

**UNIVERSIDADE PRESBITERIANA MACKENZIE
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA E COMPUTAÇÃO**

GUSTAVO SCALABRINI SAMPAIO

**DESENVOLVIMENTO DE UMA INTERFACE
COMPUTACIONAL NATURAL PARA PESSOAS COM
DEFICIÊNCIA MOTORA BASEADA EM VISÃO
COMPUTACIONAL**

São Paulo
2018

**UNIVERSIDADE PRESBITERIANA MACKENZIE
PROGRAMA DE PÓS-GRADUAÇÃO EM
ENGENHARIA ELÉTRICA E COMPUTAÇÃO**

GUSTAVO SCALABRINI SAMPAIO

**DESENVOLVIMENTO DE UMA INTERFACE
COMPUTACIONAL NATURAL PARA PESSOAS COM
DEFICIÊNCIA MOTORA BASEADA EM VISÃO
COMPUTACIONAL**

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Presbiteriana Mackenzie, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica e Computação.

ORIENTADOR: Prof. Dr. Maurício Marengoni.

São Paulo
2018

S192 Sampaio, Gustavo Scalabrini
Desenvolvimento de uma interface computacional natural para pessoas com deficiência motora baseada em visão computacional / Gustavo Scalabrini Sampaio.

107 f.: il.; 30 cm

Dissertação (Mestrado em Engenharia Elétrica e Computação) -
Universidade Presbiteriana Mackenzie, São Paulo, 2018.

Orientador: Maurício Marengoni.

Bibliografia: f. 81-89

1. Interação Humano-Computador. 2. Visão Computacional.
3. Rastreamento Facial. 4. Pontos Faciais. 5. Pessoas com deficiência.
I. Título.

CDD 004.1

AGRADECIMENTOS

Agradeço a todos os envolvidos nesse trabalho que, de forma ativa, possibilitaram sua elaboração.

Ao Prof. Dr. Maurício Marengoni que, de forma inteligente, me ajudou a estruturar o trabalho e sugeriu conteúdos que facilitariam seu entendimento.

Aos meus pais, Ademar e Maria Catarina, bem como meu irmão Heitor, que durante todo o processo de desenvolvimento do sistema deram valiosas sugestões e realizaram diversos testes parciais, sem os quais muitas funcionalidades não seriam desenvolvidas.

À Prof. Dra. Valeria Farinazzo Martins que me ajudou à elaborar o plano de testes do sistema e a organizar a realização desses testes.

Aos alunos do Programa de Pós-Graduação em Engenharia Elétrica e Computação e da Faculdade de Computação e Informática da Universidade Presbiteriana Mackenzie, que foram os usuários de teste do sistema desenvolvido.

À Universidade Presbiteriana Mackenzie e à CAPES pela concessão da bolsa de estudos, viabilizando minha participação no curso.

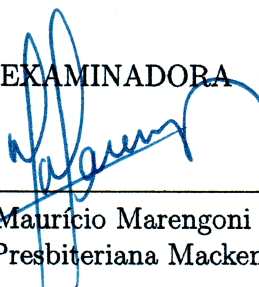
GUSTAVO SCALABRINI SAMPAIO

**DESENVOLVIMENTO DE UMA INTERFACE
COMPUTACIONAL NATURAL PARA PESSOAS COM
DEFICIÊNCIA MOTORA BASEADA EM VISÃO
COMPUTACIONAL**

Dissertação de Mestrado apresentado ao Programa de Pós-Graduação em Engenharia Elétrica e Computação da Universidade Presbiteriana Mackenzie, como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica e Computação

Aprovada em 8 de fevereiro de 2018

BANCA EXAMINADORA



Prof. Dr. Maurício Marengoni
Universidade Presbiteriana Mackenzie



Prof. Dr. Elizeu Coutinho de Macedo
Universidade Presbiteriana Mackenzie



Prof. Dr. Clodoaldo Aparecido de Moraes Lima
Universidade de São Paulo

RESUMO

O desenvolvimento de novas formas de interação com o computador, como telas *touchscreen* e óculos de realidade virtual, tornaram mais intuitiva e natural a sua utilização. O presente trabalho apresenta conceitos e técnicas utilizados para o desenvolvimento de um sistema que permite aos usuários utilizarem o computador pessoal de maneira simples e eficiente por meio dos movimentos da face. Foram utilizadas técnicas de visão computacional como segmentação, detecção facial e detecção dos pontos da face, bem como técnicas de programação e matemáticas para o desenvolvimento desse sistema. Trata-se de uma interface humano-computador do tipo natural que utiliza os movimentos da face, dos olhos e da boca para executar funções de controle como movimentação do cursor do mouse e acionamento de cliques e teclas. Diversos testes mostraram que o sistema desenvolvido apresentou desempenho superior em relação a sistemas similares, desempenho de processamento de 100 *fps*, *throughput* de 1,20 *bits/s* para movimentações com o cursor do mouse e 1,15 teclas acionadas pela face por segundo, é de fácil utilização, rápido aprendizado e pode ser utilizado em um grande número de aplicações, como navegação na internet, redes sociais e jogos. O sistema desenvolvido pode ser utilizado por pessoas com deficiências motoras como uma tecnologia assistiva, promovendo a inclusão social, além de proporcionar maiores oportunidades educacionais e profissionais para esse público.

Palavras-chave: *Interação Humano-Computador, Visão Computacional, Rastreamento Facial, Pontos Faciais, Pessoas com deficiência.*

ABSTRACT

New interaction devices such as touchscreens and virtual reality googles has made human computer interaction more intuitive and natural. This work presents concepts and techniques used for the development of a system that allows users to interact with personal computer in a simple and efficient way through face movements. Computer vision techniques such as segmentation, facial detection and landmarks detection were used, as well as programming and mathematical techniques for the development of this system. It is a natural human-computer interface that uses face, eye and mouth movements to perform control functions such as mouse cursor movement and triggering of clicks and keys. Several tests have shown that the developed system has superior performance compared to similar systems, 100 *fps* processing performance, throughput of 1.20 *bits/s* for mouse cursor movements, and 1.15 keystrokes with the face per second, it easy to use, fast learning and it can be used in a large number of applications such as internet browsing, social networking and games. The developed system can be used by people with motor disabilities as an assistive technology, promoting social inclusion, as well as providing greater educational and professional opportunities for this public.

Keywords: *Human-Computer Interaction, Computer Vision, Face Tracking, Face Landmarks, People with disabilities.*

Sumário

1	INTRODUÇÃO	1
2	REFERENCIAL TEÓRICO	5
2.1	Interação Humano-Computador	5
2.1.1	Visão geral	5
2.1.2	Interfaces Naturais	9
2.1.3	Interfaces computacionais para pessoas com deficiência motora	12
2.2	Visão Computacional	14
3	METODOLOGIA	20
4	DESENVOLVIMENTO DO SISTEMA	25
4.1	Pré-Processamentos	26
4.1.1	Conversão do domínio de cor	28
4.1.2	Ajuste de iluminação	29
4.1.3	Segmentação de pele	31
4.1.4	Filtragem de ruídos	33
4.1.5	Determinação de região de possível face	34
4.2	Detecção Facial	35
4.3	Detecção dos Pontos Faciais	42
4.4	Posicionamento do cursor do mouse	46
4.5	Acionamento de cliques e teclas	52
4.6	Tela de configuração e funções do sistema	53
5	TESTES COM USUÁRIOS	56
5.1	Teste de movimentação do cursor do mouse	58
5.2	Teste comparativo entre sistemas	64
5.3	Teste de escrita em teclado virtual	67
5.4	Teste de acionamentos com olhos e boca	68
5.5	Teste de acionamentos com a face	70
5.6	Pesquisa de satisfação de usabilidade	71
5.6.1	Perfil do Usuário	72
5.6.2	Tela de Configuração	72

5.6.3	Opções do Sistema	73
5.6.4	Funcionamento do Sistema	74
5.6.5	Facilidade de Uso e Aprendizado	75
5.6.6	Conforto de Utilização	75
5.6.7	Aplicação do Sistema Desenvolvido	76
5.6.8	Satisfação	77
6	CONCLUSÃO	79
	REFERÊNCIAS BIBLIOGRÁFICAS	81
	APÊNDICE	90
	Apêndice A – Questionário da Pesquisa de Satisfação	90

Lista de Figuras

1	Interface de <i>software</i> estilo WIMP	8
2	Diagrama do mecanismo de um sistema baseado em gestos	10
3	Exemplo de processamento de imagem	15
4	Exemplo de segmentação do plano de fundo em uma imagem	16
5	Exemplo de reconhecimento de um cachorro pelo IBM <i>Watson</i>	18
6	Fluxograma do sistema desenvolvido	25
7	Pré-Processamentos realizados pelo sistema desenvolvido	27
8	Conversão de domínio de cor. a) <i>Frame</i> de entrada	29
9	Ajuste de iluminação da imagem	30
10	Plano chroma Cr e Cb	32
11	Segmentação de cor de pele	32
12	Filtragem de ruídos da segmentação de pele por <i>Median Blur</i>	34
13	Definição e extração da região de possível face	34
14	<i>Kernels</i> para cálculo de características propostos por Viola e Jones	37
15	Exemplo de cálculo de imagem integral	38
16	Exemplo de cálculo das intensidades dos <i>pixels</i> de uma região	39
17	Características definidas pelo <i>Adaboost</i>	41
18	68 pontos faciais detectados	43
19	Exemplo de <i>Árvore de Regressão</i>	45
20	Exemplo de detecção dos pontos faciais	45
21	Pontos faciais e regiões utilizados no sistema desenvolvido	46
22	Área de movimentação de referência de eixo considerado pelo sistema	47
23	Curvas de Funções logarítmicas	48
24	Curva logarítmica natural e linear da correlação entre escala de distância e porcentagem de movimentação	50
25	Comparação de posicionamento do cursor do mouse utilizando ou não a função de transferência	51
26	Tela de configuração do sistema desenvolvido	54
27	Divisão dos testes em grupos de usuários	57
28	Modelo do ambiente de teste definido na norma ISO/TS 9241-411	58
29	Sentido de movimentação realizado pelo cursor do mouse	59
30	Resultados gráficos dos Testes ISO/TS 9241-411	61

31	Média dos valores de <i>throughput</i> dos ambientes de teste	63
32	Caminho percorrido pela cursor do mouse no Amb. 1	63
33	Teste comparativo entre sistemas	66
34	Teclado virtual utilizado no teste de escrita	67
35	Resultado do teste de escrita	68
36	Resultado do teste de acionamentos com olhos e boca	69
37	Ambiente de teste de acionamento de teclas com a face	70
38	Pesquisa de Satisfação - Perfil	72
39	Pesquisa de Satisfação - Experiência em Informática	73
40	Pesquisa de Satisfação - Tela de Configuração	73
41	Pesquisa de Satisfação - Opções do Sistema	74
42	Pesquisa de Satisfação - Funcionamento do Sistema	75
43	Pesquisa de Satisfação - Facilidade de Uso e Aprendizado	76
44	Pesquisa de Satisfação - Conforto de Utilização	76
45	Pesquisa de Satisfação - Aplicação do Sistema Desenvolvido	77
46	Pesquisa de Satisfação - Visão Geral	78

Lista de Tabelas

1	Ambientes utilizados no teste de eficiência de movimentação do cursor do mouse	60
2	Resultados numéricos do teste ISO/TS 9241-411	62
3	Resultado do teste de escrita em teclado virtual	68
4	Resultado do teste de acionamentos com olhos e boca	69

1 INTRODUÇÃO

Os recursos e dispositivos computacionais e os meios de comunicação oferecem a seus usuários cada vez mais funcionalidades e formas alternativas de usabilidade. O computador pessoal que inicialmente era utilizado para realizar cálculos, hoje se apresenta como um recurso de entretenimento, ferramenta de aprendizado e de trabalho, consumo de informação e comunicação. O desenvolvimento dessas funcionalidades foram acompanhadas, também, pelo desenvolvimento de novas formas de interação com os usuários, telas *touchscreen* e óculos de realidade virtual por exemplo, já fazem parte dos mecanismos de interação humano-computador(IHC). Apesar dos avanços das formas de interação com o computador para a maioria dos usuários, as pessoas com deficiência motora ainda encontram dificuldade de utilizar o computador, excluindo-as das interações sociais já comuns no cotidiano dos outros usuários e limitando seus meios de desenvolvimento acadêmico e profissional (ITS, 2008; BARBOSA, 2010).

No Brasil, segundo dados do IBGE (Instituto Brasileiro de Geografia e Estatística), em 2000 a porcentagem da população de pessoas com algum tipo de deficiência era de 14,4%, sendo 5,5% com deficiências motoras. Em 2010 esse número cresceu para 23,9% de pessoas com algum tipo de deficiência, 6,95% com deficiências motoras, representando cerca de 13 milhões de brasileiros que apresentam dificuldade não só de locomoção, mas também ao acesso a recursos de entretenimento e comunicação (IBGE, 2003; IBGE, 2012). Esses números mostram como essa parcela da população vem crescendo, exigindo que melhorias sejam direcionadas a todos os setores tendo em vista as reais necessidades desse público.

O Instituto de Tecnologia Social ITS (2008) mostrou a importância do desenvolvimento de ferramentas computacionais que atendam as necessidades das pessoas com deficiência. Essas pessoas tendo livre e independente acesso aos recursos computacionais, seja para realizar tarefas básicas como produzir documentos e assistir vídeos, ou usar a internet para acessar redes sociais e interagir com outros usuários, faz com que essas pessoas sejam integradas à sociedade digital e se sintam iguais aos outros usuários desse meio, proporcionando o aumento de sua autoestima. O acesso à recursos computacionais, possibilita também que essas pessoas possam se desenvolver de forma mais eficiente acadêmica e profissionalmente, por meio do ensino à distância e treinamento de ferramentas profissionais, permitindo melhorar sua qualificação e facilitar sua entrada no mercado

de trabalho.

O presente trabalho propõe um sistema, desenvolvido com técnicas de visão computacional, como segmentação, detecção facial e detecção dos pontos da face, bem como conversões entre domínios de posicionamento, que possibilita o controle do cursor do mouse e do acionamento de botões e teclas do computador por meio dos movimentos faciais e de ações como a abertura e fechamento de olhos e boca. Tais funcionalidades permitem utilizar os principais recursos do sistema operacional *Windows* e aplicações como navegadores de internet e jogos, de forma independente, eficiente e de maneira simples. Tecnicamente, o sistema pode ser classificado como uma interface de interação humano-computador do tipo natural, ou seja, uma interface que não exige equipamentos físicos para realizar a interação com o computador. Tendo em vista as necessidades das pessoas com deficiência motora, o sistema pode ser utilizado como uma tecnologia assistiva e proporcionar o acesso dessas pessoas aos recursos computacionais.

As técnicas utilizadas para o desenvolvimento do sistema são baseadas nas técnicas de melhor eficiência analisadas nos diversos trabalhos que compõem o estado da arte das interfaces computacionais para pessoas com deficiência motora, introduzindo novas abordagens e conceitos, permitindo o desenvolvimento de um sistema com maior desempenho e precisão.

