dispositivo possa "ver" o paciente com eficiência e/ou desenvolver uma *interface* muito pequena, provavelmente o fisioterapeuta não conseguirá avalia-lo e poderá condenar o sistema. Sabendo das distâncias mínimas, o próprio *kinect* calcula essa distância e o sistema pode apresentar mensagem de alerta até que o paciente se encontre na distância ideal.

2.8. KINECT

O *kinect* é um dispositivo criado inicialmente como um acessório a um console de game *Xbox* (figura 5) da gigante da informática americana *microsoft*. Ele possui um sistema de interação revolucionário à época de seu lançamento de interação com os jogos somente com movimentos corporais com um preço acessível (GUIMARÃES, 2015).

Foi desenvolvido pela empresa israelita *Prime Sense* e idealizado pelo brasileiro Alex Kipman. Este dispositivo foi projetado para ser capaz de permitir aos jogadores interagirem com os jogos sem a necessidade de ter um controle/*joystick* em mãos (CAMARGO, 2015). O *kinect* foi criado para ser um acessório do console de videogame *Xbox* da *microsoft*, todavia após alguns anos a empresa fez a liberação da *interface* de programação do mesmo. API (*Application Programming Interface*) permitiu que qualquer pessoa com acesso à linguagem de programação pudesse criar *softwares* para ser utilizado com o *kinect* e este fosse utilizado junto de um computador (JUNIOR et al., 2015). A figura 5 abaixo representa o *kinect*.

Figura 5 - *Kinect Hardware*



Fonte: Junior et al (2015, p. 5)

Devido ao fato de ter uma visão computacional que possui coordenadas com um marcador de erro menor que um centímetro na detecção de objetos, ele se tornou apto para a realização de inúmeras tarefas. A partir disso surgiu uma nova e ampla gama de utilização para o sensor de movimento, desde a exploração de petróleo, treinamento militares e tratamento médicos (JUNIOR et al., 2015). Em sua arquitetura o *kinect* conta com sensores de cor e profundidade, microfones, sensor infravermelho – utilizados para localização do jogador e para utilização de comandos de voz câmera VGA (*Vídeo Graphics Array*) colorida. O *kinect* capta as seguintes informações, a figura 6 a seguir mostra o resumo da arquitetura básica do *kinect* e de como ele capta as informações.

Figura 6 - Resumo da arquitetura básica do *Kinect*

Fonte: Flores; Souza (2013, p. 04)

A câmera RGB admite o reconhecimento corporal das pessoas que se encontram à frente do console, o sensor de profundidade permite o mapeamento em três dimensões do ambiente à frente, seus quatro microfones são capazes de capturar e identificar vozes de diferentes pessoas no mesmo ambiente e ainda identifica em qual direção estão vindo os sons e cancela ruídos e ecos que venham atrapalhar na identificação das vozes. Além disso, o *kinect* ainda possui um motor na base do aparelho que movimenta a cabeça para cima ou para baixo conforme a necessidade (FLORES; SOUZA, 2013). A figura 7 apresenta o sensor *kinect*.

Figura 7 - Sensor *Kinect*

Fonte: Camargo (2015, p. 28)

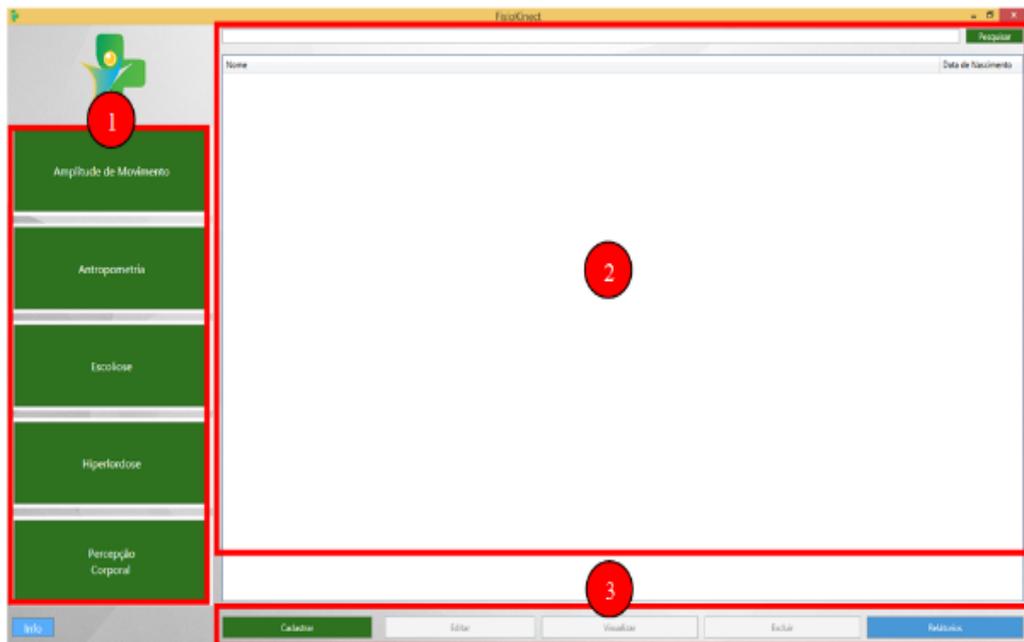
Nota-se que o *kinect* faz parte da revolução silenciosa de dispositivos de leitura de *interface* natural que neste momento culmina com o conceito de realidade aumentada (COLPANI; HOMEM, 2016).

A próxima seção discorre sobre o sistema a ser estudado neste trabalho, onde vê-se a aplicação do *kinect* no *software* em estudo.

2.9. FISIOKINECT

O FizioKinect é uma plataforma que utiliza o *kinect* como apoio em ferramentas de avaliação e controle de ambiente virtual focado em sessões de fisioterapia. O uso do *kinect* pode trazer diversos benefícios, entre eles: ganho de concentração, equilíbrio, força muscular e avaliações ao usuário. No mais, seu uso faz com que as atividades se tornem mais agradáveis de modo que envolva o paciente, o qual poderá realizar os movimentos com mais espontaneidade. E a ferramenta sendo utilizada por um profissional qualificado da área de fisioterapia, pode ser um importante auxiliar no diagnóstico e tratamento fisioterápico. A tela inicial do FizioKinect foi desenvolvida em conjunto com o grupo de pesquisa Tecnologia, Saúde e Qualidade de Vida. A figura 8 apresenta a tela inicial que foi criada para o sistema.

Figura 8 - Tela inicial FizioKinect



Fonte: Camargo (2015, p. 42)

Esta plataforma apresenta em (1) os módulos desenvolvidos pelo grupo de pesquisa, eles estão separados por “botões” com as funcionalidades pertinentes aos seguintes trabalhos:

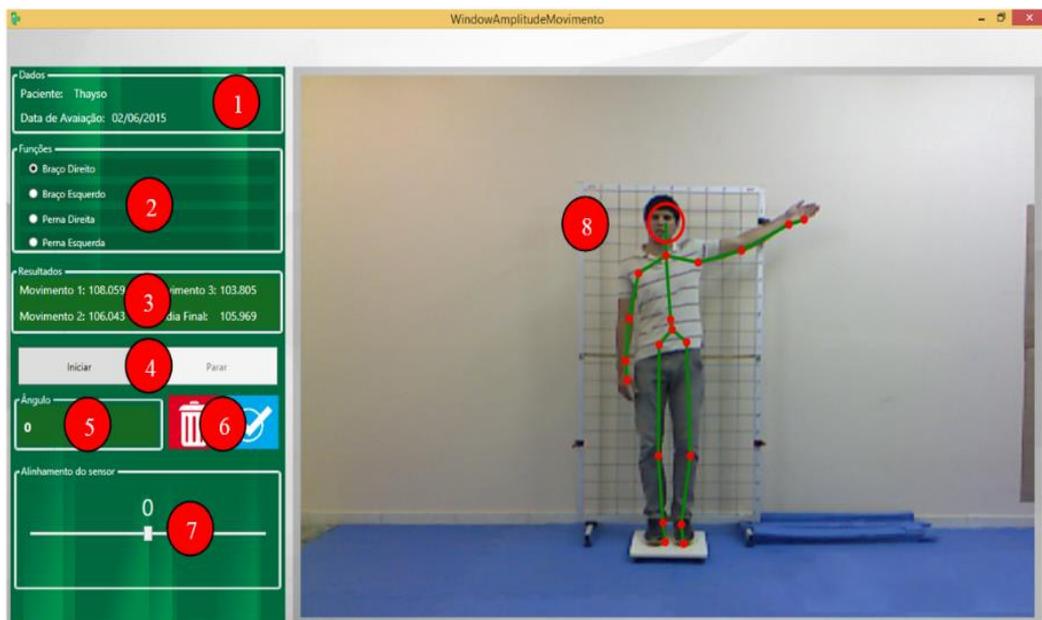
- Amplitude de Movimento – Avaliação da amplitude de movimento (ADM) em plano coronal anterior utilizando o *kinect*: articulações do ombro e do quadril.
- Antropometria – Utilizando o *microsoft kinect* na obtenção de atributos antropométricos.
- Hiperlordose – Análise postural computadorizada para identificação de

hiperlordose utilizando o *kinect*.