A partir da análise de trabalhos que tratam do desenvolvimento de interfaces computacionais para pessoas com deficiência motora, especificamente os baseados em movimentação facial, foi possível identificar as características principais desse tipo de sistema e as funcionalidades que devem fazer parte do sistema proposto. Basicamente, esses sistemas são compostos pela detecção da face do usuário, da conversão da movimentação da face para as coordenadas de posicionamento do cursor do mouse, e da simulação dos cliques a partir da movimentação dos elementos faciais, como olhos e boca. O grande desafio no desenvolvimento desse tipo de sistema é garantir o desempenho quanto ao tempo de processamento e precisão. Muitos dos trabalhos analisados relataram baixo tempo de processamento, fator que compromete a precisão do sistema tornando-o muitas vezes inutilizável para atividades reais. Essa falta de desempenho, causada por excesso de processamentos desnecessários ou técnicas aplicadas de maneira incorreta, não permite que o sistema interprete todos os movimentos do usuário, movimentos do cursor do mouse po-

dem apresentar comportamentos de travamento e imprecisão e o acionamento de cliques e teclas podem não acontecer da maneira esperada.

Trabalhos como os de Tu, Huang e Tao (2005), Fu e Huang (2007), Xu, Wang e Feng (2009), Pallejà et al. (2011) e Kraichan e Pumrin (2014) utilizaram para detecção facial um algoritmo em cascata de classificadores fracos, classificadores estes encontrados a partir do treinamento por Adaboost, técnica proposta por Viola e Jones (2001), por garantir precisão e rapidez na execução dessa tarefa. Outros trabalhos que estudaram técnicas eficazes de realizar a detecção facial, como os de Ji, Lu e Xu (2014) e Gor e Bhatt (2015), apontaram que a detecção facial por *Adaboost/Haar Cascade* aliada a redução da região de busca por segmentação do tom de pele, constitui uma forma rápida e precisa de realizar essa tarefa, atendendo a necessidade de sistemas que são dependentes de desempenho. Trabalhos como os de Ji, Lu e Xu (2014) e Gor e Bhatt (2015), bem como o de Hjelmås e Low (2001), apresentaram técnicas de detecção facial no desenvolvimento dos sistemas baseados em visão computacional, facilitando a compreensão do tema.

Diferentemente das técnicas de detecção facial, as técnicas de conversão da movimentação facial para o posicionamento do cursor do mouse são distintas entre os trabalhos, e ainda não foram determinadas técnicas mais eficientes ou mais utilizadas para realizar essa função, assim este problema ainda está em aberto na literatura. Fu e Huang (2007) fizeram a estimativa angular do posicionamento da face e utilizaram os ângulos encontrados para realizar a movimentação do cursor. Pallejà et al. (2011) utilizaram o centro da face detectada como ponto de referência em uma área fixa pré-determinada. Já Naizhong, Jing e Jun (2015) utilizaram o posicionamento da boca no decorrer de um intervalo de *frames* em relação ao centro da face.

Embora a conversão de posições possa parecer uma tarefa simples, não basta apenas converter a movimentação facial em movimentação do cursor do mouse, é preciso aplicar uma função de transferência para tornar o sistema viável, melhorando a precisão do posicionamento do cursor. Kjeldsen (2006) e Pallejà et al. (2011) por exemplo, propuseram funções de transferência para o posicionamento do cursor do mouse, mostrando o motivo da função de transferência ser um requisito importante, uma vez que permite a movimentação do cursor de forma suave e precisa.

Outro elemento dos sistemas de interface computacional baseados em movimentação

facial é o mecanismo de simulação dos cliques. De maneira geral, essa tarefa é realizada de duas formas diferentes, a primeira utiliza o modo *dwell* que executa o clique após a permanência do cursor em uma determinada região por um determinado tempo, Betke, Gips e Fleming (2002) e Kjeldsen (2006) utilizaram esse tipo de mecanismo. A segunda forma utiliza os estados de elementos faciais, como abertura e fechamento de olhos e boca, como pode ser visto nos trabalhos de Kraichan e Pumrin (2014) e Naizhong, Jing e Jun (2015).

Nos próximos capítulos serão apresentados os conceitos teóricos e os métodos utilizados para o desenvolvimento do sistema, bem como os diversos testes com usuários, realizados para medir a eficiência do sistema e demonstrar sua possibilidade de utilização em atividades cotidianas.

2 REFERENCIAL TEÓRICO

2.1 Interação Humano-Computador

2.1.1 Visão geral

Os computadores são utilizados pelo homem em diferentes tarefas de seu cotidiano, seja em atividades profissionais ou como meio de comunicação e entretenimento. A interação entre o homem e o computador pode acontecer de diferentes maneiras, no que diz respeito aos meios físicos e aos recursos de *software*, e sua forma é fator determinante para a correta utilização dos recursos computacionais e pelo bem estar do usuário.

A interação humano-computador é caracterizada por sua multidisciplinaridade, uma vez que envolve estudos voltados aos computadores, como ciência e engenharia da computação, e voltados ao homem como psicologia, cognição e ergonomia, e tem como objetivo traçar diretrizes e boas práticas para o desenvolvimento, implementação e avaliação de sistemas que de alguma forma interagem com humanos (DIX et al., 2003; ROCHA; BARANAUSKAS, 2003).

O homem é o elemento principal da IHC. Conhecer suas limitações e capacidades é fundamental para direcionar os esforços e procedimentos durante o desenvolvimento de sistemas que atendam suas necessidades. Basicamente, deve-se entender como o homem processa as informações advindas do sistema e como ele se expressa durante a interação tendo em vista sua individualidade, moldada pelo sexo, capacidades intelectuais e físicas, idade, experiências passadas, entre outras. A psicologia auxilia nesse processo, buscando entender o comportamento humano e os processos mentais relacionados a atenção, memória, resolução de problemas e aprendizado, já a ergonomia direciona o desenvolvimento de ferramentas e artefatos tendo em vista o ambiente e o público alvo dos sistemas, garantindo maior conforto, segurança e eficiência de utilização.

O desenvolvimento de sistemas interativos deve levar em conta diferentes fatores humanos, uma vez que emoções e características humanas alteram a forma com que os indivíduos executam suas ações. Emoções negativas pioram, de maneira geral, as capacidades do homem que sob estresse pode passar a não conseguir utilizar determinado tipo de interface, ou ao se deparar com sistemas de difícil utilização pode ter mudanças de

humor, podendo até abandonar o uso desses sistemas. Assim a interação deve acontecer sempre da forma mais intuitiva possível e deve possuir elementos que realmente auxiliem o usuário a atingir seu objetivo. O excesso ou a falta de informações e recursos em um sistema pode ocasionar uma experiência ruim de utilização, muita informação exige maior concentração e demanda uma maior capacidade de memória e velocidade de aprendizado, sua falta pode levar o usuário a ter dúvidas sobre o que é apresentado e do que deve servir de entrada para o sistema (DIX et al., 2003; ROCHA; BARANAUSKAS, 2003).

Do ponto de vista do computador, a IHC trata dos recursos que possibilitam aos diversos dispositivos computacionais trocarem informações com os usuários. Mais do que simplesmente executar processamentos, os sistemas interativos recebem e transmitem informações. Assim, a forma pela qual a troca de informações é realizada impacta diretamente na relação entre homem e computador. O estudo e desenvolvimento de recursos computacionais são essenciais e úteis do ponto de vista técnico da IHC, já que o efetivo trabalho de desenvolvimento é realizado nessa vertente. Entender e aplicar novos conceitos nos sistemas é uma necessidade inerente ao desenvolvimento das tecnologias. Sistemas desenvolvidos com técnicas de inteligência artificial, aprendizado de máquina e visão computacional, estabeleceram novas formas de IHC e possibilitaram uma interação mais abrangente. O IBM *Watson* é um exemplo de sistema que utiliza essas abordagens, sendo capaz de interagir de forma natural com o usuário por meio de perguntas e respostas, de "aprender" com as informações trocadas e realizar reconhecimentos de imagens (DIX et al., 2003; ROCHA; BARANAUSKAS, 2003; IBM, 2017).

A interação entre homem e computador se da por meio da interface de usuário, que pode ser descrita pela composição de 3 elementos fundamentais: os dispositivos de entrada e saída o modelo conceitual e as técnicas de interação do sistema. Os dispositivos de entrada e saída são os meios físicos da interação, tipicamente em computadores pessoais temos como dispositivos de entrada o mouse, o teclado e, ainda embrionário, os sistemas de reconhecimento de voz e como dispositivos de saída o monitor, caixas de som e impressoras 2D ou 3D. O modelo conceitual de um sistema se refere à visão que o usuário tem sobre as funcionalidades do sistema, como ele funciona e como as respostas são apresentadas aos usuários. Já as técnicas de interação correspondem aos recursos utilizados por dispositivos de entrada e saída, modelos conceituais e elementos de *software* que constituem a forma com que o usuário pode interagir com o sistema para realizar a tarefa desejada (DIX et

al., 2003; BARBOSA, 2010; JACKO, 2012).

A utilização da IHC no desenvolvimento de sistemas interativos busca garantir que determinadas características de interação e interface estejam presentes, permitindo aos usuários utilizarem esses sistemas de maneira eficiente aproveitando todos os recursos disponíveis. Para Barbosa (2010) alguns critérios de qualidade direcionam o desenvolvimento de características que tornam a interface e a interação adequadas para os usuários, são eles a usabilidade, a experiência do usuário, a acessibilidade e a comunicabilidade. A usabilidade diz respeito à facilidade de aprendizado e o uso do sistema. A experiência do usuário está relacionada com os sentimentos, emoções e sensações decorrentes da utilização do sistema, em sistemas de *videogame* por exemplo, o usuário geralmente experimenta prazer, diversão e desafio. A acessibilidade está relacionada com a flexibilidade do sistema de permitir seu uso por usuários com diferentes níveis de capacidades físicas e intelectuais, não impondo obstáculos em sua interface. Por sua vez, comunicabilidade se refere a capacidade do sistema de comunicar ao usuário as intenções e a lógica do *design*, uma vez que a compreensão de como o sistema funciona faz com que sua utilização seja mais eficiente, um exemplo desse conceito é o sistema transmitir ao usuário dicas de algumas funcionalidades no momento em que esse usuário posicionar o cursor do mouse em um elemento da tela ou durante o carregamento de algum recurso ou página.

Apesar de IHC e Engenharia de *Software* serem relacionados, por fazer parte do desenvolvimento de sistemas, temos que a engenharia de *software* é focada no sistema, enquanto a IHC é focada nos usuários. Da mesma forma que a engenharia de *software* utiliza metodologias para realizar seus desenvolvimentos, a IHC também possui métodos, procedimentos e ferramentas para realizar a construção de *design* de interfaces. De forma geral, o processo de *design* inclui atividades realizadas para analisar o cenário do sistema, que envolve a análise dos usuários e das tarefas e descrição dos requisitos de usabilidade; *design* das interfaces do sistema, que envolve modelagens e prototipações; *design* conceitual e de interação; e avaliação das interfaces (BARBOSA, 2010). No que diz respeito aos estilos de interfaces de aplicações de *software*, o estilo de interface predominante hoje na interação humano-computador é o WIMP (*Windows Icons Menus and Pointers*), alguns outros estilos são (DIX et al., 2003):

- Interface de Linha de Comando



Figura 1: Interface de *software* estilo WIMP

Fonte: (GRFOOD, 2016)

- Menus
- Linguagem Natural
- Diálogos de Perguntas e Respostas
- Formulários e Planilhas
- *Point and Click*
- Interfaces Tridimensionais

O estilo WIMP, utilizado nos sistemas operacionais como *Windows* e *MacOS*, permite aos usuários terem, de forma interativa e transparente, acesso às opções do sistema e aos aplicativos, através de janelas, menus e ícones, facilitando o uso por pessoas com qualquer nível de conhecimento computacional. Esse estilo de interface permite o uso de diferentes interfaces físicas, por exemplo, uma determinada aplicação pode ser utilizada por meio do mouse e por meio de telas *touchscreen*. A figura 1 apresenta uma interface de *software* baseada no estilo WIMP, que pode ser utilizada tanto via *touchscreen* quanto por mouse.

As interfaces de *software* são muito flexíveis, onde vários formatos podem ser criados e misturados, já as interfaces de *hardware* são restritas ao desenvolvimento tecnológico, entretanto novas interfaces de *hardware* foram desenvolvidas recentemente como óculos de realidade virtual, e interfaces naturais como *touchscreen* e baseadas em gestos como o *Kinect* da *Microsoft*.

2.1.2 Interfaces Naturais

As interfaces naturais estão presentes em inúmeros dispositivos de uso cotidiano, seja em *smartphones* ou em *videogames*, proporcionando aos usuários uma forma mais interativa e intuitiva de utilização dos sistemas.

Blake (2011) define interfaces naturais de usuário como uma interface desenvolvida para utilizar habilidades já adquiridas pelo usuário, essas habilidades são oriundas da interação humano-humano e envolve a fala, o toque e gestos, proporcionando uma forma intuitiva de utilização dos sistemas. Jain, Lund e Wixon (2011) aprofundaram esse conceito por definir interfaces naturais como uma interface que possui interação semelhante a nossa interação com o mundo, ou seja, o usuário não percebe a tecnologia no momento da utilização. Wigdor e Wixon (2011) traçaram um panorama dos objetivos das interfaces naturais com foco na experiência que o usuário deve ter ao utilizar esse tipo de interface:

- O usuário deve entender a interface como a extensão do seu corpo.
- A experiência de utilização deve ser a mesma para novos usuários e para usuários experientes.
- A experiência de utilização deve ser autêntica, não imitando o mundo real.
- A interface deve ser condizente com o contexto de utilização, e deve considerar as indicações visuais, a resposta do sistema e as entradas e saídas.
- A interface não deve ser baseada nos paradigmas de interface já existentes.