- Escoliose – Avaliação postural computadorizada utilizando o *kinect*.
- Percepção Corporal – Automatização do *Image Marking Procedure* para análise do esquema corporal utilizando o *kinect*.

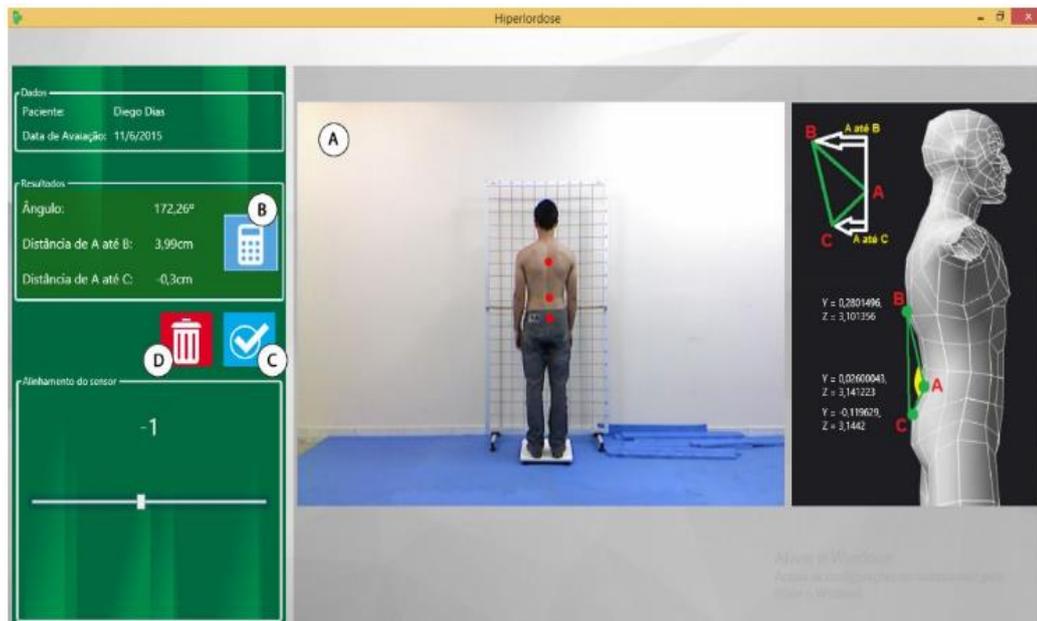
Em (2) a *interface* apresenta um campo de busca para pesquisa de pacientes já cadastrados e logo a baixo apresenta uma lista de todos os pacientes cadastrados. Em (3) se encontram os “botões” para cadastro, edição, visualização e exclusão de pacientes e por último o “botão” para geração de relatórios. A figura 9 apresenta a tela do módulo Amplitude de Movimento (ADM) e a figura 10 apresenta a tela do módulo Hiperlordose, respectivamente.

Figura 9 - Tela do módulo ADM



Fonte: Camargo (2015, p. 54)

Figura 10 - Tela do módulo Antropometria



Fonte: Tainha (2015, p. 42)

A *interface* da plataforma permite que as aplicações integradas sejam utilizadas de forma fácil, mas ela necessita de alguns ajustes. Os botões e *layout* da ferramenta são simples e apresentam um design não atrativo, e considerando o fator ético, talvez não seja correto os pacientes cadastrados serem listados logo na tela inicial.

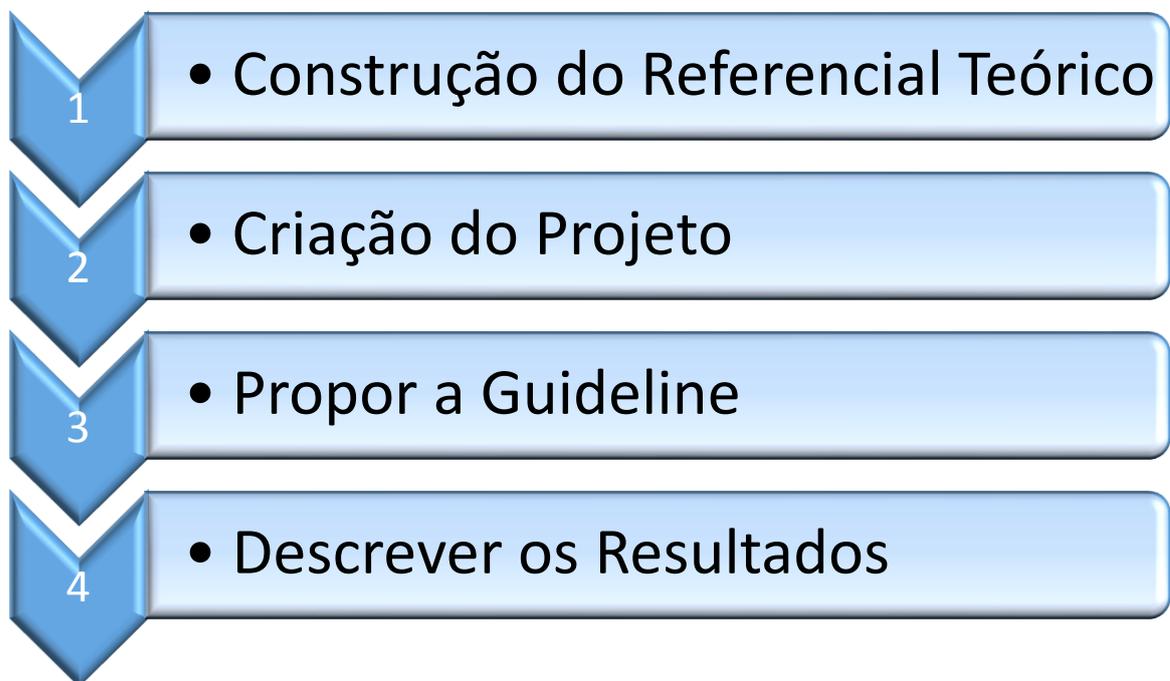
3. MATERIAIS E MÉTODOS

3.1. DESENHO DE ESTUDO

Neste, foi proposta uma pesquisa aplicada em laboratório, com o objetivo metodológico exploratório e procedimentos bibliográfico e documental.

Para conseguir propor as diretrizes de *interface*, foram necessárias 4 etapas, a figura 11, a seguir, demonstra a sequência destas etapas.

Figura 11 - Desenho de Estudo



A figura a cima mostra a sequência de etapas utilizadas para o desenvolvimento deste trabalho. Esse conjunto de etapas foram de suma importância para a realização deste trabalho. Cada etapa será explicada a baixo para um melhor entendimento:

- Construção do Referencial Teórico - Etapa primordial sobre o objeto em estudo e temas relacionados, buscando em periódicos científicos, dissertações de mestrado, teses de doutorado e outros, o conhecimento mínimo necessário para o desenvolvimento do presente trabalho.
- Criação do Projeto - Após o estudo dos temas, foi possível criar o projeto, onde foram definidas a hipótese problemática, a justificativa e o objetivo geral.
- Propor a *guideline* - Depois de possuir o conhecimento mínimo necessário, ter

definido o projeto e feito uma análise das *interfaces* atuais do sistema, foi proposta a *guideline*.

- Descrever os Resultados - Nesta etapa foram descritos os resultados alcançados no desenvolvimento do presente trabalho.

3.2. MATERIAIS

Nesta seção serão apresentados os materiais empregados para a efetivação do presente trabalho.

3.2.1. Fontes bibliográficas

Foram consultados artigos de periódicos científicos, dissertações de mestrado e teses de doutorado, livros e outros materiais, sobre os conceitos envolvidos no trabalho e que apresentavam conhecimento sobre o tema, e, ainda, duas *guidelines*, muito utilizadas no mundo todo, que direcionam desenvolvedores na criação de *interfaces*, que são a *Microsoft Guidelines* e o *GNOME Human Interface Guidelines*. As referências destas fontes são apresentadas na seção 6, Referências.

3.2.2. Hardware

Para o desenvolvimento do trabalho foi utilizado um computador com processador Intel core i3 2.30 GHz de 64 bits, barramento USB 3.0 e 2.0 e 6 GB de RAM.

3.2.3. Software

Foi necessária a utilização do *software* em estudo, FísioKinect, com o objetivo de obter conhecimento suficiente para se alcançar os resultados esperados e um *software* de edição de texto.

3.2.4. Local de desenvolvimento

Para que a pesquisa e o desenvolvimento da *guideline* fosse possível, foi utilizado o Laboratório de Multimídia (LabMídia) e o Laboratório de Tecnologia em Saúde (LTS), ambos localizados no Complexo de Informática do Centro Universitário Luterano de Palmas (CEULP/ULBRA), que pertence ao grupo de Pesquisa Tecnologia, Saúde e Qualidade de

Vida. Esse grupo tem por objetivo realizar pesquisas com o apoio do sensor *microsoft kinect* para auxílio em procedimentos terapêuticos.