A obra de Saffer (2008) trata de interfaces naturais baseadas em gestos, considerando sistemas que podem captar movimentos físicos e responder a esses movimentos sem o auxílio de equipamentos tradicionais, como mouse e canetas *stylus*, assim movimentos das mãos, cabeça, toques e até movimentos faciais são considerados gestos. Os dispositivos que utilizam interfaces naturais podem variar bastante em forma, tamanho e sensibilidade,

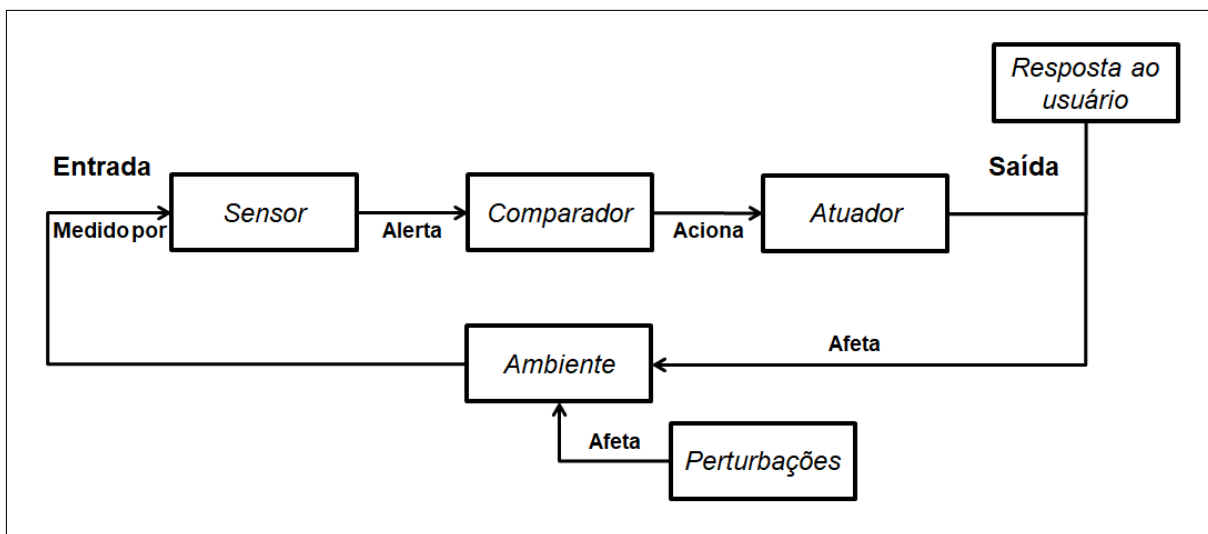


Figura 2: Diagrama do mecanismo de um sistema baseado em gestos

Fonte: Adaptado. (SAFFER, 2008)

entretanto todos possuem 3 elementos básicos, um sensor, um comparador e um atuador. A figura 2 apresenta o diagrama do mecanismo de um sistema baseado em gestos.

O sensor, ou o conjunto de sensores, é utilizado para detectar as mudanças que ocorrem no ambiente. O tipo do sensor define quais interações podem ser realizadas, delimitando o sistema. Tipicamente sistemas com funcionalidade *touchscreen* utilizam sensores de contato, sistemas de detecção de presença utilizam sensores infravermelho, já sistemas baseados em imagens utilizam câmeras para captar a movimentação do usuário. A calibração desses sensores é importante para viabilizar a utilização efetiva do sistema, sensores com sensibilidade elevada podem captar sinais falsos ou muito ruído, já sensores pouco sensíveis podem passar a percepção de mal funcionamento do sistema ao usuário. Os sinais obtidos por meio dos sensores é analisado pelo comparador do sistema, esse componente compara o estado atual do sensor com um valor desejado ou com o valor anterior, definido assim a ação que o atuador deve realizar. O atuador, por sua vez, determina o que o sistema deve fazer, por exemplo, em sistemas *touchscreen* o atuador pode executar um comando após o usuário pressionar a posição em tela correspondente a um botão.

Saffer (2008) enfatizou a ideia de que o *design* de interfaces baseadas em gestos segue o mesmo conceito de desenvolvimento que outros tipos de interfaces, uma vez que interfaces bem sucedidas devem atender as necessidades dos usuários e do ambiente em que o sistema estará inserido. Em alguns casos a utilização de interfaces baseadas em gestos pode

não beneficiar o usuário ou tornar a realização da tarefa desejada ineficiente, o autor cita um exemplo de sistema com o recurso de *touchscreen* em um ambiente em que o usuário pode estar usando luvas, nesse contexto o recurso de *touchscreen* não funcionaria corretamente. Por outro lado os benefícios de interfaces naturais, além de uma interação mais intuitiva com os usuários comparado às interfaces tradicionais, incluem dispositivos mais compactos, flexibilidade de aplicações e maior acessibilidade. Ainda para Saffer (2008), sistemas interativos baseados em gestos devem possuir algumas características para que sejam considerados de qualidade, entre outras:

- **Identificável:** Pode ser considerado como o maior problema na concepção de um sistema de interface natural e representa a facilidade com que o sistema é identificado pelo usuário como sendo de interface natural, por exemplo, uma tela *touchscreen* pode ser considerada comum, sem a funcionalidade *touch*, pelo usuário caso não tenha nenhuma indicação dessa funcionalidade.
- **Confiável:** O usuário deve se sentir seguro ao utilizar interfaces naturais, tanto em termos de segurança dos dados que envolve sua privacidade, ou de fragilidade do dispositivo.
- **Responsivo:** O sistema deve ter a capacidade de indicar ao usuário o momento em que um comando é recebido pelo sistema, indicações sonoras, visuais ou mesmo mecânicas precisam ser transmitidas ao usuário. A falta dessa característica pode levar o usuário a realizar a mesma ação diversas vezes levando à execução de ações indesejadas, como por exemplo a compra de um item diversas vezes em um site de compras.
- **Apropriado ao contexto:** As interfaces baseadas em gestos, principalmente, devem ser apropriadas à cultura dos usuários e do ambiente em que estará inserido. Os gestos que são apropriados a uma cultura pode não ser em outra, movimentos espalhafatosos em público pode levar o sistema a não ser utilizado nesses ambientes.
- **Significativo:** O sistema deve atender as necessidades do usuário.
- **Inteligente:** O sistema deve realizar tarefas que seus usuários não realizariam de forma simples, grandes quantidades de processamento de forma rápida, grande capacidade de armazenamento de informação, detecção de padrões complicados, etc.
- **Adaptativo:** O sistema deve ser capaz de se adaptar a todos os seus usuários e considerar que algumas ações, mesmo não sendo a indicada pelo sistema, podem

representar comandos, por exemplo em uma tela *touchscreen* uma ação esperada de arrastar um elemento em linha reta pode não ser realizado em linha reta por todos os usuários, entretanto o sistema deve concluir a ação caso ela esteja dentro de um determinado limite de usabilidade.

A característica dos sistema de interface natural de permitir ao usuário realizar tarefas sem a utilização de dispositivos intermediários, possibilita a criação de sistemas altamente acessíveis do ponto de vista das capacidades físicas desses usuários. As pessoas com deficiências motoras podem ser beneficiadas por esses sistemas, já que esses podem ser desenvolvidos baseados nos gestos que esse público pode realizar.

2.1.3 Interfaces computacionais para pessoas com deficiência motora

O desenvolvimento de sistemas interativos, como foi apresentado nas seções anteriores, tem como elemento central do desenvolvimento os usuários e o ambiente de utilização. Esse princípio é essencial em sistemas voltados às pessoas com deficiência. O desenvolvimento de sistemas que visam a acessibilidade de pessoas com deficiência é bem mais complexo em comparação com os sistemas tradicionais, isso se deve aos tipos de deficiência e ao nível de gravidade que cada tipo pode apresentar, exigindo o desenvolvimento de sistemas que atendam cada um desses cenários. Sistemas desenvolvidos às pessoas com deficiência visual, por exemplo, utilizam recursos de áudio e tato para se comunicar com o usuário, como o reconhecimento de voz e teclados em braille. Jacko (2012) apresentou a importância das interfaces perceptivas, naturais e adaptativas no desenvolvimento de sistemas para pessoas com deficiência, bem como alguns recursos que podem fazer parte de interfaces para pessoas com deficiência visual como o *display* em braille, já Zhang e Ye (2016) apresentaram um sistema capaz de guiar um deficiente visual em um ambiente fechado por meio de mecanismos de localização e comandos de voz.

Da mesma forma que as pessoas com deficiência visual, as pessoas com deficiência motora também encontram dificuldades para utilizar recursos computacionais, entretanto possuem características diferentes e devem ser concebidos focados no público com limitações de movimentação. Algumas aplicações como permitir a movimentação de cadeiras de rodas e utilização de computadores pessoais, vem sendo estudadas e desenvolvidas melhorando a qualidade de vida desses usuários.

A visão computacional tem papel importante no desenvolvimento desses sistemas, uma vez que as entradas não podem ser realizadas por meios tradicionais, como teclado e mouse, nem por alguns tipos de interfaces naturais como *touchscreen*. Interfaces naturais baseadas em movimentação facial podem ser utilizadas para o desenvolvimento de sistemas voltados às pessoas com deficiência motora, atendendo usuários de diversos níveis de gravidade, até por tetraplégicos desde que tenham pelo menos movimentos da cabeça e da face.

Trabalhos como os de Ballard e Stockman (1992), Betke, Gips e Fleming (2002), Tu, Huang e Tao (2005), Kjeldsen (2006), Fu e Huang (2007), Xu, Wang e Feng (2009), Pallejà et al. (2011), Kraichan e Pumrin (2014), Naizhong, Jing e Jun (2015), Bian et al. (2016) apresentaram sistemas que possibilitam a utilização de computadores pessoais por meio da movimentação facial do usuário, onde o controle do cursor do mouse e seus cliques são realizados por meio da face e dos elementos faciais como olhos e boca. A principal técnica envolvida nesse processo é a detecção facial, não só por permitir que a posição da face e dos seus elementos sejam encontrados na imagem de entrada e sejam utilizados para controlar recursos do sistema operacional, mas também por ser uma operação que exige muito processamento o que pode levar o sistema a apresentar baixo desempenho tornando-o inutilizável. A técnica de detecção facial deve ser implementada com técnicas auxiliares que preservem o desempenho. Alguns desses sistemas conseguem mesmo que no limiar executarem seus processamentos em um tempo tal que possibilite sua utilização em atividades reais, como o *CameraMouse* desenvolvido por Betke, Gips e Fleming (2002) e o *HeadMouse* desenvolvido por Pallejà et al. (2011).

Outros exemplos de sistemas voltados às pessoas com deficiência motora podem ser vistos nos trabalhos de Takahashi et al. (2001), que propuseram um braço robótico para servir comida à pessoas com deficiência, e Sato, Yoda e Inoue (2009), que demonstraram a aplicação do rastreamento dos movimentos corporais para controlar uma cadeira de rodas elétrica.

A análise desses trabalhos mostra a importância de delimitar o sistema com funcionalidades e modos de entrada específicos para o tipo de deficiência do usuário. A visão computacional possibilitou que sistemas voltados para pessoas com deficiência motora não necessitem de instrumentos vestindo o usuário, trazendo maior conforto de utilização, bem

como flexibilidade no desenvolvimento das funções do sistema. O grande desafio desses sistemas é o de garantir o desempenho, uma vez que as técnicas de visão computacional, se não configuradas adequadamente, podem exigir uma grande capacidade de processamento. O baixo desempenho desses sistemas muitas vezes torna sua utilização inviável, ou então ineficiente, não cumprindo com o objetivo proposto. Assim, a implementação de técnicas de aumento de desempenho são necessárias no desenvolvimento dos sistemas que utilizam visão computacional. Kraichan e Pumrin (2014), por exemplo, apresentaram uma forma de reduzir o tempo de detecção facial guardando a posição da face da última detecção, a atualização dessa posição acontece em intervalos regulares, entre esses intervalos a detecção acontece de forma muito mais rápida pois a área de busca é menor.

2.2 Visão Computacional

A visão humana é capaz de identificar, de maneira natural, as características de pessoas e objetos, interpretando de forma precisa os elementos observados, como cores, formas, luminosidade, etc., podendo ainda realizar classificações desses elementos. Szeliski (2010) define a visão computacional como a área do conhecimento que busca reproduzir a visão humana quanto a capacidade de extrair das imagens suas características, permitindo interpretar e descrever essas imagens e os elementos nelas presentes.

Atingir o objetivo da visão computacional não é uma tarefa fácil, principalmente quando a imagem possui um grande número de elementos. Grande parte das aplicações que utilizam visão computacional necessitam realizar diversos pré-processamentos para tratar a imagem de entrada, para então extrair ou identificar os elementos desejados por meio de algoritmos teorizados e desenvolvidos por estudiosos da área (PRINCE, 2012).

A visão computacional é composta por subáreas, cada qual com seu grupo de técnicas, utilizadas por sistemas cuja entrada seja uma imagem, um conjunto de imagens ou vídeo. Dentre outras subáreas, temos (SZELISKI, 2010):

- Processamento de Imagem
- Segmentação
- Detecção
- Reconhecimento



Figura 3: Exemplo de processamento de imagem. a) Imagem de entrada; b) Correção de iluminação na imagem de entrada.

O processamento de imagem é realizado na maioria dos casos durante a fase de pré-processamento de um sistema de visão computacional, ou seja, antes de quaisquer outras técnicas serem utilizadas, com o objetivo de preparar e corrigir a imagem quanto ao posicionamento, cores, brilho, contraste, ruídos, entre outros. Krig (2014) exemplifica a utilização de correção de iluminação como pré-processamento de um banco de imagens em um processo de treinamento, definindo um contraste comum em todas as imagens para padronizar o resultado do treinamento, garantindo assim um melhor resultado durante uma detecção ou correspondência de objetos. A figura 3 apresenta um exemplo de processamento de uma imagem, onde é aplicada uma correção de iluminação.

A segmentação de elementos em uma imagem é caracterizada pelo processo de agrupar os *pixels* (px) que possuem características semelhantes, criando diversos segmentos dos mesmo. Essa tarefa é realizada para que as análises e operações possam ser executadas de formas diferentes sobre esses grupos. O processo de segmentação pode ser realizado após, por exemplo, a detecção de bordas e extração do elemento desejado da imagem. Um exemplo de utilização de segmentação pode ser observado no trabalho de Shrivastava e Matlani (2016), onde a detecção de fumaça provinda de incêndio é realizada por meio da segmentação por cor. A figura 4 apresenta um exemplo da operação de segmentação em uma imagem com o objetivo de remover o plano de fundo, destacando o elemento principal.

A tarefa de detectar elementos em uma imagem é muito utilizada nas aplicações de



Figura 4: Exemplo de segmentação do plano de fundo em uma imagem. a) Imagem de entrada; b) Máscara de segmentação; c) Imagem com o plano de fundo removido.

visão computacional. Os algoritmos de detecção de bordas, de pontos característicos, de face, entre outros, são utilizados para localizar e delimitar o elemento foco de uma análise ou extração, para então realizar as operações desejadas sobre o ponto ou área detectada. O trabalho de Atadjanov e Lee (2015), por exemplo, utilizou a detecção de bordas e de pontos característicos para realizar a análise de simetria em uma imagem.

Dentre as subáreas de detecção em visão computacional, a detecção de faces é uma subárea fundamental, sem a qual tarefas que envolvam análises faciais não seriam possíveis de serem realizadas. Dentre as análises faciais derivadas da detecção facial temos: o reconhecimento facial, onde a face detectada é comparada com uma base de dados para determinar sua presença ou não na base; rastreamento facial, que informa de forma contínua a localização da face em vídeos; reconhecimento de expressões faciais, utilizado para detectar um estado emocional, como alegria, tristeza e medo, entre outras. Tecnicamente, a detecção facial de forma computacional é realizada por um sistema capaz de apontar a localização e o tamanho de uma ou mais faces presentes em uma imagem qualquer, entretanto não é uma tarefa fácil de ser realizada devido ao grande número de variações que podem existir em posição, localização, escala, expressões faciais, condições de iluminação, oclusão, etc. (HJELMÅS; LOW, 2001; YANG; KRIEGMAN; AHUJA, 2002; ZHANG; ZHANG, 2010).

Embora seja considerada uma tarefa difícil, a detecção facial foi alvo de muitas pesquisas e desenvolvimentos, e com a crescente melhoria dos recursos computacionais foi possível incorporá-la em dispositivos como celulares e sistemas embarcados.