O presente Trabalho é requisito parcial para a disciplina de Trabalho de Conclusão de Curso II em Sistemas de Informação (TCC II) e foi realizado durante o primeiro semestre de 2017.

4. RESULTADOS E DISCUSSÃO

Nesta seção serão apresentados os resultados obtidos no desenvolvimento deste projeto, que tem por objetivo principal propor uma *guideline* para o desenvolvimento da *interface* do sistema FisioKinect, onde serão sugeridas as melhores formas para se desenvolver uma *interface* que satisfaça o usuário final, contando com acessibilidade e usabilidade.

4.1. A GUIDELINE

Para chegar à *guideline* desenvolvida, foram utilizadas como base as *guidelines* do GNOME e da *Microsoft* além de estudos de trabalhos de autores sobre esses temas. A próxima seção apresenta as diretrizes de desenvolvimento de *Interface* Homem-Computador para o FisioKinect.

4.1.1. Princípios básicos da aplicação

Deve-se compreender que essas diretrizes de *interface* servem para auxiliar no desenvolvimento das “telas” para o FisioKinect, propiciando uma maior probabilidade de se desenvolver um *software* mais eficaz e consistente. É válido ressaltar que a primeira experiência do usuário com a aplicação conta muito para a aprovação da mesma, visto que, caso o usuário julgue a *interface* como difícil de trabalhar, provavelmente ele irá reprovar ou perder o interesse sobre a aplicação.

Pensando nisso, é imprescindível que se desenvolva o necessário da forma mais clara possível, de forma direta e simples para que o usuário possa atingir o objetivo principal da aplicação sem se perder. A tabela 1, a seguir, apresenta atributos de um bom e de um mau *software* baseado no *windows* e *linux*.

Tabela 1 - Atributos de um bom e de um mau *software*

Atributos de um Bom <i>Software</i>	Atributos de um Mau <i>Software</i>
Estabelece um estilo claro, conciso e com personalidade.	Não possui identidade própria.
Cria uma conexão emocional.	O usuário não consegue navegar muito tempo por não ser atrativo.
Tem alta qualidade.	Qualidade baixa.
Possui um nome para o <i>software</i> que seja memorável e que	Não pensa no benefício do usuário.

transmita o benefício da aplicação.	
Obtenha ajuda de usuários experientes na validação.	Desenvolvido sem a parceria com o usuário final.

Fonte: próprio autor

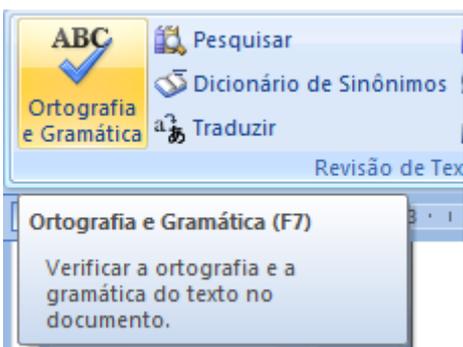
Nota-se que a tabela se resume ao fato de que uma *interface* de qualidade e que satisfaça ao usuário final deve ser pensada nos possíveis tipos de usuários e tendo o apoio deles na fase de validação. Isso serve tanto para a *interface* como para o *software* em si.

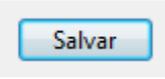
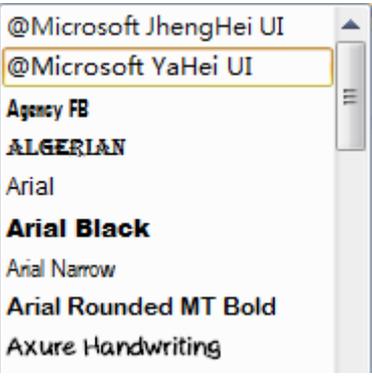
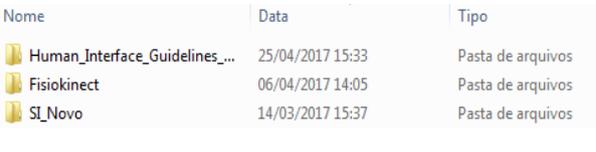
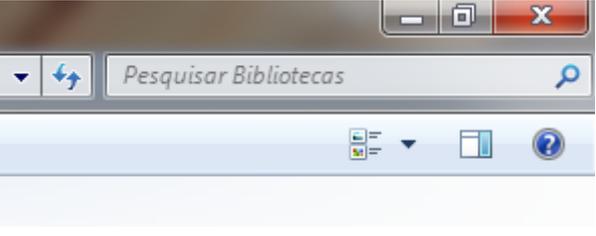
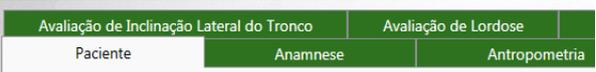
Nas próximas seções, serão apresentadas diretrizes sobre pontos essenciais de uma *interface*, voltada para o FisioKinect. Eles são baseados em *guidelines* muito utilizadas no mundo todo, que direcionam desenvolvedores para as plataformas *Windows e Linux*.

4.1.2. Controles

Existem vários tipos de controles disponíveis para serem usados, mas cada controle possui um fim específico e que melhor se enquadra às necessidades da aplicação em questão. Assim, cabe ao desenvolvedor conhecer cada tipo e função dos controles que lhe são disponíveis para que possa escolher o que melhor se adéqua em cada parte da *interface*. Pensando nisso, serão apresentadas na sequência (Tabela 2) um conjunto de controles e suas funções para que o desenvolvedor opte pela mais adequada ao seu projeto.

Tabela 2 - Controles

Controle	Exemplo	Função
<i>Check Box</i>		Permitir ao usuário selecionar uma ou mais opções em um dado contexto
<i>Balloon</i>		Informar problema não crítico ou condições especiais de um controle

<i>Command Buttons</i>		Permitir que os usuários executem uma ação imediata												
<i>List Boxes</i>		Permitir ao usuário selecionar um item da lista a partir de um conjunto de itens apresentados no <i>List Boxes</i> e sempre visíveis												
<i>List View</i>	 <table border="1"> <thead> <tr> <th>Nome</th> <th>Data</th> <th>Tipo</th> </tr> </thead> <tbody> <tr> <td>Human_Interface_Guidelines_...</td> <td>25/04/2017 15:33</td> <td>Pasta de arquivos</td> </tr> <tr> <td>Fisiokinect</td> <td>06/04/2017 14:05</td> <td>Pasta de arquivos</td> </tr> <tr> <td>SI_Novo</td> <td>14/03/2017 15:37</td> <td>Pasta de arquivos</td> </tr> </tbody> </table>	Nome	Data	Tipo	Human_Interface_Guidelines_...	25/04/2017 15:33	Pasta de arquivos	Fisiokinect	06/04/2017 14:05	Pasta de arquivos	SI_Novo	14/03/2017 15:37	Pasta de arquivos	Permitir ao usuário visualizar e interagir com uma coleção de objetos de dados, usando seleção única ou múltipla
Nome	Data	Tipo												
Human_Interface_Guidelines_...	25/04/2017 15:33	Pasta de arquivos												
Fisiokinect	06/04/2017 14:05	Pasta de arquivos												
SI_Novo	14/03/2017 15:37	Pasta de arquivos												
<i>Notifications</i>		Notificar ao usuário sobre eventos que não estão relacionados com a atividade atual do usuário												
<i>Progress Bars</i>		Permitir ao usuário acompanhar o andamento de uma operação demorada												
<i>Radio Buttons</i>	Sexo: <input checked="" type="radio"/> Masculino <input type="radio"/> Feminino	Permitir ao usuário a fazer apenas uma escolha entre um conjunto de escolhas mutuamente definidas												
<i>Search Box</i>		Fornecer ao usuário uma maneira ágil de se localizar objeto ou texto específico												
<i>Tabs</i>		Fornecer ao usuário uma maneira rápida de mudar de página através de abas rotuladas separadamente												
<i>Text Box</i>	Endereço: <input type="text"/>	Permitir ao usuário inserir, exibir ou editar um texto ou valor numérico												

Tooltips		Serve para rotular um controle sem rótulo
----------	---	---

Fonte: *Microsoft* (2017)

Nesta tabela foram apresentados alguns comandos básicos e as suas funções, bem como exemplos. Fica a critério do desenvolvedor escolher os comandos mais adequados para cada tela a ser desenvolvida, lembrando que pode haver comandos não especificados nesta *guideline*, mas que se fazem necessários para uma boa usabilidade do produto.

4.1.3. Disposição visual

O *layout* visual do conteúdo, afeta o entendimento que o usuário terá da aplicação, bem como se possui uma *interface* atraente (DAY, 2014). Um mau *software* exige mais esforço do usuário adicionando funcionalidades desnecessárias, ou não necessárias naquele momento, enquanto uma *interface* usual e acessível exige menos esforço.

Este tópico busca ajudar o desenvolvedor a construir um sistema bonito, de fácil compreensão e eficiente. Destacam-se a seguir algumas dicas para um bom *layout* (DAY, 2014):

1. Um ponto de alinhamento é uma linha vertical ou horizontal imaginária através da janela que toca a borda de um ou mais controles ou rótulos na janela. Quanto menos pontos de alinhamento houver, mais limpo e simples o *layout* será, e mais fácil para as pessoas entenderem;
2. Alinhar exatamente o conteúdo e controles do seu *layout*. As pessoas percebem facilmente objetos alinhados e desalinhados. Elementos desalinhados se tornam perturbadores ao olhar e podem passar despercebidos; A seguir a figura 12 apresenta um *layout* desalinhado e a figura 13 apresenta um *layout* alinhado.

Figura 12 - Exemplo de *layout* desalinhado

The image shows a web form for patient registration, titled "Anamnese". The form is misaligned, with various fields not properly centered or aligned with each other. The form is framed by a green border. The form contains the following fields:

- Paciente: Anamnese (highlighted in green)
- Antropometria
- Amplitude de Movimento
- Nome:
- Data Nascimento: Selecionar um:
- Sexo: Masculino Feminino
- Situação Ocupacional:
- Estado civil:
- Escolaridade:
- Telefone Principal:
- E-mail Principal:
- Telefone Secundário:
- E-mail Secundário:
- Endereço:
- Bairro:
- CEP:
- Casa:
- Complemento:
- Estado:
- Cidade:

Buttons: Cadastrar (green), Sair (red)

Fonte: FisisKinect

Figura 13 - Exemplo de *layout* alinhado

The screenshot shows a web interface for patient registration. The form is titled 'Anamnese' and is part of a larger system with tabs for 'Paciente', 'Anamnese', 'Antropometria', and 'Amplitude de Movimento'. The form fields are organized into two columns and include:

- Nome: [text input]
- Data Nascimento: [date picker]
- Sexo: Masculino Feminino
- Situação Ocupacional: [dropdown menu]
- Estado civil: [dropdown menu]
- Escolaridade: [dropdown menu]
- Telefone Principal: [text input with area code mask]
- E-mail Principal: [text input]
- Telefone Secundário: [text input with area code mask]
- E-mail Secundário: [text input]
- Endereço: [text input]
- Bairro: [text input]
- CEP: [text input with mask] Casa: [text input]
- Complemento: [text input]
- Estado: [dropdown menu]
- Cidade: [dropdown menu]

At the bottom of the form, there are two buttons: a green 'Cadastrar' button and a red 'Sair' button.

3. Controles e informações devem estar organizados, relacionados em grupos e use o espaçamento para diferenciá-los. Isso torna a *interface* de mais fácil leitura e compreensão;
4. Deixar o espaço entre os elementos em incremento de 6 *pixels*, subindo à medida que a relação entre elementos relacionados se torna mais distantes;
5. Entre rótulos e componentes associados, o espaçamento deve ser de 12 *pixels* horizontais;
6. Espaçamento de 18 *pixels* entre grupos de componentes;
7. Recomenda-se também uma distância de 18 *pixels* entre o conteúdo de uma janela de diálogo e as bordas da janela;
8. Deve-se utilizar sempre o mesmo espaçamento, manter a consistência é essencial; A figura 14 apresenta exemplo de um *layout* alinhado e consistente.

Figura 14 - Exemplo de *layout*

The image shows a web form titled 'Avaliação de Inclinação Lateral do Tronco' (Lateral Trunk Tilt Assessment). The form is organized into a grid with four main tabs: 'Avaliação de Inclinação Lateral do Tronco', 'Avaliação de Lordose', 'Percepção Corporal', and 'Silhueta de Kakeshita'. The first tab is active, showing a 'Paciente' (Patient) section with various input fields. The form is styled with a green header and footer, and a white background for the content area. The input fields are arranged in a clear, organized manner, with labels and values clearly visible. At the bottom of the form, there are two buttons: 'Editar' (Edit) in green and 'Sair' (Exit) in red.

Avaliação de Inclinação Lateral do Tronco		Avaliação de Lordose		Percepção Corporal		Silhueta de Kakeshita	
Paciente		Anamnese		Antropometria		Amplitude de Movimento	
Nome:	Fulano de Tal			Data Nascimento:	01/01/0001		
Sexo:	<input checked="" type="radio"/> Masculino	<input type="radio"/> Feminino	Situação Ocupacional:	Estudante			
Estado civil:	Solteiro			Escolaridade:	Ensino Médio Incompleto		
Telefone Principal:	(00) 00000-0000			E-mail Principal:	fulanodetal@algumacoisa.com		
Telefone Secundário:	(00) 00000-0000			E-mail Secundário:	Fulanodetal2@algumacoisa.com		
Endereço:	1501 Sul			Bairro:	1501 Sul		
CEP:	77.270-000	Casa:	0	Complemento:	s/ complemento		
Estado:	Tocantins			Cidade:	Palmas		

Buttons: **Editar** (green), **Sair** (red)

9. Recomenda-se não utilizar quadros com bordas visíveis para separar grupos de controles, usa-se espaçamento e cabeçalhos ao invés disso.

10. Os elementos visuais devem ser organizados de cima para baixo e da esquerda para direita. Esta é a direção que as pessoas de locais ocidentais tendem a ler uma *interface*, de modo que os itens na parte superior esquerda serão encontrados em primeiro lugar;

11. Os componentes que são vistos primeiro, são vistos como prioridade sobre aqueles que vêm depois deles. Assim, é interessante colocar os controles principais acima e à esquerda dos controles e conteúdos que eles afetam;

4.1.4. Comandos

Os comandos são ações que o usuário pode executar ao usar a plataforma. Aqui serão apresentadas as diretrizes para adicionar comandos aos menus do sistema. Os menus são listas hierárquicas e eles possuem os seguintes padrões de uso e condições para serem aplicados (MICROSOFT, 2017):

Tabela 3 - Comandos de Menus

Tipo de Menu	Padrões de uso	Aplicam-se as seguintes questões?
<i>Menu Bars</i>	O <i>menu bars</i> exibe comandos e opções dos menus suspensos e são muito comuns e fáceis de encontrar, bem como demonstram um uso eficiente de espaço.	<ul style="list-style-type: none"> • É uma janela principal? • Existem muitos itens de menu? • Existem muitas categorias de menu? • A maioria dos itens de menu se aplica a todo o programa e à janela principal? • O menu precisa funcionar para todos os usuários?
<i>Toolbar Menus</i>	É implementada como uma barra de ferramentas que consiste principalmente de comandos em botões de menus e botões de divisão, com apenas alguns comandos diretos, se houver.	<ul style="list-style-type: none"> • A janela é uma janela principal? • A janela tem uma barra de ferramentas? • Existem apenas algumas categorias de menu? • O menu precisa funcionar para todos os usuários?
<i>Tab Menus</i>	Caracteriza-se por ter botões dentro de guias que exibem um pequeno conjunto de comandos e opções relacionadas a uma guia em um menu suspenso.	<ul style="list-style-type: none"> • A janela é uma janela principal? • A janela tem guias, onde cada guia é utilizada para um conjunto de tarefas dedicado (Ao contrário de usar guias para exibir diferentes modos de exibição)? • Existe uma categoria de menu que se aplica a cada guia? • Existem muitos comandos e opções, mas apenas um pequeno conjunto para cada guia?
<i>Menu Buttons</i>	São botões de comando que exibem um pequeno conjunto de comandos relacionados em um menu suspenso. Quando eles veem com uma seta à direita, significa que ao clicar sobre ele é exibido um menu suspenso com mais opções de comando relacionado à aquele botão.	
<i>Context Menus</i>	É quando um usuário clica com o botão direito do <i>mouse</i> em um objeto ou região da janela que suporta um menu de contexto.	<ul style="list-style-type: none"> • Existe um pequeno conjunto de comandos contextuais e opções que se aplicam ao objeto selecionado ou região da janela? • Estes itens de menus são redundantes? • O usuário final está familiarizado com o menu de contexto?
<i>Task Pane Menu</i>	Ao contrário dos menus de contexto, eles são exibidos automaticamente dentro de um painel de janela, em vez de sob demanda.	<ul style="list-style-type: none"> • A janela é uma janela principal? • Existe um pequeno conjunto de comandos contextuais e opções que se aplicam ao objeto selecionado ou ao modo do programa?