O reconhecimento é uma das tarefas mais difíceis de ser implementada em visão computacional, essa dificuldade provém das infinidades de objetos possíveis no mundo real, que variam em formas, cores, e posição na imagem analisada. As técnicas de reconhecimento necessitam antes de realizar suas devidas classificações detectar o objeto desejado a partir das técnicas de detecção. Essa classificação pode ocorrer a partir de uma rede neural, técnicas de *boosting*, comparação com bases de imagens, entre outras.

Szeliski (2010) cita um exemplo de uso de reconhecimento facial em sistemas de verificação de identidade, após a face ser detectada a mesma é comparada com faces cadastradas em um banco de imagens, caso corresponda com uma das faces dessa base o usuário é identificado. Em 2016 a IBM implementou no sistema *Watson*, que é um sistema cognitivo para interação com pessoas. Dentre outras funcionalidades, esse sistema disponibiliza a funcionalidade de reconhecimento de imagens, onde o usuário pode treinar o *Watson* com imagens positivas e negativas de um determinado objeto, animal ou pessoas, e então realizar o reconhecimento da imagem de entrada, a figura 5 apresenta um exemplo de reconhecimento de uma imagem, no caso um cachorro, por meio do IBM *Watson* (IBM, 2017).

A visão computacional já está presente em aplicações de diversas áreas, melhorando a qualidade de vida dos usuários e viabilizando a execução automática de atividades antes realizadas de forma manual.

Na medicina, exames de imagens e procedimentos cirúrgicos são realizados com o auxílio da visão computacional. El-Bialy (2008) demonstrou a utilização de computação gráfica, visão computacional, reconhecimento de padrões, entre outras técnicas, em sistemas de reconstrução 3D de modelos reais de arcadas dentárias e maxilar a fim de auxiliar no tratamento dentário por meio de análises automáticas, simulações de tratamentos e planejamento de procedimentos.

Na área de acessibilidade, a visão computacional auxilia em atividades do dia-a-dia. Benjamim et al. (2012) propuseram um sistema baseado em visão computacional que possibilita pessoas cegas identificar embalagens de medicamentos.

Na indústria, em operações de análise de qualidade e detecção também já se utilizam sistemas baseados em visão computacional. Essa forma de utilização de visão computacional pode ser observado no trabalho de Semeniuta, Dransfeld e Falkman (2016) que

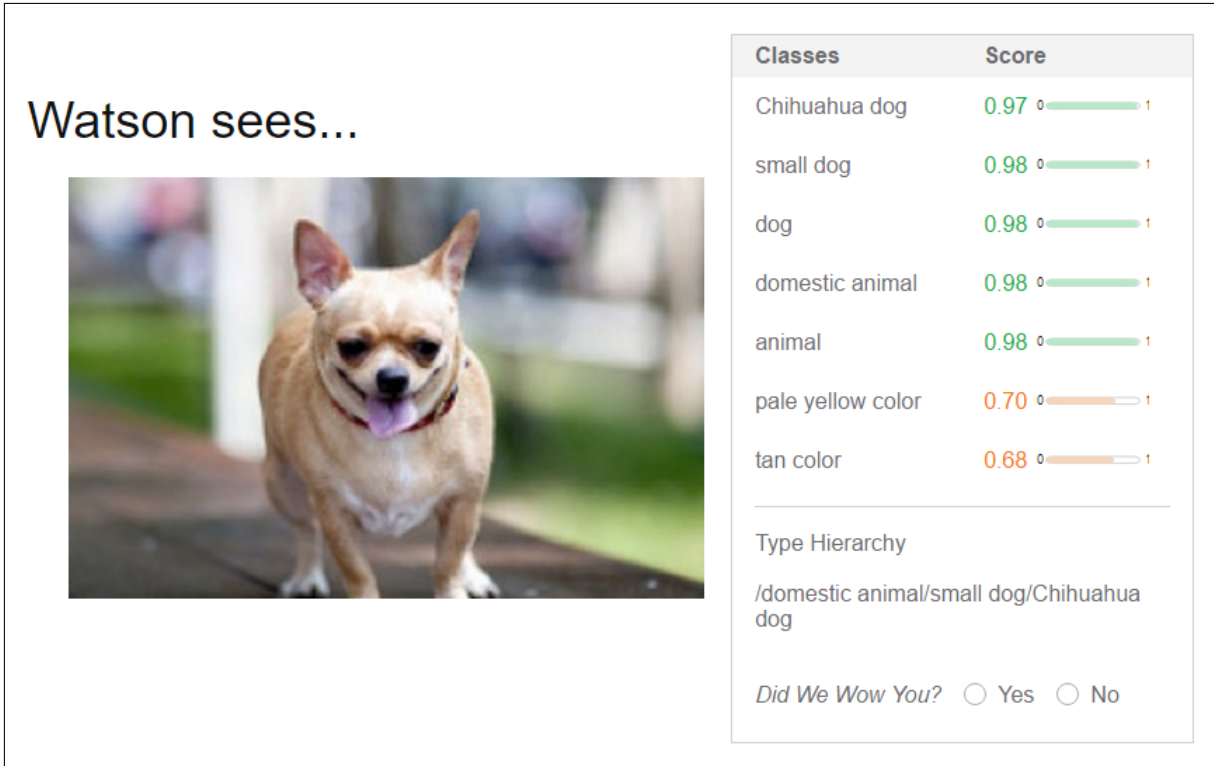


Figura 5: Exemplo de reconhecimento de um cachorro pelo IBM *Watson*.

Fonte: (IBM, 2017)

propõe um sistema de inspeção de peças automotivas.

Na área de mobilidade e segurança urbana, sistemas de análise de tráfego e monitoramento de velocidade já utilizam técnicas de visão computacional. Krishna et al. (2016) e Osman et al. (2017) propuseram esse tipo de sistema, já Khan e Iqbal (2011) propuseram um sistema de segurança residencial baseado em visão computacional que alerta o usuário via celular caso exista movimentos, de pessoas ou objetos, inesperados no ambiente desejado.

As transmissões de esportes pela televisão também já utilizam diversos recursos de visão computacional, para realizar análises de jogadas e até mesmo para auxiliar os árbitros em diferentes esportes. Liang et al. (2007) propuseram um sistema que permite a reprodução 3D em tempo real de uma partida de futebol, possibilitando a visualização do jogo e a análise de jogadas de diferentes ângulos, já Takahashi, Fujii e Yagi (2008) propuseram um sistema de identificação do tipo de arremesso realizado pelo arremessador em um jogo de *baseball* (em inglês) ou beisebol (em português), definido a trajetória e a velocidade da bola.

Assim, pode ser observado que o desenvolvimento de sistemas baseados em visão computacional é cada vez maior, e a medida em que se tornam mais precisos e seguros podem ser utilizados em atividade mais críticas, provendo uma boa expectativa para o futuro da área.

3 METODOLOGIA

Tendo em vista o objetivo do sistema proposto, de criar uma interface computacional natural baseada em visão computacional, a metodologia adotada para o seu desenvolvimento foi constituída principalmente de técnicas de engenharia de *software* com elementos de pesquisa e teste. Os procedimentos envolvidos tiveram por fim garantir que, dentre as técnicas estudadas, as mais eficientes fossem utilizadas neste sistema, tornando a ferramenta eficiente para a aplicação proposta. As etapas do método para o desenvolvimento do projeto foi composto pelos seguintes passos:

- Definição das funcionalidades do sistema.
- Levantamento das informações e técnicas necessárias para a construção das partes mais importantes do sistema.
- Definição das ferramentas necessárias para a construção do sistema.
- Desenvolvimento do sistema seguindo uma metodologia de desenvolvimento de *software*.
- Realização de testes com o sistema.
- Análise dos resultados obtidos com os testes.

A seguir, o detalhe de cada uma dessas ações:

Definição das funcionalidades do sistema: O sistema visa permitir o controle da movimentação do cursor do mouse, simulação dos cliques dos botões do mouse e simulação de digitação de teclas que podem ser configuradas pelo usuário, garantindo um grau elevado de controle do sistema operacional *Windows* e das aplicações desejadas. As especificações do sistema, determinadas antes do seu efetivo desenvolvimento, são listadas a seguir:

- O sistema deve ter como sinal de entrada a movimentação facial do usuário captada por meio de uma *webcam*. A movimentação de olhos e boca também devem ser considerados sinais de entrada.
- O sistema deve possuir 2 modos de funcionamento, o modo *mouse* e o modo *teclado*.
- No modo *mouse*, o cursor do mouse deve ser controlado pelo usuário pela direção de seu nariz, e garantir precisão para cliques em botões de pelo menos 30x30 *px* ,

bem como possibilitar a função de arrastar.

- No modo *teclado*, o usuário deve poder configurar o sistema para realizar o acionamento de cliques e teclas por meio da movimentação facial.
- A abertura e fechamento dos olhos e boca devem executar ações configuradas pelo usuário.
- O sistema deve garantir que apenas uma face detenha o controle das funções estabelecidas, a face mais próxima ao centro da imagem capturada tem prioridade de controle.
- O desempenho do sistema deve ser maior que 30 *fps*, tendo em vista que valores menores de desempenho tornam o sistema inviável para utilização devido aos atrasos que podem acontecer durante a movimentação do cursor do mouse e no acionamento de cliques e teclas.

Levantamento das informações e técnicas necessárias para a construção das partes mais importantes do sistema: Antes de realizar a construção efetiva do *software* foi necessário fazer levantamentos de informações e técnicas para viabilizar algumas funcionalidades ou determinar a melhor maneira de realizar alguma tarefa. Prioritariamente, diferentes técnicas de detecção facial e conversão de posições faciais para posições do cursor do mouse foram analisadas para determinar a melhor em termos de precisão e desempenho.

Foram realizadas buscas exploratórias e empíricas em 2 bases de dados de publicações acadêmicas, *IEEE Explorer* e *ACM Digital Library*. A partir dessa busca 56 artigos relacionados a área de pesquisa e de técnicas para detecção facial e interfaces computacionais baseadas em visão computacional, foram selecionados para análise. Desses 56 artigos 21 foram filtrados para uma leitura detalhada, pois apresentaram de maneira mais precisa o conteúdo de interesse do sistema desenvolvido. A partir da leitura desses artigos foram coletadas informações sobre técnicas principais e de pré-processamento que foram utilizadas para melhorar a eficiência e a velocidade de processamento dos sistemas de visão computacional. Como resultado dessa pesquisa, foram definidas as estratégias para o desenvolvimento do sistema.

De acordo com os artigos estudados a detecção facial realizada por classificadores fracos de características do tipo *Haar* em cascata, treinados a partir da técnica de *Adaboost*

é a forma mais precisa e com melhor desempenho, então essa foi a técnica escolhida para realizar a detecção facial. Para melhorar a velocidade de detecção alguns autores realizam pré-processamentos, que incluem regulagem de luminosidade e segmentação da cor de pele em domínio de cor YCbCr, essa segmentação diminuí as áreas para verificação de possíveis faces, essa técnica de pré-processamento foi escolhida para fazer parte do sistema desenvolvido.

As técnicas de conversão de posição face/cursor não puderam ser aproveitadas dos trabalhos analisados, pois as poucas disponíveis não são bem explicadas impossibilitando sua reprodução, assim o sistema desenvolvido utiliza uma função de transferência própria para realizar a conversão e suavização da posição do cursor do mouse. O sistema será desenvolvido para ambiente *Windows*, assim as funções de controle dos recursos do sistema operacional foram encontradas no site de referência para desenvolvedores da Microsoft (MICROSOFT, 2017).

Definição das ferramentas necessárias para a construção do sistema: As ferramentas que foram utilizadas para construção do sistema são a própria linguagem de programação e bibliotecas de apoio à linguagem. A linguagem de programação escolhida foi C++, o motivo dessa escolha se deve a 2 fatores principais, o primeiro é sua utilização em grande escala no meio profissional e acadêmico. Segundo Cass (2016), no artigo sobre o *ranking* anual de linguagens de programação mais populares, aponta a linguagem C, base da linguagem C++, em primeiro lugar ficando C++ em quarto lugar. A linguagem C, que pode ser utilizada para aplicações *desktop*, móvel e embarcado, é utilizada quando o sistema necessita de desempenho e flexibilidade, C++ além de oferecer essas características possibilita a implementação do paradigma de programação orientada a objetos e a de interfaces gráficas.

O segundo motivo para utilização dessa linguagem foi a escolha das bibliotecas utilizadas para auxiliar a implementação de recursos de visão computacional, a OpenCV e a Dlib. A biblioteca OpenCV (*Open Source Computer Vision*), de uso comercial e acadêmico é escrita em C/C++ otimizado, que aproveita ao máximo o processamento *multi-core* e aceleração de *hardware*, e possui uma grande comunidade de usuários, grande parte dela focada no desenvolvimento em C++. Muitos são os manuais de OpenCV que utilizam a linguagem C/C++, inclusive os manuais oficiais, facilitando o aprendizado e aumentando

as fontes de consulta (OPENCV, 2017). Já a biblioteca Dlib, também escrita em C++, possui algoritmos de aprendizado de máquina e visão computacional, utilizados para o desenvolvimento de aplicações voltadas a resolução de problemas do mundo real, em dispositivos como robôs e sistemas embarcados, e computação de alto desempenho (DLIB, 2017).

Desenvolvimento do sistema seguindo uma metodologia de desenvolvimento de *software*: Utilizar uma metodologia de desenvolvimento de *software* é essencial para nortear o desenvolvimento de sistemas. A metodologia escolhida para este propósito foi a SCRUM, que é uma metodologia de desenvolvimento ágil de aspecto iterativo e incremental.

O SCRUM é hoje o método mais utilizado para o desenvolvimento de *software*, pois seu uso possibilita a redução de riscos, entregas mais rápidas, flexibilidade a mudanças de escopo, aumento do valor do produto e melhora na produtividade das equipes de desenvolvimento (SABBAGH, 2014).

O presente trabalho seguiu a metodologia SCRUM em seus aspectos teóricos e de organização, como elaborar *Backlogs* e seguir *sprints* com tempo máximo de entrega.

Realização de testes com usuários: Durante o desenvolvimento do sistema foram realizados pequenos testes para garantir o funcionamento dos componentes construídos. Após a completa construção do sistema foi realizada uma bateria de testes com usuários, contemplando aspectos qualitativos de usabilidade, e aspectos quantitativos de desempenho e aplicação. Foram realizados 5 tipos de testes, para que aspectos técnicos e de usabilidade pudessem ser verificados, alguns usuários ainda foram selecionados para responder a uma pesquisa de satisfação para avaliar a usabilidade e opções do sistema. O capítulo 5 deste trabalho apresenta em detalhes os testes realizados e seus resultados. Os testes foram divididos de acordo com o modo de utilização do sistema, assim os testes 1, 2 e 3 foram elaborados para avaliar o modo *mouse*, os testes 4 e 5 para avaliar o modo *teclado* e a pesquisa de satisfação foi realizada para avaliar de forma qualitativa diversos aspectos do sistema. Os testes realizados foram:

1) Teste de movimentação do cursor do mouse: Baseado no padrão ISO/TS 9241-411, que se refere a métodos de avaliação para o desenvolvimento de dispositivos físicos de entrada sobre ergonomia de interação humano-computador. Teve por objetivo avaliar

se o sistema possibilita ao usuário controlar o cursor do mouse de forma eficiente.