		<ul style="list-style-type: none"> • Existem algumas categorias de menu? • O menu precisa funcionar para todos os usuários?
--	--	---

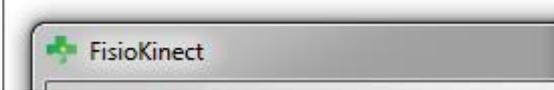
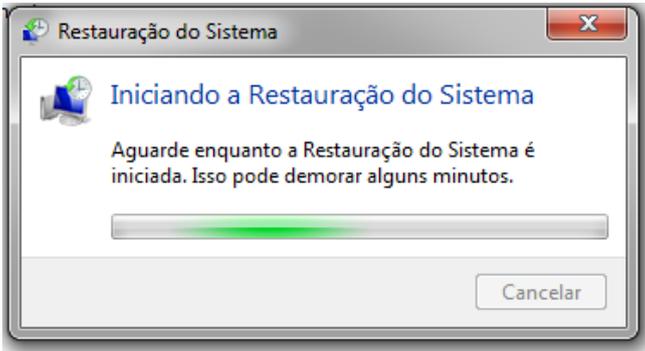
Fonte: Próprio autor

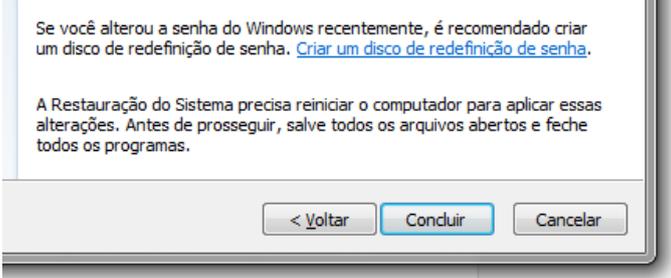
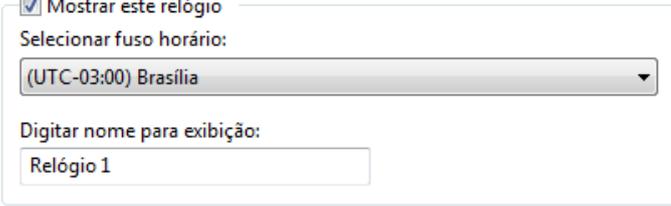
A tabela anterior apresenta tipos de menus, descreve quanto utilizá-los e os fatores a se observar para utilizá-los. Vale lembrar que o desenvolvedor deve tomar cuidado com a utilização de submenus, que são menus dentro dos menus. O cuidado é para não estender demais, quando maior o caminho, mais difícil de ser encontrado.

4.1.5. Estilo de escrita

O estilo de escrita não está ligado diretamente nas palavras, mas como utiliza-las. Isso implica dizer que é necessário passar ao usuário uma emoção específica, a fim de fazer o mesmo se envolver com a aplicação. Muitos desenvolvedores pensam que os textos são uma particularidade da documentação do produto, mas deve-se pensar que textos simples e claros na *interface* são essenciais para uma boa experiência do usuário com o *software*. Na tabela 4 a seguir serão explicados e exemplificados padrões de escrita utilizados em uma *interface* de usuário.

Tabela 4 - Padrões de Escrita

<p>Texto do Title Bar</p> <p><i>title bar</i> de texto para identificar uma janela ou a fonte de uma caixa de diálogo.</p>	 <p>Neste exemplo, o texto da barra de título identifica uma janela.</p>
<p>Instruções Principais</p> <p>Serve para explicar de forma concisa o que fazer na janela.</p>	<p>Uma boa instrução tem por objetivo comunicar com o usuário ao invés de focar apenas em manipular a <i>interface</i>.</p>  <p>Neste exemplo o texto de instrução principal envolve diretamente o usuário em termos de benefício próprio ou do interesse do usuário.</p>

<p>Instruções Suplementares</p> <p>Utilizada para apresentar informações adicionais úteis para a compreensão.</p>	 <p>Se necessário, fornecer informações mais detalhadas, fornecer o contexto e definir a terminologia.</p>
<p>Etiquetas de Controle</p> <p>Usar diretamente ao controle ou próximo a ele.</p>	 <p>Neste exemplo as etiquetas de controle identifica as definições de data e hora que o usuário pode selecionar ou modificar.</p>
<p>Explicações suplementares</p> <p>Um detalhamento das etiquetas de controle.</p>	<p>Compartilhamento de pasta pública</p> <p>Quando um compartilhamento de Pasta pública está ativado, as pessoas na rede, incluindo membros do grupo doméstico, podem acessar arquivos nas Pastas públicas. O que são as Pastas públicas?</p> <ul style="list-style-type: none"> <input checked="" type="radio"/> Ativar compartilhamento para que qualquer pessoa com acesso à rede possa ler e gravar arquivos nas Pastas públicas <input type="radio"/> Desativar compartilhamento de Pasta pública (pessoas que fizeram logon neste computador ainda podem acessar essas pastas) <p>Neste exemplo, as explicações suplementares esclarecem as escolhas.</p>

Fonte: *Microsoft* (2017)

É importante saber que, se a *interface* está difícil de ser explicada, talvez não seja a forma de explicar que tenha que mudar, mas sim o *design* do produto (MICROSOFT, 2017). Considerar também as seguintes dicas para o modelo de texto da aplicação:

1. Nas culturas ocidentais, o leitor costuma ler da esquerda para a direita e de cima para baixo;
2. Durante a utilização do *software* os usuários não estão ligados no produto, mas em seu trabalho. Assim, os usuários acabam não lendo todo o texto de uma *interface*, seja simples e direto, mas não incompleto, avalie cada situação;
3. O usuário sempre vai buscar completar sua ação o mais rápido possível, focando a atenção primeiramente nos rótulos dos controles, especialmente aqueles relevantes para completar a tarefa em mãos. O texto estático só é lido quando eles acham que precisam ler;

4. Mesmo sabendo da cultura ocidental, não se deve acreditar que o usuário vai ler todo o texto da *interface* e nem que ele vai seguir a cultura ocidental. O usuário sempre irá olhar toda a *interface* e em seguida ler o conteúdo da seguinte ordem: controles interativos; botões de comando; instrução principal; explicações suplementares; título da janela, outro texto estático; notas de rodapé;
5. A redundância toma tempo do usuário e espaço na tela.
6. *Links* para o conteúdo Ajuda são eficazes;
7. O uso do negrito é regulado, aplica-se apenas para enfatizar o que for mais importante para dados como um todo. Não costuma se utilizar negrito e itálico em uma *interface*;
8. Utilizar o sublinhado apenas para *links*;
9. Não colocar ponto final em etiquetas de controle, instruções principais ou *links* de ajuda;
10. Terminologias ou jargões devem ser analisados antes de se utilizar, alguns usuários podem não entender ou interpretar mal;
11. Para uma maior uniformidade, indica-se a utilização de uma única fonte de texto.

4.1.6. Mensagens

Elas são qualquer tipo de mensagem que o usuário quer ou necessita ver ao usar o sistema. Existem três tipos principais de mensagens, a saber: Mensagens de Erro; Mensagens de Aviso; e Confirmações (MICROSOFT, 2017).

Uma mensagem de erro alerta o usuário de um problema que já ocorreu, informa a causa explicando por que o problema ocorreu e fornece uma solução para que o usuário possa corrigir o problema. E para fazer isso da melhor forma, a mensagem de erro deve ser apresentada considerando os pontos descritos na Tabela 5.

Tabela 5 - Pontos Mensagens de Erro

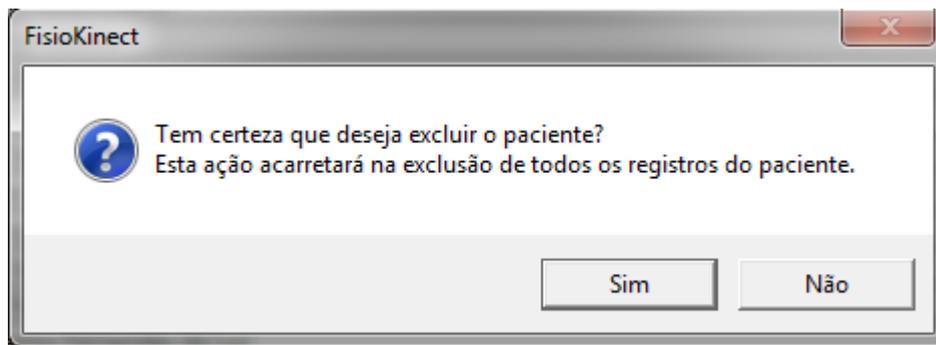
Principais Pontos	Motivo
Relevante	Apresentar um problema que os usuários se preocupam.
Acionável	Os usuários devem poder executar uma ação como resultado da mensagem.
Centrado no usuário	A mensagem deve descrever o erro em termos entendíveis pelo usuário.

Breve	A mensagem deve ser o mais curta possível, mas não incompleta.
Clara	A linguagem deve ser específica e pensada no usuário para que haja compreensão do problema e da solução.
Específico	A mensagem deve ser específica, dando nomes específicos aos locais e objetos envolvidos.
Cortês	O usuário não pode se sentir culpado do erro.
Rara	Mensagens de erro exibidas com muita frequência são um sinal de design ruim.

Fonte: *Microsoft* (2017)

Uma mensagem de aviso alerta o usuário de uma condição que pode causar um problema no futuro. A figura 15 a seguir apresenta um exemplo de mensagem de aviso.