2) Teste comparativo entre sistemas: Teste voltado à comparação do sistema desenvolvido com os sistemas propostos por Betke, Gips e Fleming (2002) e por Pallejà et al. (2011). Esse teste teve por objetivo verificar se o sistema desenvolvido possui vantagens sobre os disponíveis atualmente.

3) Teste de escrita em teclado virtual: Com o objetivo de verificar se essa função pode ser realizada pelos usuários do sistema, estes realizaram a escrita de frases em um editor de texto.

4) Teste de acionamentos com olhos e boca: Teste para verificar o nível de eficiência das funções de clicar e digitar teclas, a partir da movimentação de olhos e boca.

5) Teste de acionamentos com a face: Teste para verificar o nível de eficiência das funções de clicar e digitar teclas, pelos movimentos da face.

6) Pesquisa de satisfação: Foi realizado para complementar os demais testes, analisando características qualitativas como facilidade de uso, conforto e perspectivas de utilização.

Análise dos resultados obtidos com os testes: A partir dos testes do sistema foi possível chegar às conclusões cabíveis, e analisar os resultados obtidos, bem como atestar se o objetivo do projeto foi atingido.

4 DESENVOLVIMENTO DO SISTEMA

A fase de desenvolvimento do sistema proposto foi executada tendo em vista os requisitos do mesmo. Como descrito na metodologia deste trabalho, diversos trabalhos que apresentaram funcionalidades que poderiam ser utilizadas para desenvolver o sistema foram analisados. Dessa análise, as técnicas que se mostraram mais eficientes para executar determinada tarefa foram utilizadas ou serviram de inspiração para desenvolver as funcionalidades do sistema.

O sistema proposto possui 2 modos de utilização, o modo *mouse* e o modo *teclado*. O modo *mouse* permite ao usuário controlar o cursor do mouse com os movimentos da face, ativar o clique do botão esquerdo ou arrastar ícones com os movimentos da boca e ativar cliques ou digitação de teclas com as piscadas dos olhos. O modo *teclado* permite o acionamento de cliques e teclas, por meio da movimentação da face e da abertura e fechamento de olhos e boca. O usuário pode configurar quais botões serão clicados e quais teclas serão digitadas, a sensibilidade da movimentação do cursor e da abertura da boca, o tempo que o sistema permanece em pausa ao pressionar o botão [Pausar] e pode decidir se as funcionalidades relacionadas com a abertura e fechamento de olhos e boca serão habilitadas ou não.

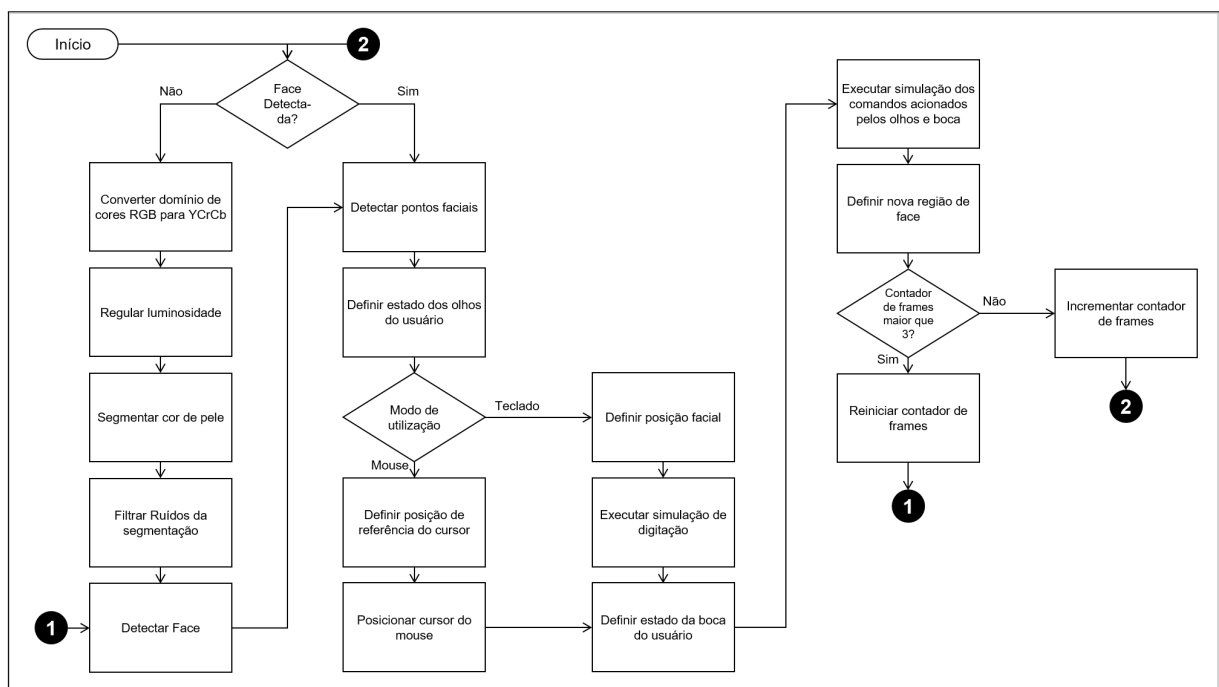


Figura 6: Fluxograma do sistema desenvolvido.

As configurações realizadas pelos usuários são gravadas pelo sistema, assim não é necessário reconfigurar as opções nas utilizações posteriores. O sistema possui 3 botões de controle, 'Parar', 'Pausar' e 'Iniciar'. Os botões de 'Parar' e 'Iniciar' possibilitam que o sistema entre ou não em funcionamento, permitindo assim que as configurações de usuário possam ser realizadas. O botão de 'Pausar' possibilita ao usuário colocar o sistema em espera, podendo então realizar outras atividades sem a interferência da ferramenta no sistema operacional, por exemplo falar sem ativar cliques com o mouse ou olhar para os lados sem movimentar o cursor.

Para que as funcionalidades do sistema possam ser melhor visualizadas, a figura 6 apresenta o fluxograma do sistema desenvolvido. Além das funções que caracterizam sistemas de interface baseados em movimentos faciais, representadas pela detecção facial, interpretação e conversão da posição da face para posição do cursor do mouse e acionamento de cliques e teclas, foi necessário implementar outras funções, entre elas pré-processamentos com o objetivo de melhorar a eficiência das funções principais. Cada uma dessas funções serão detalhadas a seguir, podendo ser observadas as técnicas utilizadas e como foram desenvolvidas.

4.1 Pré-Processamentos

A funcionalidade principal do sistema é realizar a detecção da face do usuário, sem a qual nenhuma outra funcionalidade pode ser realizada, e deve possuir duas características fundamentais, precisão e desempenho. A precisão pode ser afetada por falsas detecções, pois o ambiente em que o usuário está inserido pode conter elementos que podem ser classificados erroneamente como face. O desempenho pode ser reduzido por processamentos desnecessários e levar o sistema a apresentar atrasos, comprometendo sua utilização. Para garantir essas duas características, foi necessário implementar alguns pré-processamentos antes de realizar a primeira detecção facial, ou quando a face deixa de ser rastreada e reaparece no *frame* de entrada. A figura 7 apresenta a sequência de pré-processamentos para reduzir a área de busca facial, garantindo maior precisão e desempenho durante a detecção facial.

O primeiro pré-processamento é converter o domínio de cores de RGB para YCrCb, a *webcam* capta os *frames* no domínio de cores RGB (Figura 7 a), entretanto realizar

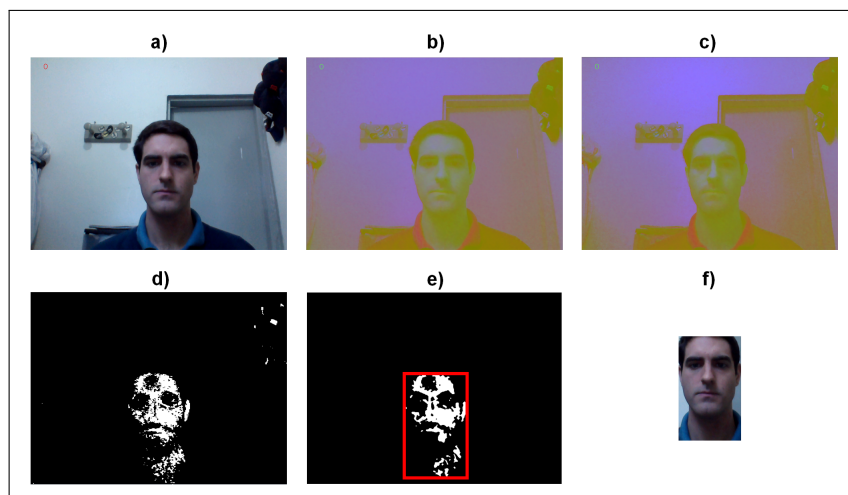


Figura 7: Pré-Processamentos realizados pelo sistema desenvolvido. a) *Frame* de entrada; b) Conversão de domínio de cor RGB para YCrCb; c) Ajuste de luminosidade por equalização de histograma; d) Segmentação de pele; e) Filtro de Ruídos; f) Imagem extraída para realizar a detecção facial.

ajustes de iluminação e segmentação nesse domínio é mais custoso computacionalmente do que no domínio YCrCb. No domínio YCrCb (Figura 7 b), o canal Y representa a luminosidade, já os canais Cr e Cb em conjunto definem a cor, assim fica mais simples realizar o ajuste de luminosidade e segmentar a cor de pele. O ajuste de luminosidade é o segundo pré-processamento realizado (Figura 7 c), esse ajuste é feito por meio de equalização de histograma do canal Y. O terceiro pré-processamento realizado é a segmentação dos *pixels* que possuem valores dos canais YCrCb correspondentes ao tom de pele (Figura 7 d), onde os valores dos canais Cr e Cb possuem intervalos pequenos e o canal Y um intervalo maior, ou seja, a cor de pele é definida por uma variação muito pequena de cor e uma variação maior de luz. A quarta operação de pré-processamento é a remoção dos ruídos da segmentação de *pixels* (Figura 7 e), o ambiente de utilização pode apresentar regiões onde os *pixels* possuem valores dentro do intervalo determinado como tom de pele, entretanto essas regiões concentram poucos *pixels* nesse intervalo, assim é aplicado um filtro *blur* no *frame*, eliminando *pixels* esparsos e mantendo apenas a região que possui uma concentração maior de *pixels* com tom de cor de pele. Por fim, a região definida como possível região de face é extraída (Figura 7 f), representando a região onde efetivamente a detecção facial será realizada. Todos esses pré-processamentos serão detalhados a seguir.

4.1.1 Conversão do domínio de cor

O objetivo dos pré-processamentos do sistema desenvolvido poderia ser atingido utilizando o domínio de cor RGB. Entretanto, a utilização desse domínio de cor em diversas operações é menos eficiente em relação às mesmas operações realizadas por meio de outros domínios de cor. Para evitar que diversas operações desnecessárias de pré-processamento fossem executadas, foi escolhido um domínio de cor diferente para compor essa fase do sistema. Trabalhos como os de Al-Mohair, Mohamad-Saleh e Suandi (2014), Qureshi et al. (2014) e Yadav e Nain (2015) apresentaram comparações entre segmentações de pele realizadas com diferentes domínios de cor, onde o domínio YCrCb se sobressai ao domínio de cor RGB para realizar essa tarefa. O ajuste de iluminação realizado no domínio de cor YCrCb pode ser realizado apenas em um canal, já no domínio RGB essa operação envolveria os 3 canais de sua estrutura. Assim, o domínio de cor escolhido foi o YCrCb, essa escolha foi determinada pelas tarefas que deveriam ser realizadas como pré-processamentos, especificamente realizar o ajuste de iluminação e segmentar *pixels* com tom de pele.

A imagem de entrada do sistema é capturada por padrão no domínio de cor RGB, então foi necessário realizar a conversão entre domínios de cor. A conversão do domínio de cor RGB para YCrCb é obtida a partir das seguintes equações (OPENCV, 2017):

$$Y = 0.299 \cdot R + 0.587 \cdot G + 0.114 \cdot B \quad (1)$$

$$Cr = (R - Y) \cdot 0.713 + 128 \quad (2)$$

$$Cb = (B - Y) \cdot 0.564 + 128 \quad (3)$$

Nesse domínio de cor, a cor do *pixel* é definida pelo canal *luma* (Y) que representa a intensidade de luz, e pelos canais *chroma* (Cr e Cb) que representam a diferença de cores em relação ao *luma*, definindo a cor do *pixel*. Uma vez que o canal Y representa a iluminação na imagem, o ajuste de iluminação pode ser realizado apenas nesse canal. A figura 8 apresenta a conversão de domínio de cor em um frame do sistema desenvolvido.

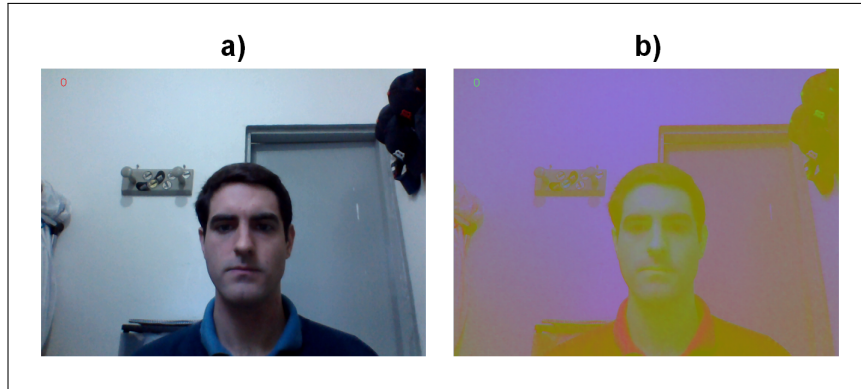


Figura 8: Conversão de domínio de cor. a) *Frame* de entrada; b) Conversão de domínio de cor RGB para YCrCb.

4.1.2 Ajuste de iluminação

A operação de segmentação de pele é altamente dependente de fatores de iluminação, assim para melhorar a eficiência dessa segmentação foi necessário implementar uma forma de compensação da iluminação. Sem essa compensação, regiões muito claras ou muito escuras podem levar o sistema a fazer segmentações falsas ou não ser segmentada (YADAV; NAIN, 2015).

No sistema desenvolvido a compensação de iluminação é realizada por meio da equalização de histograma de intensidade do canal Y da imagem. Essa operação, na prática, fez com que a região da pele do usuário se tornasse mais homogênea, evitando variações indesejadas de luz.

A operação de equalização mapeia a distribuição de intensidades dos *pixels* da imagem de entrada, para outra onde as intensidades são mais uniformes. O primeiro passo para realizar essa tarefa é traçar o histograma da imagem de entrada $h(i)$, contabilizando a quantidade de *pixel* para cada intensidade presente na imagem. Em seguida, deve-se calcular a distribuição acumulada $c(I)$ em porcentagem, dividindo o valor acumulado pelo número de *pixels* N da imagem (SZELISKI, 2010):

$$c(I) = \frac{1}{N} \cdot \sum_{i=0}^I h(i) \quad (4)$$

A função de mapeamento fme da nova intensidade do *pixel* considerando a equalização do histograma da imagem é encontrada multiplicando a função de distribuição

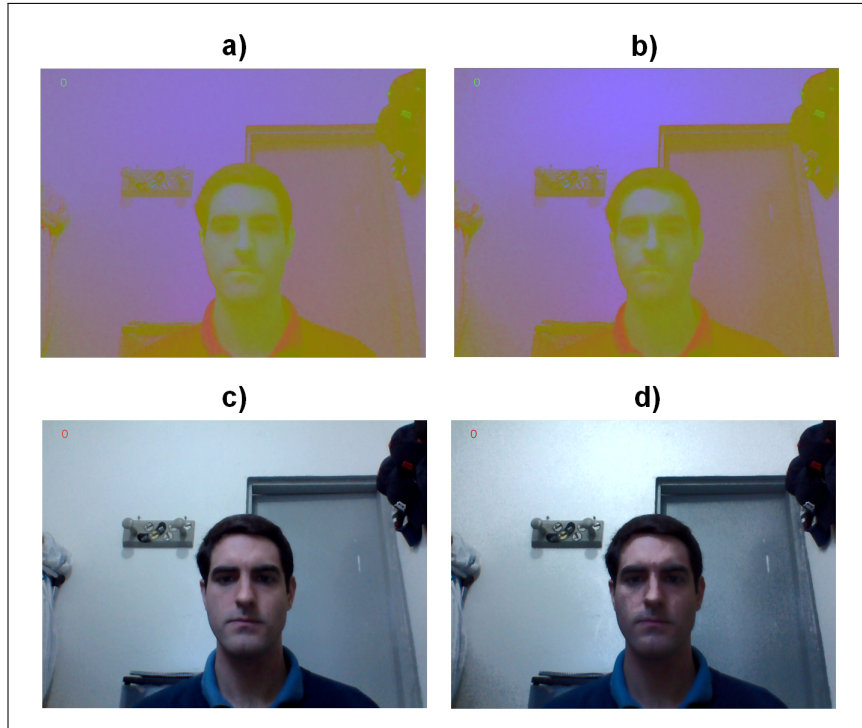


Figura 9: Ajuste de iluminação da imagem, com o objetivo de deixar a região da pele mais homogênea. a) Imagem YCrCb; b) Imagem YCrCb com iluminação ajustada; c) Imagem (a) equivalente em RGB; d) Imagem (b) equivalente em RGB.

acumulativa pela quantidade Q de intensidades possíveis na imagem menos 1, no caso 255, considerando o resultado arredondado:

$$fme(I) = c(I) \cdot (Q - 1) \quad (5)$$

Assim a nova intensidade $I_p'(x, y)$ de determinado *pixel* $I_p(x, y)$ é dado por:

$$I_p'(x, y) = fme(I_p(x, y)) \quad (6)$$

A figura 9 apresenta um exemplo de ajuste de iluminação pela técnica de equalização de histograma. Na figura 9 imagem (d) pode ser observado que a região da pele do usuário ficou mais homogênea, ou seja, essa região possui uma menor variação de luz facilitando sua segmentação.

4.1.3 Segmentação de pele

A segmentação da cor de pele do usuário tem por objetivo limitar a área de busca da detecção facial e eliminar possíveis falsos positivos de face. Tecnicamente, a segmentação de pele no sistema desenvolvido é realizada filtrando os *pixels* pelos canais do domínio de cor YCrCb dentro em um determinado intervalo.

Utilizando a biblioteca OpenCV, temos que a matriz que descreve a imagem é composta pelos canais do domínio de cor. Para o domínio de cor YCrCb, então, a matriz possui 3 canais, sendo que o intervalo de valores desses canais varia de 0 a 255. Assim, para cada um dos canais foi definido um intervalo de valores para segmentar a cor de pele, tendo como resultado uma máscara binária indicando quais *pixels* representam a pele do usuário.

Os intervalos dos canais YCrCb parametrizados no sistema desenvolvido foram baseados nos parâmetros propostos por Wang (2017):

- Y: [10, 255]
- Cr: [135, 180]
- Cb: [85, 135]

Para ser possível melhor visualizar o intervalo definido, a figura 10 apresenta o plano dos valores *chroma* Cr e Cb, o retângulo em preto representa a região de cor utilizada para compor o filtro, na figura apresentada o valor do canal Y é 128 ou 50%. Nessa figura é possível observar que a área de cor utilizada no filtro compreende tons de vermelho, amarelo e rosa, a variação do canal Y escurece ou esclarece esses tons possibilitando a detecção de diversas tonalidades de pele. A figura 11, por sua vez, apresenta um exemplo de segmentação de cor de pele do sistema desenvolvido, a imagem (a) representa o frame onde a segmentação será realizada, já a imagem (b) representa a máscara binária resultante da segmentação onde os *pixels* brancos representam possíveis regiões de pele.

Apesar da técnica de segmentação de pele poder reduzir o tempo da detecção facial, isolando o máximo possível a face do usuário, falsas detecção de pele podem acontecer. Já que a técnica faz a segmentação por intervalo de cor, objetos nesse intervalo podem ser detectados, aumentando a área detectada, móveis de madeira ou cortinas são exemplos de

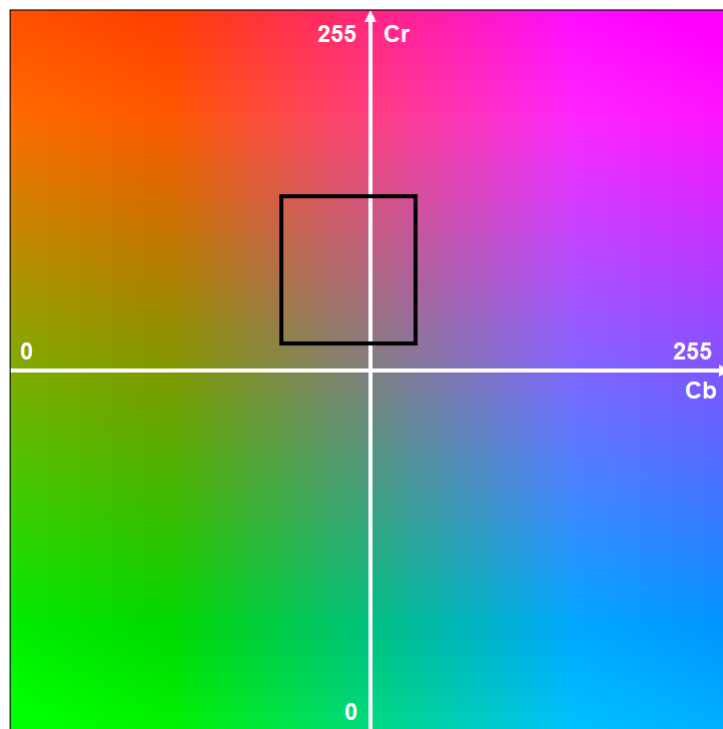


Figura 10: Plano chroma Cr e Cb com indicação da região utilizada na segmentação de pele.



Figura 11: Segmentação de cor de pele. a) Imagem com a iluminação ajustada; b) Máscara de segmentação da pele do usuário.

detecções falsas. Além desses objetos, partes do corpo que não são úteis também podem ser detectadas, como braços ou partes do peito, fazendo parte da região que será utilizada para realizar a detecção facial.

4.1.4 Filtragem de ruídos

A segmentação da cor de pele do usuário tem como resultado uma máscara binária, onde os *pixels* pretos representam regiões onde não foi detectada pele e os *pixels* brancos indicam presença de pele. Como descrito anteriormente, essa segmentação pode considerar *pixels* que não representam pele, uma vez que essa operação é apenas uma filtragem de cor. Essas falsas detecções geram na máscara de resultado *pixels* brancos esparsos, que podem ser considerados ruídos. Já que a determinação da área de realização da detecção facial é realizada por meio do mapeamento desses *pixels* brancos, a presença de ruídos aumenta a área de busca da face, comprometendo o desempenho do sistema para executar essa operação. Para remover esses *pixels* indesejáveis um filtro *blur* é aplicado sobre a máscara.

No sistema desenvolvido, o filtro *blur* utilizado para remover os ruídos da segmentação de pele é o *Median Blur*. Nessa técnica, o valor de determinado *pixel* é atualizado com os valores médios dos *pixels* vizinhos. Para o caso de uma máscara binária, uma simples contagem dos valores de *pixels* pretos e brancos vizinhos é realizada, a intensidade que obtiver maior número será o novo valor do *pixel*. O único parâmetro dessa técnica de *blur* é o tamanho do *kernel*, que caracteriza a quantidade de *pixels* vizinhos que serão considerados para realizar o cálculo médio. O sistema desenvolvido utiliza um *kernel* de tamanho $5 \times 5 \text{ px}$. Valores menores que esse se mostraram ineficientes por não remover os *pixels* esparsos. Valores maiores como $7 \times 7 \text{ px}$ e $9 \times 9 \text{ px}$ cumpriram o esperado, mas de forma não muito diferente do valor de *kernel* $5 \times 5 \text{ px}$ para *pixels* esparsos. Em blocos maiores de *pixels* brancos esses valores chegam a diminuir os blocos, mas não eliminá-los, não sendo eficientes para o sistema desenvolvido. A figura 12 apresenta um exemplo da técnica de *Median Blur* implementada no sistema desenvolvido, onde pode ser observado a remoção dos *pixels* esparsos na região superior direita da imagem.

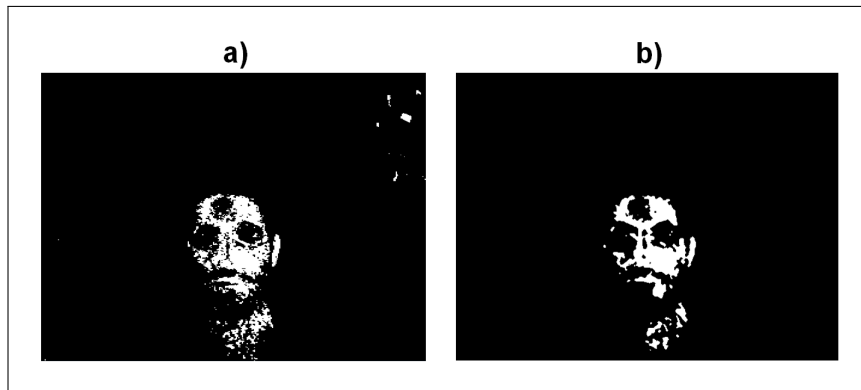


Figura 12: Filtragem de ruídos da segmentação de pele por *Median Blur*. a) Máscara de segmentação da pele do usuário; b) Máscara após a operação de *Median Blur*.

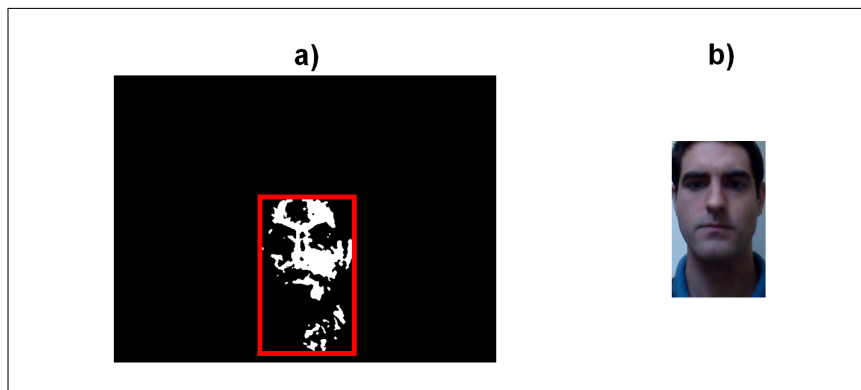


Figura 13: Definição e extração da região de possível face. a) Definição da região da pele do usuário; b) Extração da região da imagem de entrada.

4.1.5 Determinação de região de possível face

Após a finalização da segmentação de pele do usuário, o sistema desenvolvido deve determinar a região onde a face do usuário possivelmente esta localizada. Essa região é representativamente menor que a imagem de entrada, e é definida analisando-se a máscara da segmentação.

A definição da região onde será realizada a detecção facial é realizada fazendo uma varredura em toda a máscara da segmentação. As posições mínimas e máximas em x e y são armazenadas, definindo assim um retângulo, esse por sua vez é extraído da imagem de entrada formando uma imagem nova. A figura 13 apresenta o retângulo formado pelos limites da máscara de segmentação e a imagem extraída da imagem de entrada.

4.2 Detecção Facial

Com os pré-processamentos definidos e uma imagem com tamanho reduzido para realizar as análises posteriores, a tarefa de detecção facial foi implementada no sistema desenvolvido.

Yang, Kriegman e Ahuja (2002) propuseram taxinomias e classificações dessa matéria que foram e são utilizadas na maioria dos trabalhos relacionados com detecção facial, e que ajudam a descrever o estado da arte do assunto e entender as técnicas desenvolvidas, segundo o autor os métodos de detecção facial podem ser divididos em 4 categorias:

- Métodos baseados no conhecimento
- Métodos baseados na invariabilidade de características
- Métodos de correspondência de modelos
- Métodos baseados em aparência

De maneira geral, métodos baseados no conhecimento realizam a detecção a partir dos conhecimentos que o desenvolvedor possui sobre faces, passando-os ao sistema como regras que definem as características de uma face e o relacionamento entre essas características. Métodos baseados na invariabilidade de características realizam a detecção a partir da presença de elementos faciais na imagem, como olhos, nariz, boca, etc., levando em conta pontos característicos que não sofrem variações de escala, rotação, iluminação e perspectiva, os elementos faciais são encontrados e relacionados para determinar a presença ou não de uma face na imagem. Métodos de correspondência de modelos utilizam modelos de faces, geralmente faces frontais, ou características definidas por funções para realizar a detecção. No sistema desenvolvido a técnica detecção facial utilizada faz parte dos métodos baseados em aparência, especificamente a técnica proposta por Viola e Jones (2001).

Métodos baseados em aparência também utilizam comparações com modelos para realizar a detecção. A diferença desses métodos em relação aos métodos por correspondência de modelos é a definição dos modelos a serem utilizados. Os modelos baseados em aparência são construídos a partir de treinamento, onde o sistema "aprende" a identificar características de faces e de não faces, essas características são traduzidas para funções ou conjuntos de modelos de imagens, utilizados para a detecção.

Segundo Szeliski (2010) o método baseado em aparência proposto por Viola e Jones (2001) é o mais conhecido e mais utilizado método de detecção facial. Zhang e Zhang (2010) apontaram esse método como responsável por um grande avanço na área tornando possível a realização de detecção facial em tempo real, sendo considerado um padrão para a realização dessa tarefa. Gor e Bhatt (2015) mostraram as vantagens computacionais desse método tendo em vista o desempenho e precisão. O método proposto por Viola e Jones (2001) no trabalho *Rapid Object Detection using a Boosted Cascade of Simple Features*, pode ser aplicado para a detecção de qualquer objeto uma vez que o sistema deve passar por um processo de treinamento, entretanto no próprio artigo a detecção facial é utilizada nos exemplos como uma aplicação. O método, em linhas gerais, consiste em definir as *features* ou características presentes nas imagens de treinamento, determinar as características que separam as imagens positivas e negativas, e verificar em uma imagem de entrada se essas características estão presentes ou não, definindo o resultado da detecção.