Figura 15 - Exemplo de mensagem de aviso

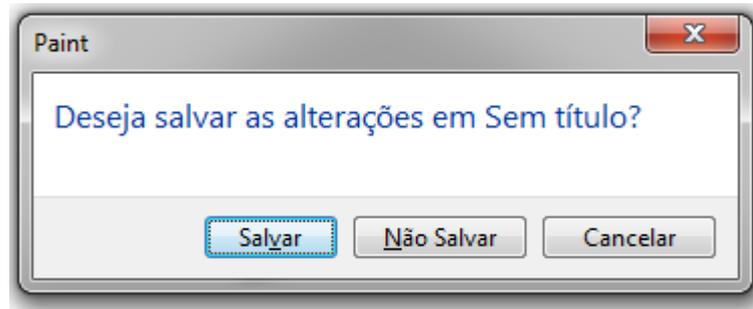


Fonte: FisioKinect

Este tipo de mensagem tem por característica principal advertir sobre riscos de se perder dados importantes como dados de avaliações que ainda não foram salvos e exclusão ou edição de dados importantes. Bons avisos alertam de algo significativo, com relevância imediata, que leva a ação imediata e deve ocorrer com pouca frequência.

As mensagens de confirmação são bem simples, ela é uma caixa de diálogo que pergunta se o usuário deseja prosseguir com uma ação. Elas possuem como principais características: serem apresentadas como resultado direto de uma ação iniciada pelo usuário, verificar se o usuário deseja prosseguir com a ação e consiste com uma pergunta simples e duas ou mais respostas, (MICROSOFT, 2017), como pode-se verificar na figura 16 abaixo.

Figura 16 - Exemplo de mensagem de confirmação



Fonte: *Paint*

4.1.7. Ícones

De acordo com Day (2014), os ícones possuem um papel importante na *interface*. Utilizar o ícone correto na aplicação é essencial para garantir que o sistema seja utilizado. Existem situações onde se é necessário decidir usar um ícone ou um rótulo de texto, especialmente para botões. Os ícones possuem a vantagem de serem menores, mas o uso incorreto de um ícone pode tornar a *interface* difícil de ser compreendida.

Deve-se utilizar apenas ícones cujo significado é usualmente reconhecido e caso não encontre esse tipo de ícone, talvez seja melhor utilizar um rótulo de texto. O desenvolvedor deve considerar quais ícones serão significativos no contexto específico da aplicação. Vale lembrar que alguns ícones só serão significativos ao usuário quando estiverem ao lado de outro ícone do mesmo tipo. Por exemplo, um ícone para remover um item de uma lista é um símbolo de subtração, ele acaba não possuindo significado se não vier acompanhado de um ícone de adição correspondente (DAY, 2014).

Caso o desenvolvedor não encontre o ícone ideal e seja mesmo necessária a utilização de um ícone, pode-se construir um ícone seguindo os padrões dos demais ícones utilizados ao longo da *interface*. Procurar sempre utilizar ícones em um ângulo reto de 90°, estas costumam ser mais rápidas para processar pelo sistema visual humano.

4.1.8. Interação

A interação é uma variedade das formas que os usuários interagem com o sistema: *mouse*, teclado, toque, movimentos naturais do corpo, fala, entre outros. Nesse caso, o tipo de interação é por *mouse*, teclado e natural. O *mouse* tem sido um dispositivo de entrada de sucesso porque é intuitivo e fácil de usar, para esse tipo de interação não tem segredo, basta atribuir comportamentos de interação do *mouse* consistentes com seus efeitos padrão. Limite interações avançadas como duplo clique e cliques com o botão direito.

O teclado é o dispositivo principal para a entrada de texto, não existe segredo quanto à sua utilização. Onde existirem campos onde o usuário necessite inserir ou editar dados, isso será feito com o uso do teclado.

No que tange aos movimentos naturais do corpo, nesta aplicação, o *kinect* é utilizado na hora das avaliações dos pacientes, ou seja, essa captura dos movimentos naturais é uma forma de avaliação e os dados de entrada servem para avaliar os pacientes.

A acessibilidade deve ser levada em consideração em uma interação. A tabela 6 a seguir apresenta alguns tipos de deficiência comum nas pessoas e uma lista de algumas soluções mais importantes para ajudar a tornar o FisioKinect mais acessível.

Tabela 6 - Tipos de Deficiência

Tipo de Deficiência	Descrição	Solução
Visual	Varia de leve (afetando 17% dos usuários) a grave (afetando 9% dos usuários)	Ampliação, cores e contrastes personalizáveis
Audição	Varia de leve (afetando 18% dos usuários) a grave (afetando 2% dos usuários)	Redundância de informação: som utilizado apenas como suplemento ao texto ou comunicação visual
Destreza	Varia de leve (afetando 19% dos usuários) a grave (afetando 5% dos usuários). Caracteriza pela dificuldade de executar certas habilidades motoras com o teclado ou <i>mouse</i> .	Redundância do método de entrada: recursos do programa acessados por equivalência entre o <i>mouse</i> ou teclado.
Cognitivo	Incluiu prejuízos de memória e diferenças perceptuais. Afeta 16% dos usuários.	<i>Interface</i> altamente personalizável. Uso de ícones e outras ajudas visuais.
Apreensão	Inclui sensibilidade visual ao movimento.	Abordagem conservadora quanto ao uso de animações.

Fonte: *Microsoft* (2017)

A tabela 6 serve como uma diretriz importante em relação à acessibilidade. A *interface* deve ser projetada pensando em todo o tipo possível de usuário. Por mais que não seja possível contemplar todos esses pontos em uma única *interface*, pode-se, por exemplo, oferecer opções de tipos de *interface* conforme o tipo de deficiência.

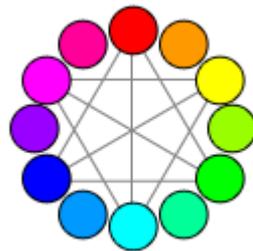
4.1.9. Cor

A cor é um elemento visual importante na *interface*. As cores possuem significados, provocam respostas emocionais e fazem a plataforma parecer mais agradável e refinada, por isso devem ser utilizadas de forma adequada. A má utilização da cor pode ser pior do que não usar a cor, deve ser feita uma análise também cultural ao se utilizar uma cor. Por exemplo, no Brasil, trajes de casamento para noiva é associado com a cor branca enquanto o preto é associado com funerais. No entanto, no Japão o simbolismo das cores era exatamente o oposto, o branco era a cor predominante em funerais e o preto era considerado uma cor que trazia sorte ao casamento (MICROSOFT, 2017).

As cores também contribuem para a acessibilidade do *software* para o maior número de usuários possível. Usuários que possuem algum tipo de deficiência visual podem não conseguir ver as cores bem. Cerca de 8% dos homens adultos têm algum tipo de confusão sobre cor, dos quais a confusão com vermelho e verde é a mais comum. As cores normalmente utilizadas no status são vermelha, amarela e verde, onde a vermelha significa um problema ou erro, o amarelo significa uma advertência e o verde é algo bom. Esses significados para essas cores são globais, uma vez que a Convenção de Viena da UNESCO sobre Sinalização Rodoviária definiu essas cores para os semáforos, onde vermelho significa pare, amarelo significa atenção e o verde significa prosseguir (MICROSOFT, 2017). Assim, pode-se utilizar essas cores sem se preocupar com a questão cultural.

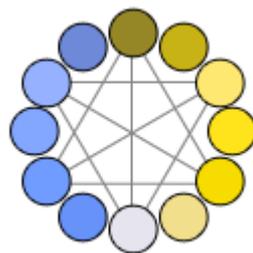
Nas figuras 17, 18, 19 e 20 a baixo, serão exemplificadas as cores primárias vista de forma normal e com algum tipo de deficiência.

Figura 17 - Cores primárias visão normal



Fonte: *Microsoft* (2017)

Figura 18 - Cores primárias visão Protan



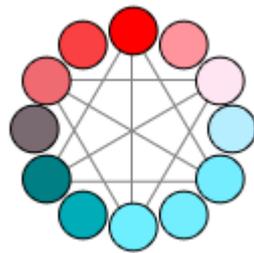
Fonte: *Microsoft* (2017)

Figura 19 - Cores primárias visão Deuteranopia



Fonte: *Microsoft* (2017)

Figura 20 - Cores primárias visão Tritanopia



Fonte: *Microsoft* (2017)

O ideal é utilizar uma cor tema, que no caso do FísioKinect é o verde. Não utilize a cor como o principal método de comunicação, mas como secundário a fim de reforçar o que significa visualmente. Outra dica é procurar projetar em escala monocromática para garantir que a cor se torne secundária.

4.1.10 *Interface Natural*

Sabe-se que as *interfaces* gráficas utilizam-se de dispositivos como teclado e *mouse* para que o usuário possa se comunicar com o sistema. No entanto, as *interfaces* naturais, permitem a interação direta do usuário com o sistema. Atualmente o FísioKinect possui uma *interface* gráfica onde o usuário utiliza-se de periféricos para se comunicar com a plataforma e uma *interface* natural onde o paciente é avaliado.