Viola e Jones (2001) definiram característica como um valor numérico, produzido a partir da relação entre as intensidades dos *pixels* de uma determinada área, considerando uma imagem no domínio de cor monocromático onde um *pixel* possui um valor de intensidade que varia de 0 a 255, ou do preto para o branco. Essa relação é definida por 3 regras, chamadas de *kernels* ou modelos, e são baseadas nas regras de características do tipo *Haar* (*Haar-like*) propostas por Papageorgiou, Oren e Poggio (1998). A primeira delas é a bi-retangular (*two-rectangle feature*), definida como a diferença entre a soma de intensidade dos *pixels* de duas regiões retangulares. A segunda é a tri-retangular (*three-rectangle feature*), definida como a diferença entre a soma de intensidade dos *pixels* de dois retângulos externos com a soma de intensidade dos *pixels* da região de um retângulo interno. Já a terceira é a tetra-retangular (*four-rectangle feature*), que corresponde a diferença entre a soma de intensidade dos *pixels* dos pares diagonais de quatro retângulos. Para os 3 tipos de *kernel* as regiões retangulares possuem mesma dimensão e são adjacentes. A figura 14 apresenta a forma padrão desses 3 *kernels* e exemplos de cálculos das características.

O sistema deve realizar a extração dessas características das imagens de treinamento, considerando todas as possibilidades de tamanhos dos *kernels*, no trabalho de Viola e Jones (2001) as imagens utilizadas no treinamento possuem dimensão de 24x24 *px*, o que resulta em cerca de 180.000 características. Pensando no desempenho de processamento, e considerando que o cálculo das características é custoso, os autores propõe uma abordagem

Kernel			
Intensidade dos Pixels			
Valor da característica	150	-100	300

Figura 14: *Kernels* para cálculo de características propostos por Viola e Jones

que possibilita a computação dessas características de forma rápida, a imagem integral, com a qual o sistema despende pouco recurso de processamento para determinar a soma de intensidade dos *pixels* de uma região. Primeiramente a imagem precisa ser integrada, definindo para cada posição de *pixel* o valor da soma das intensidades dos *pixels* acima e à esquerda da posição de cálculo, que é incluso na soma. A fórmula a seguir representa o cálculo da imagem integral, onde $i(x', y')$ representa a imagem original e $ii(x, y)$ representa a imagem integral.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (7)$$

Computacionalmente, para que o cálculo da imagem integral possa ser realizado com apenas uma passagem pelos *pixels* da imagem original, o valor para uma determinada posição é dado por:

Para $x = 0$:

$$ii(x, y) = i(x, y) + ii(x, y - 1) \quad (8)$$

Para $y = 0$:

$$ii(x, y) = i(x, y) + ii(x - 1, y) \quad (9)$$

Para as demais posições:

$$ii(x, y) = i(x, y) + ii(x, y - 1) + ii(x - 1, y) - ii(x - 1, y - 1) \quad (10)$$

A figura 15 apresenta um exemplo de cálculo da imagem integral, a qual pode ser observado a facilidade do cálculo. O valor da posição $ii(2, 3)$ por exemplo, é obtido a partir apenas dos valores das posições $i(2, 3)$ da imagem original e $ii(2, 2)$, $ii(1, 3)$ e $ii(1, 2)$ da imagem integral, aplicando a fórmula proposta por Viola e Jones (2001) tem-se:

$$ii(2, 3) = i(2, 3) + ii(2, 2) + ii(1, 3) - ii(1, 2)$$

$$ii(2, 3) = 50 + 1200 + 1000 - 750$$

$$ii(2, 3) = 1500$$

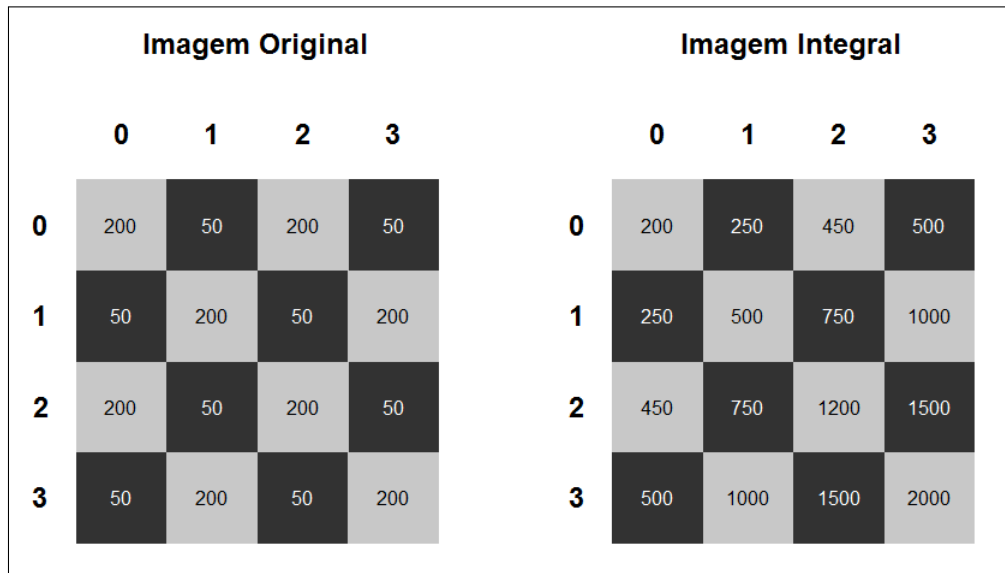


Figura 15: Exemplo de cálculo de imagem integral

A partir da imagem integral, Viola e Jones (2001) propuseram o seguinte cálculo para se obter a soma de *pixels* de uma região, onde SR representa o valor da soma da

intensidade dos *pixels* em uma determinada região, já as letras A, B, C e D representam o valor integral das intensidades dos *pixels* que delimitam a região:

$$SR = (A + D) - (B + C) \quad (11)$$

A figura 16 apresenta um exemplo de cálculo da soma das intensidades dos *pixels* em uma região. A demarcação em vermelho representa a área do cálculo, aplicando a fórmula proposta por Viola e Jones (2001) tem-se:

$$A = ii(0, 0) = 200$$

$$B = ii(3, 0) = 500$$

$$C = ii(0, 3) = 500$$

$$D = ii(3, 3) = 2000$$

$$SR = (200 + 2000) - (500 + 500)$$

$$SR = 1200$$



Figura 16: Exemplo de cálculo das intensidades dos *pixels* de uma região

Realizar o cálculo de um região é importante para possibilitar o cálculo das características de forma simples e rápida. Portanto ao realizar o cálculo das características são

utilizados 6 pontos de referência para as bi-retangulares, 8 pontos para as tri-retangulares e 9 pontos para as tetra-retangulares.

A primeira vista pode-se inferir que para realizar a detecção do objeto, tendo as características calculadas, basta verificar se as características encontradas com a base de treinamento estão presentes na imagem de entrada. Entretanto realizar essa operação considerando cerca de 180.000 características é computacionalmente muito custoso, pois exigiria a verificação de forma exaustiva. Para solucionar esse problema Viola e Jones (2001) propuseram a classificação das características que melhor distinguem as imagens positivas das negativas, reduzindo assim o número de características utilizadas na detecção, por meio do algoritmo de aprendizado *Adaboost* (*Adaptive Boosting*). Em teoria outras técnicas de aprendizado poderiam ser utilizadas, mas a simplicidade e a rapidez do *Adaboost* foram considerados para a escolha desse algoritmo por Viola e Jones (2001).

Szeliski (2010) aponta que o trabalho de Viola e Jones (2004) foi o primeiro a introduzir o conceito de *boosting* à comunidade de Visão Computacional. Essa técnica de aprendizado de máquina, consiste em treinar uma série de classificadores simples (*simple classifiers*). Cada classificador simples $h_j(x)$ é composto de uma característica f_j , um limiar numérico de característica (*threshold*) θ_j e uma paridade p_j que indica o sinal de inequação, a fórmula

$$h_j(x) = \begin{cases} 1 & \text{se } p_j f_j(x) < p_j \theta_j \\ 0 & \text{caso contrário} \end{cases} \quad (12)$$

representa um classificador simples. O algoritmo de aprendizado determina o *threshold* ótimo para cada uma das características, dependendo da base de treinamento utilizada. De maneira geral, o classificador final é a somatória de todos os classificadores simples definidos.

Segundo Viola e Jones (2001) cerca de 200 características garantem 95% de precisão. A figura 17 apresenta exemplos de características selecionadas pelo algoritmo do *Adaboost* e representam as áreas da região dos olhos.

Na prática, o treinamento e a detecção não são realizadas juntas. Após a fase de treinamento um arquivo é gerado para ser utilizado pelo algoritmo de detecção, e possui uma estrutura que atenda ao conceito, também proposto por Viola e Jones (2001), de

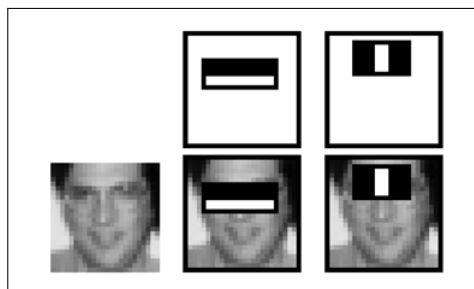


Figura 17: Características definidas pelo *Adaboost*

Fonte: (VIOLA; JONES, 2001)

classificação em cascata.

Para aumentar a velocidade de detecção a imagem de entrada que será analisada é dividida em regiões em diferentes escalas, para cada região verifica-se a existência das características descritas no arquivo de treinamento. Essa análise é executada em camadas, onde cada camada possui um conjunto de características de quantidades crescentes, ou seja, nas primeiras camadas são descritas poucas características nas últimas um número maior de características. Caso a região analisada não possua as características da camada a mesma é classificada como não face, sendo excluída da análise, no momento em que uma determinada região atender todas as características de todas as camadas então é classificada como face.

No sistema desenvolvido, a base de treinamento utilizada é disponibilizada na biblioteca OpenCV treinada pelo algoritmo *Adaboost* com imagens positivas e negativas de faces de tamanho $20 \times 20 \text{ px}$. O algoritmo de detecção facial tem como resultado as regiões retangulares das faces encontradas, dispostas em um vetor, desse resultado é necessário determinar qual face, caso seja encontrado mais de uma, deve ser considerada a face do usuário do sistema. A regra para essa definição é a face mais próxima do centro do *frame* analisado, uma vez que é esperado que o usuário se posicione em frente à câmera olhando-a diretamente. Determinar apenas uma face para controlar o sistema, além de não causar problemas de utilização, permite que uma pessoa que acompanhe as atividades do usuário possa permanecer ao seu lado.

Todos os procedimentos de pré-processamento e a detecção facial são realizados, quando o sistema não possui ainda nenhuma face registrada. Após a primeira detecção, ou quando a face deixa de ser detectada e precisa ser redetectada, o sistema desenvol-

vido adota procedimentos de rastreamento da face detectada. Esse rastreamento, e não detecções e redetecções constantes, visa garantir maior desempenho e eficiência no sistema. Para tanto, o sistema armazena a área da face detectada e incrementa o tamanho dessa área em um fator de 13%. As próximas detecções faciais somente acontecem a cada 3 *frames*, sendo utilizada apenas a região da última detecção, o incremento das regiões detectadas é necessário para que a movimentação da face do usuário permaneça dentro dessa região possibilitando a nova detecção. Esse valor de incremento foi encontrado de forma empírica, verificando a melhor taxa de incremento levando em consideração a nova detecção e seu tempo de processamento. Essa forma de detectar e rastrear a face, baseada na forma de realizar essa tarefa proposta por Kraichan e Pumrin (2014), permitiu uma melhora significativa no desempenho do sistema, praticamente dobrando a velocidade de processamento da detecção facial. A área com acréscimo de tamanho foi utilizada para realizar todos os procedimentos posteriores do sistema, como a detecção dos pontos faciais.

4.3 Detecção dos Pontos Faciais

A detecção facial provê para o sistema a região de localização da face do usuário. Entretanto essa região não é informação suficiente para o sistema realizar suas demais funções. Para executar funções como a movimentação do cursor do mouse e análise de movimentação da boca, o sistema necessita interpretar a posição da face e de seus elementos de forma isolada, a fim de transformar essas posições em funcionalidades. A detecção dos pontos da face permite, a partir da distância entre esses pontos, determinar a posição da face, a posição dos olhos e a extensão da abertura da boca.

O sistema desenvolvido utiliza a biblioteca Dlib para implementar a detecção dos pontos faciais. O algoritmo disponibilizado nessa biblioteca foi proposto por Kazemi e Sullivan (2014), que define uma fase de treinamento e outra de detecção para realizar essa tarefa. A detecção dos pontos da face é realizada utilizando uma cascata de regressores, assim a fase de treinamento visa definir as funções de regressão e determinar o número de regressões.

Para realizar o treinamento dos regressores Kazemi e Sullivan (2014) utilizaram o algoritmo de aprendizado de máquina *gradient boosting* e necessitam de 3 elementos para construir a cascata de regressores: uma base de treinamento com imagens de faces; os

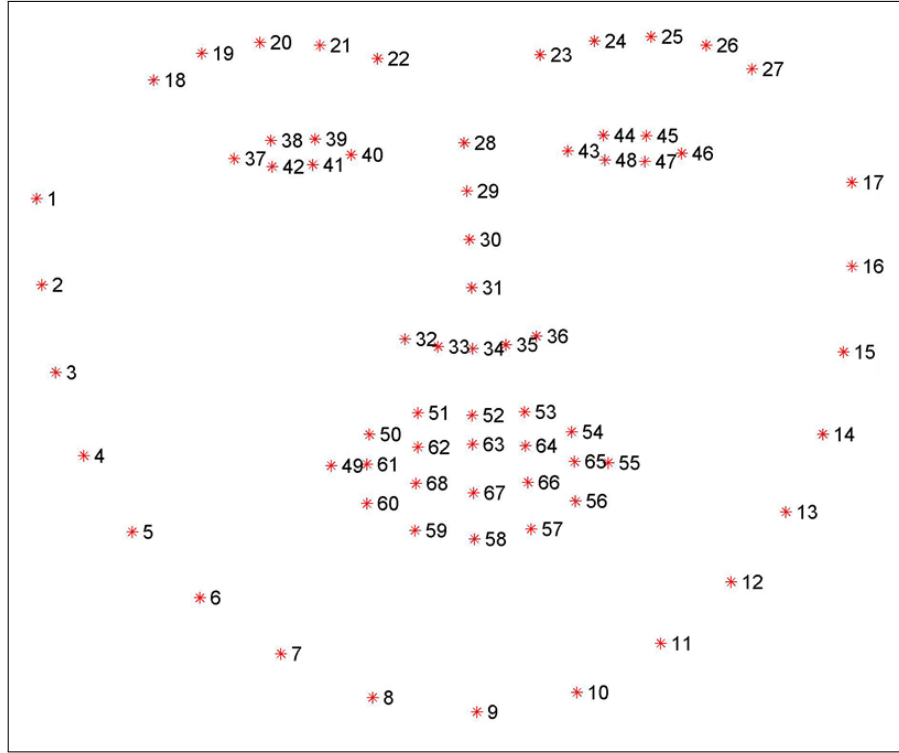


Figura 18: 68 pontos faciais detectados

Fonte: (SAGONAS et al., 2013a)

pontos faciais de cada imagem dispostos em forma de vetor; e a diferença entre o vetor real dos pontos de uma face e a estimativa dos pontos na iteração atual do treinamento. Esses 3 elementos formam uma tupla, $(I_{\pi i}, \hat{S}_i^{(t)}, \Delta S_i^{(t)})$, que é chamada pelos autores de *triplet*.