Não distante, acredita-se que a plataforma em estudo venha a evoluir e sua interação com o usuário profissional possa ser mais natural, por meio do toque, gestos e fala. A seguir serão apresentados 12 princípios básicos para se projetar uma *interface* natural de sucesso. Esses princípios foram sugeridos por (SAFFER, 2011) e servem para o FísioKinect:

- 1 Projetar para dedos e não para cursores: caso a *interface* venha suportar o "toque", as áreas de toque precisam ser muito maiores do que em um sistema para o *mouse*. Elas devem ser de 8 - 10 mm para canetas e 10 - 14 mm para as pontas dos dedos.
- 2 Fisiologia e cinesiologia não devem ser esquecidas: o usuário não deve executar tarefas genéricas ou repetitivas.
- 3 Sem braço de Gorila: o homem não foi feito para fazer muitas tarefas com as mãos para cima, a frente de seu corpo, por um longo período de tempo.
- 4 Cobertura de tela: os dedos estão ligados a uma palma, que pode cobrir a tela enquanto o usuário está tentando fazer um gesto ou toque. Elementos essenciais, como rótulos, não devem ser alinhados à baixo de um controle, de forma que possa ser encoberto pela própria mão do usuário. Uma sugestão seria sempre tentar colocar os rótulos a cima dos controles. Outra sugestão é colocar o menu na parte inferior da *interface*.
- 5 Conhecer a tecnologia: o tipo de tela sensível ao toque, sensor ou câmera é que determina o tipo de gestos que você pode projetar para a interação. Atualmente o FizioKinect não trabalha com a tela sensível ao toque, mas trabalha com o kinect v1 que possui muitos recursos, como especificados na seção 2.8 deste trabalho.
- 6 Quanto mais difícil for o gesto, menos pessoas serão capazes, ou desejaram, executá-lo.
- 7 Ativar ações quando o usuário remover o dedo, e não durante o toque na tela.
- 8 Reconhecimento: para que os usuários se sintam atraídos a utilizar o sistema, deve-se utilizar gestos simples e intuitivos.
- 9 Evitar a ativação de ações de forma não intencional: movimentos naturais diários de uma pessoa pode acidentalmente acionar o sistema, isso não deve ocorrer.
- 10 Gestos e tecla de comando: dispõem ao usuário maneiras fáceis de acessar a funcionalidade, mas também formas avançadas e ágeis de gestos aprendidos como atalhos.
- 11 Variedade de requisitos: um mesmo gesto pode ser realizado de várias maneiras. Deve-se ficar atento a isso.
- 12 Determinar a complexidade do gesto de acordo com a complexidade e a frequência da tarefa: isso implica dizer que, as tarefas simples e frequentemente usadas devem possuir gestos simples para executá-las.

O FizioKinect ainda possui pouca utilização da NUI, mas acredita-se que futuramente a plataforma possa atualizar a sua *interface*, propiciando ao usuário uma maior interação natural. Assim, é importante que os novos desenvolvedores se atentem que uma nova

interface deve ser construída, pois a *interface* atual foi desenvolvida em grande parte para a GUI. É errado pensar que a *interface* GUI vai servir para NUI, o melhor é construir uma nova *interface* seguindo as recomendações aqui apresentadas.

A plataforma em estudo já atende um pouco do que é proposto, é consistente, possui elementos comuns, *feedback* com o usuário, etc. O sistema necessita apenas que se adequar em pequenos detalhes que talvez passaram despercebidos aos olhos do desenvolvedor. Por fim, muito do que foi proposto aqui a ferramenta já aplica.

5. CONSIDERAÇÕES FINAIS

O presente trabalho teve o objetivo de desenvolver uma *guideline* que fosse capaz de direcionar os desenvolvedores das *interfaces* do *software* FizioKinect. O guia busca reunir informações básicas sobre controles, tipos de mensagens, cores, tamanho de botões, espaçamento, disposição visual, entre outros, a fim de sugerir a melhor forma de se utilizar os elementos de uma *interface*. Lembrando que a *guideline* não é para ser necessariamente seguida à risca, nela estão contidas diretrizes básicas para se desenvolver uma boa *interface* e cabe ao desenvolvedor ter a capacidade de decidir a melhor forma de usar o recurso que lhe é disponível.

Inicialmente, foram realizados estudos sobre os temas relacionados ao trabalho, onde, através de referências bibliográficas, foi possível entender e aprofundar o conhecimento sobre conceitos como a história da IHC, *guidelines*, usabilidade, acessibilidade e o próprio FizioKinect.

Após os estudos dos temas e construção do referencial teórico, foi necessário encontrar guias de *interface* para se utilizar como parâmetros. Assim, foram escolhidos dois guias bastante conhecidos e que direcionam na construção das *interfaces* mais utilizadas em todo o mundo, *Microsoft Guidelines* e o *GNOME Human Interface Guidelines*.

Os resultados desse trabalho são apresentados na seção 4, onde foram observados os principais pontos dos outros guias e que atendiam as necessidades do trabalho em questão. Posteriormente, chegou-se aos pontos principais que foram abordados na *guideline* do FizioKinect. Devido ao fato das duas *guidelines* utilizadas como parâmetro não falarem sobre NUI, foi necessário utilizar de estudos sobre autores especialistas no assunto como Dan Saffer.

Com o surgimento do *kinect v2*, será de grande valia ser desenvolvida uma nova *interface* para o FizioKinect considerando todos os pontos aqui abordados e voltada para o *kinect v2* que possui muito mais recursos que o seu antecessor, o *kinect v1* que atualmente é utilizado.

Conclui-se, portanto, que desenvolver um guia capaz de direcionar os novos desenvolvedores do sistema nos principais pontos em relação à *interface* não é fácil. No entanto, foi possível sugerir diretrizes básicas para o desenvolvimento da *interface* do FizioKinect, cabendo aos novos desenvolvedores analisar as dicas contidas na *guideline*, pois no guia não estão contidas regras, mas sim diretrizes de desenvolvimento.

REFERÊNCIAS

- ABNT - ASSOCIAÇÃO BRASILEIRA DE NORMAS TÉCNICAS. **NBR ISO/IEC 9126-1:** Engenharia de Software - Qualidade do produto: modelo de qualidade. Rio de Janeiro, 2011.
- ANDRADE, Maria Margarida de. **Como preparar trabalhos para cursos de pós-graduação:** noções práticas. 5. ed. São Paulo: Atlas, 2002. 165 p., il.
- ANNA, Andrew de Castro Sant'; FERRONATO, Ana Carolina Clivatti. **Interface Naturais e Interfaces Tangíveis.** 2010. 1 v. Dissertação (Mestrado) - Curso de Computação, Universidade Federal Fluminense, Rio de Janeiro, 2010. Disponível em: <<http://www.professores.uff.br/screspo/artigoIHC2.pdf>>. Acesso em: 15 nov. 2016.
- BRASIL. **Lei 13.146, de 06 de julho de 2015.** Institui a Lei Brasileira de Inclusão da Pessoa com Deficiência (Estatuto da Pessoa com Deficiência). Brasília, 2015. Disponível em: <http://www.planalto.gov.br/ccivil_03/_ato2015-2018/2015/lei/113146.htm>. Acesso em: maio de 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. 12 f. Monografia (Especialização) - Curso de Ciência da Computação, Centro Universitário Luterano de Palmas, Palmas, 2015.
- CARDOSO, G.; SCHMIDT, A. Biblioteca de Funções para Utilização do Kinect em Jogos Eletrônicos e Aplicações NUI. **XXVI conference on**, 2012.
- COLPANI, R.; HOMEM, M. R. P. **Realidade Aumentada e Gamificação na Educação: uma aplicação para auxiliar no processo de aprendizagem de alunos com deficiência intelectual.** v. 24, p. 83–101, 2016.
- DAY, Allan. **GNOME Human Interface Guidelines.** The GNOME project, 2014. Disponível em: <<https://developer.gnome.org/hig/stable/>>. Acesso em: abr 2017.
- DIAS, Ana Luiza. **Um processo para sistemas web com foco em acessibilidade.** Tese (Doutorado – Programama de Pós-Graduação em Ciências de Computação e Matemática Computacional). Instituto de Ciências Matemáticas e Computação, Universidade de São Paulo, 2014.
- FLORES, Juan Carlos Coto; SOUZA, Anderson Carlos de. **Uso do Microsoft Kinect na Saúde: Características e Limitações da Ferramenta.** 2013. 11 f. Dissertação (Mestrado) - Curso de Design e Expressão Gráfica, Universidade Federal de Santa Catarina, Florianópolis, 2013. Disponível em: <<file:///C:/Users/102918410/Downloads/5855-16661-1-PB.pdf>>. Acesso em: 11 nov. 2016.
- GARBIN, Sander Maeda. **Estudo da Evolução das Interfaces Homem-Computador.** 2010. 90 f. Monografia (Especialização) - Curso de Engenharia Elétrica, Universidade de São Paulo, São Paulo, 2010. Disponível em: <<http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-25112011-104445/?&lang=br>>. Acesso em: 12 nov. 2016.
- GONÇALVES, Mileni Kazedani. **Usabilidade de software: estudo de recomendações básicas para verificação do nível de conhecimento dos alunos dos cursos de Design Gráfico e Sistemas de Informação da Unesp/Bauru.** 2008. 238 f. Dissertação (Mestrado) - Curso de Design, Universidade Estadual Paulista Júlio de Mesquita Filho, Bauru, 2008.

GOOGLE. **Introduction - Material design - Material design guidelines**. Disponível em: <<https://material.google.com/#>>.

GUIMARÃES, R. M. **Gestão do conhecimento desenvolvimento de um protótipo de software de reconhecimento facial de tempo real para registro eletrônico de ponto em ambientes indoor com utilização do dispositivo Kinect**. (mestrado profissional em sistemas de informação e gestão do conhecimento). Universidade FUMEC. Belo Horizonte, 2015.

HERLOCKER, J. L.; KONSTAN, J. A.; RIEDL, J. **Explaining collaborative filtering recommendations**. Proceedings of the 2000 ACM conference on Computer supported cooperative work - CSCW '00. **Anais...**New York, New York, USA: ACM Press, 2000.

HONG, J. Matters of design. Commun. ACM, ACM, New York, NY, USA, v. 54, n. 2, p. 10-11, 2011. ISSN 0001-0782. Disponível em: <<http://doi.acm.org/10.1145/1897816.1897820>>.

INTERFACE. **Dicionário Aurélio**. Disponível em: <<https://dicionariodoaurelio.com/interface>>. Acesso em: 28 de fev. 2017.

INTERFACE. **Dicionário Dicio**. Disponível em: <<https://www.dicio.com.br/interface/>>. Acesso em: 28 de fev. 2017.

JORGE, Rodrigo Funabashi. **Estudo, Definição e Proposta de Representação de Interface Web Visando à Atividade de Teste de Software**. 2016. 146 f. Dissertação (Mestrado) - curso de Ciência da Computação, Instituto de Informática, Universidade Federal de Goiás, Goiânia, 2016.

JUNIOR, J. A. et al. **Desenvolvimento de uma plataforma para auxílio na fisioterapia de pacientes com paralisia cerebral**. Curso de Ciência da Computação e Fisioterapia. Faculdade Anglo-Americano, Foz do Iguaçu, 2015.

LIMA JUNIOR, Guaratã Alencar Ferreira de. **Guia de boas práticas para desenvolvimento de interface e interação para desenvolvedores da plataforma Android**. 2016. 1 v. TCC (Graduação) - Curso de Sistemas de Informação, Faculdade Avantis, Balneário Camboriú, 2016. Disponível em: <<http://www.lbd.dcc.ufmg.br/colecoes/wicsi/2016/008.pdf>>. Acesso em: 17 nov. 2016.

LIU, W. **Natural User Interface – Next Mainstream Product User Interface**. 2010. Disponível em <<http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=5681374>> Acessado em: 15 de nov de 2016.

MACHADO NETO, Olibário José. **Usabilidade da interface de dispositivos móveis: heurística e diretrizes para o design**. 2013. 136 f. Dissertação (Mestrado) - Curso de Ciências de Computação e Matemática Computacional, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2013. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-07012014-110754/pt-br.php>>. Acesso em: 01 mar. 2017

MARTINS, Guilherme Augusto de Almeida. **Usabilidade das interações táteis em dispositivos móveis por pessoas idosas**. 2016. 139 f. Dissertação (Mestrado) - Curso de Ciências de Computação e Matemática Computacional, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos, 2016. Cap. 2. Disponível em:

<<http://www.teses.usp.br/teses/disponiveis/55/55134/tde-23112016-163643/pt-br.php>>. Acesso em: 15 mar. 2017.

MEDEIROS FILHO, Dante Alves; VIEIRA, Anderson. **Um Estudo Sobre as Interfaces Naturais**. 2013. 1 v. TCC (Graduação) - Curso de Computação, Departamento de Informática, Universidade Estadual de Maringá, Maringá, 2013. Disponível em: <[http://www.espweb.uem.br/site/files/tcc/2012/Anderson Vieira – Um estudo sobre interfaces naturais.pdf](http://www.espweb.uem.br/site/files/tcc/2012/Anderson%20Vieira%20-%20Um%20estudo%20sobre%20interfaces%20naturais.pdf)>. Acesso em: 16 nov. 2016.

MICROSOFT, **Guidelines**. 2017. <Disponível em: <[https://msdn.microsoft.com/pt-br/library/windows/desktop/dn688964\(v=vs.85\).aspx](https://msdn.microsoft.com/pt-br/library/windows/desktop/dn688964(v=vs.85).aspx)>. Acesso em: abr 2017.

NIELSEN, Jacob. **Usability Engineering**. Boston: Academic Press, 1993.

OLIBÁRIO, José Machado Neto. **Usabilidade da interface de dispositivos móveis: heurísticas e diretrizes para o design**. Dissertação (Mestrado-Programa de Pós-Graduação em Ciências da Computação e Matemática Computacional). Universidade de São Paulo, p. 136, 2013.

PREECE, Jennifer; ROGERS, Yvonne; SHARP, Helen. **Design de Interação: Além da interação homem-computador**. John Wiley e Sons. São Paulo, 2005.

PESSOA, I. N. S.; DE SOUSA, P. H. A. Q.; NUNES, R. F. **Interacting with Physically-Based Character Using Real-Time Motion Capture with Kinect Sensor**. 2016 XVIII Symposium on Virtual and Augmented Reality (SVR). Anais...IEEE, jun. 2016.

QUEIROZ, Marco Antônio de. **Tudo tem sua primeira vez – parte I**. 2015. Disponível em: <http://acessibilidadelegal.com/13-tudotem.php>. Acesso em: abr. 2017.

ROCHA, H. da; BARANAUSKAS, M. **Design e avaliação de interfaces humano-computador**. Unicamp, 2003. ISBN 9788588833043. Disponível em: <<https://books.google.com.br/books?id=McefAAAACAAJ>>.

ROGERS, Y.; SHARP, H.; PREECE, J.; TEPPER, M. Interaction design: beyond human-computer interaction. **netWorker: The Craft of Network Computing**, v. 11, n. 4, p. 34, 2007.

SAFFER, Dan. **Designing for Interaction**, Segunda Edição: Creating Innovative Applications and Devices. Berkeley, 2011.

SHACKEL, B. Usability - context, framework, definition, design and evaluation. **Interact. Comput.**, Elsevier Science Inc., New York, NY, USA, v. 21, n. 5-6, p. 339-346, dez. 2009. ISSN 0953-5438. Disponível em: <<http://dx.doi.org/10.1016/j.intcom.2009.04.007>>.

SHNEIDERMAN, Ben. **Designing the User Interface: Strategies for Effective Human-computer Interaction**. 4. Ed. Boston: Addison Wesley Longman, Inc., 2009.

SILVEIRA, D. T.; CÓRDOVA, F. P. **Métodos de Pesquisa**. Curso de Graduação Tecnológica – Planejamento e Gestão para o Desenvolvimento Rural da SEAD/UFRGS. – Porto Alegre: Editora da UFRGS, 2009.

SONZA, Andréa Poletto. **Ambientes virtuais acessíveis sob a perspectiva de usuário com limitação visual**. 2008. 313f. Tese (Doutorado em Informática na Educação) - Universidade Federal do Rio Grande do Sul, Porto Alegre, 2008. Disponível em: <http://www.lume.ufrgs.br/bitstream/handle/10183/14661/000666392.pdf?sequence=1> Acesso em: maio de 2017.

SOUZA, L. C. **Avaliação antropométrica através da delimitação da silhueta corporal utilizando o kinect**. [s.l.] CEULP/ULBRA, 2015.

TAINHA, Diego Dias. **Análise postural computadorizada para identificação de hiperlordose utilizando o sensor Kinect**. 2015. 11 f. Monografia (Especialização) - Curso de Ciência da Computação, Centro Universitário Luterano de Palmas, Palmas, 2015.

TREVISAN, DANIELA GORSKI ; COSTA, ROSA MARIA E. MOREIRA DA ; RIEDER, RAFAEL ; PINHO, M. S. **Tendências e Técnicas em Realidade Virtual e Aumentada**. Porto Alegre: [s.n.].

W3C, Brasil. Cartilha acessibilidade na Web [livro eletrônico]: fascículo 2 : **benefícios, legislação e diretrizes da acessibilidade na Web**. São Paulo: Comitê Gestor da Internet no Brasil, 2015.

WIGDOR, D.; WIXON, D. Brave NUI World: **Designing Natural User Interfaces for Touch and Gesture**. Brave NUI World Designing Natural User Interfaces for Touch and Gesture, p. 253, 2011.