O arquivo modelo de treinamento disponibilizado pela biblioteca Dlib foi treinado com a mesma base utilizada por Kazemi e Sullivan (2014), iBUG 300-W (IBUG, 2017). Essa base de treinamento descreve em cada imagem de face um vetor com as posições de 68 pontos faciais, propostos por Sagonas et al. (2013a), e podem ser observados na figura 18.

A partir da base de treinamento é possível calcular o primeiro regressor e consequentemente os demais. O regressor inicial é encontrado a partir do vetor de pontos de cada imagem em relação à média dos vetores de todas as imagens da base. A tupla que descreve o primeiro regressor é definido por $(I_{\pi i}, \hat{S}_i^{(0)}, \Delta S_i^{(0)})$ onde:

I : Imagem de Face;

S : Vetor de pontos da face;

$$\pi i \in \{1, \dots, n\} \quad (13)$$

$$\hat{S}_i^{(0)} \in \{S_1, \dots, S_n\} \setminus S_{\pi i} \quad (14)$$

$$\Delta S_i^{(0)} = S_{\pi i} - \hat{S}_i^{(0)} \quad (15)$$

Os valores do vetor $\hat{S}_i^{(0)}$, utilizado para calcular o passo de atualização $\Delta S_i^{(0)}$, nesse primeiro regressor, são as médias dos valores dos vetores reais de pontos de todas as imagens ($S_{\pi i}$). A partir desses dados o valor do regressor r_0 é encontrado, aplicando um algoritmo de cálculo descrito por Kazemi e Sullivan (2014). Para calcular os demais regressores as tuplas devem ser atualizadas, assim

$$\hat{S}_i^{(t+1)} = \hat{S}_i^{(t)} + r_t \left(I_{\pi i}, \hat{S}_i^{(t)} \right) \quad (16)$$

$$\Delta S_i^{(t+1)} = S_{\pi i} - \hat{S}_i^{(t+1)} \quad (17)$$

O algoritmo de treinamento dos regressores proposto por Kazemi e Sullivan (2014) realiza, de forma iterada, o cálculo de T regressores $\{r_0, r_1, \dots, r_{T-1}\}$ até que juntos possam garantir um nível satisfatório de precisão na detecção dos pontos da face. Esse algoritmo estabelece como núcleo de cada função de regressão um conjunto de regressores organizados em árvore. Cada nó dessa estrutura toma uma decisão tendo como base a diferença entre as intensidades de dois *pixels*. Essa árvore de regressão mapeia, por meio da base de treinamento, as características dos *pixels* da região do entorno dos pontos da face. Assim, no momento da detecção dos pontos faciais o detector tenta reproduzir o comportamento encontrado na base de treinamento para a imagem de entrada. A figura 19 apresenta um exemplo de uma árvore de regressão que compara intensidade de *pixels*. A comparação utiliza as intensidades dos *pixels* nas posições de dois pontos faciais S e a diferença de posição entre esses pontos Δ , as folhas da árvore armazenam a variação de posição para mover determinado ponto facial para o alvo.

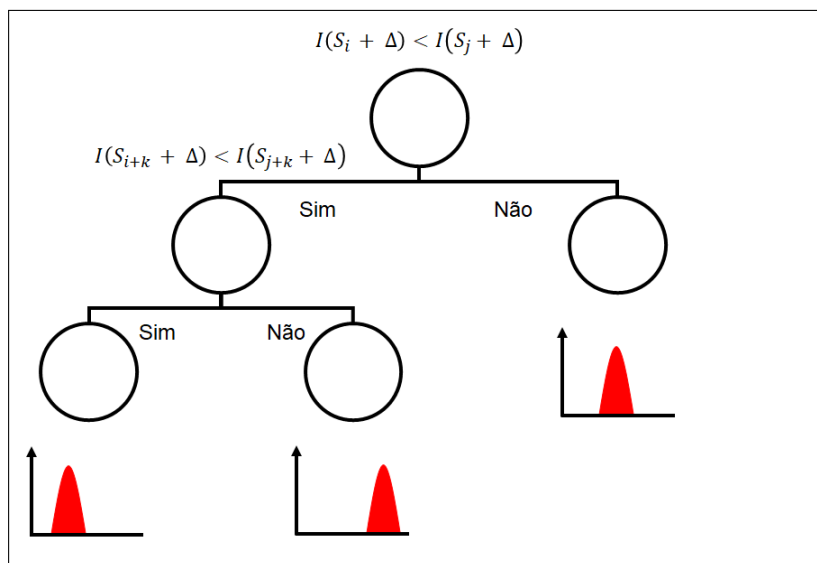


Figura 19: Exemplo de Árvore de Regressão que compara intensidade entre *pixels* para tomada de decisão.

Fonte: (MITCHELL, 2018)

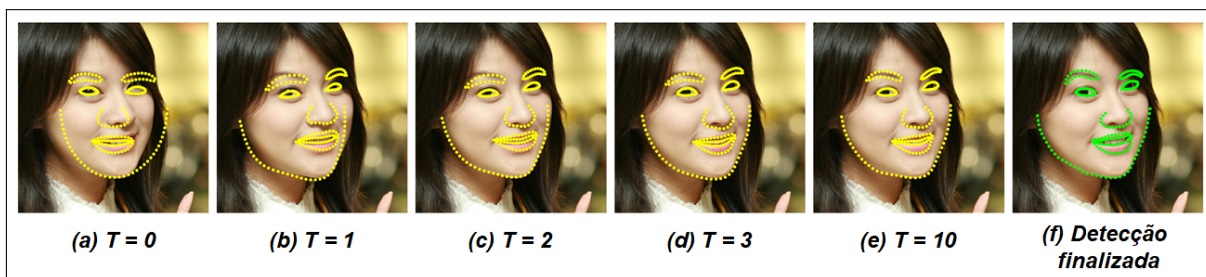


Figura 20: Exemplo de detecção dos pontos faciais.

Fonte: (KAZEMI; SULLIVAN, 2014)

Na fase de detecção a imagem da face é submetida à estimativa dos seus pontos faciais, onde cada regressor aprendido é aplicado sequencialmente, formando assim uma cascata de regressores. A figura 20 apresenta um exemplo de detecção dos pontos faciais, onde é possível verificar o ajuste de cada predição utilizando os regressores.

A figura 21 demonstra a detecção dos pontos faciais no sistema desenvolvido e algumas referências utilizadas para desenvolver suas funcionalidades. Os pontos verdes, azuis e vermelho correspondem aos 68 pontos faciais detectados. O ponto vermelho é utilizado como referência para a movimentação do cursor do mouse. Os pontos azuis representam os limitadores em x e y da movimentação do cursor. A partir dos pontos faciais da região dos olhos foi possível identificar as posições dos olhos, representadas pelos retângulos

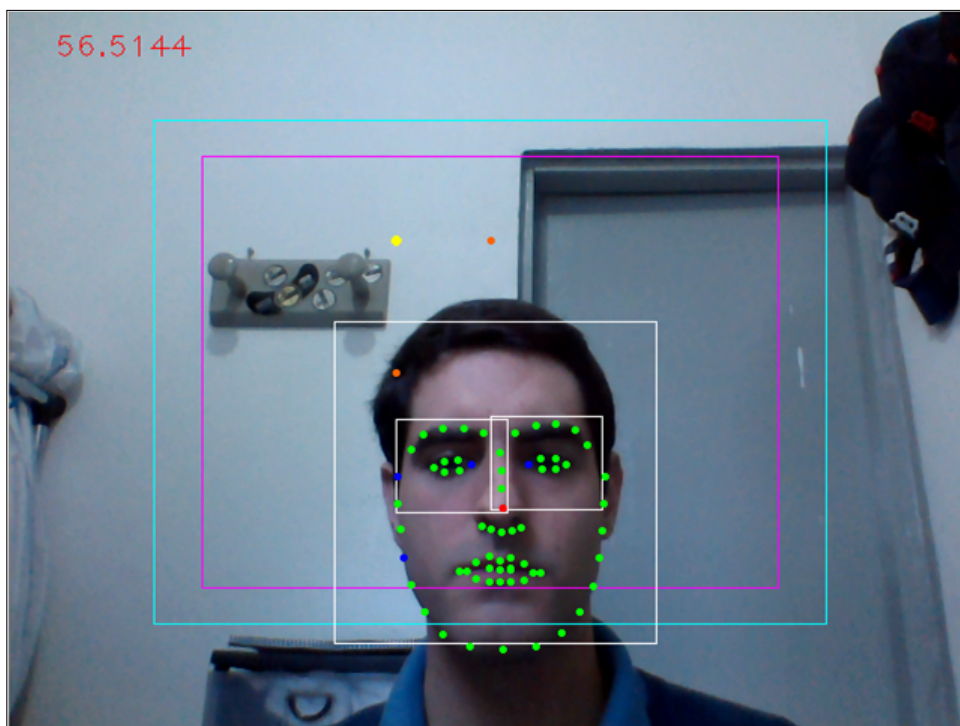


Figura 21: Pontos faciais e regiões utilizados no sistema desenvolvido.

brancos em torno dos olhos, utilizadas para definir se os olhos estão abertos ou não. Os pontos faciais da região da boca permitiram identificar a extensão da abertura da boca. Nas seções seguintes será apresentado como os pontos faciais foram utilizados para implementar diversas funções do sistema desenvolvido.

4.4 Posicionamento do cursor do mouse

O sistema foi planejado e desenvolvido para permitir que o usuário determine o modo de funcionamento do mesmo. Essa escolha define quais funcionalidades o sistema deve executar. O modo *mouse* utiliza o posicionamento da face para movimentar o cursor do mouse, já o modo *teclado* utiliza essa movimentação para acionar cliques e teclas, ambos utilizam os pontos faciais.

A principal funcionalidade do modo *mouse* é permitir ao usuário posicionar e movimentar o cursor do mouse com os movimentos da face. Para realizar essa tarefa o sistema precisa interpretar a posição da face e converter essa posição para a posição do cursor em tela. A posição da face do usuário é encontrada comparando a distância entre um ponto de referência, estabelecido como a ponta do nariz do usuário, representado pelo ponto 31

da figura 18, que apresenta a posição dos pontos faciais detectados, e os pontos faciais limitadores de eixo. Os pontos de referência para limitar a movimentação horizontal (eixo x), são os pontos 40 e 43 da figura 18, localizados nas extremidades internas dos olhos. Os pontos faciais utilizados para limitar a movimentação vertical (eixo y), são os pontos 1 e 4 da figura 18, localizados no topo e na base da orelha direita do usuário. O sistema analisa percentualmente a posição do ponto de referência em relação aos limitadores dos eixos, essa posição é utilizada para determinar o posicionamento em *pixel* dos pontos de origem e destino da movimentação do cursor do mouse. Para evitar porcentagens negativas ou superiores à 100%, o sistema considera como área útil de movimentação 60% da área delimitada pelos extremos de referência dos eixos, criando uma área morta de 20% em cada extremo. Assim, a área útil de movimentação varia de 20% a 80% da distância entre os pontos limitadores de eixo, como apresentado na figura 22.

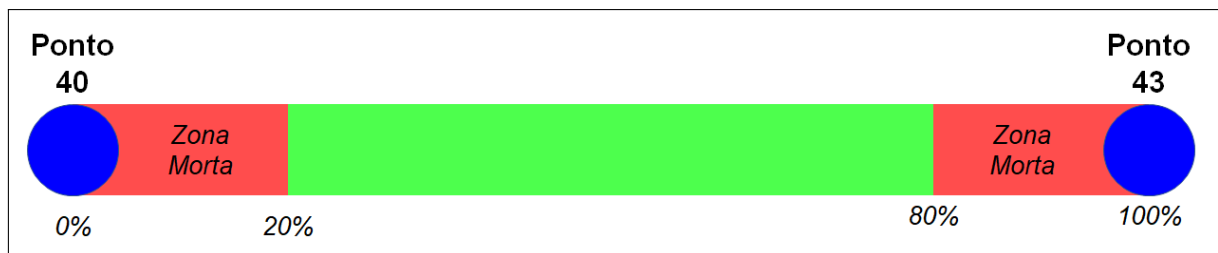


Figura 22: Área de movimentação de referência de eixo considerado pelo sistema (eixo x).

A conversão direta da posição facial para a posição do cursor do mouse, apresentou 2 problemas na sua implementação. O primeiro foi a sensibilidade acentuada, uma vez que pequenas movimentações da face geram grandes movimentações do cursor, impossibilitando que o usuário posicione o cursor com precisão e suavidade. O segundo foi a instabilidade do posicionamento do cursor, causada pela detecção não constante dos pontos faciais, ou seja, mesmo que o usuário não movimente a face a detecção dos pontos faciais não acontece sempre no mesmo lugar, já que a detecção desses pontos é uma estimativa. Para resolver os problemas citados foi necessário implementar uma função de transferência.

Kjeldsen (2006) e Pallejà et al. (2011) apresentaram em seus trabalhos funções de transferência que permitem aumentar a precisão do posicionamento do cursor. A ideia central dessa função é modificar a última posição do cursor, durante sua movimentação,

para que a mudança de posição não aconteça de forma abrupta e sim de forma progressiva. Essa progressão deve ser dinâmica, ou seja, para variações pequenas de posição a progressão deve ter determinado comportamento, geralmente lento, para variações maiores a progressão deve ter outro comportamento, geralmente rápido.

O sistema desenvolvido utiliza uma função de transferência própria para desempenhar a conversão de posicionamento da face para o posicionamento do cursor do mouse. O desenvolvimento dessa função de transferência iniciou-se com a análise de diversas curvas de funções matemáticas, para determinar qual delas apresenta uma curva de comportamento adequada para o sistema.

Foram analisadas diversas curvas de funções matemáticas, entretanto a curva da função logarítmica $f(x) = \log_n x$ possui o comportamento desejado para a movimentação do cursor do mouse, pois valores pequenos de x geram pequenos valores de $f(x)$ e valores maiores de x geram valores maiores de $f(x)$. Essa variação acontece de forma mais suave que valores de conversão linear, por exemplo. A base n da função logarítmica varia a acentuação da curva da função, como pode ser observado na figura 23. A função logarítmica natural, em laranja, foi escolhida por apresentar a acentuação de variação desejada e por ser de fácil implementação na linguagem C.

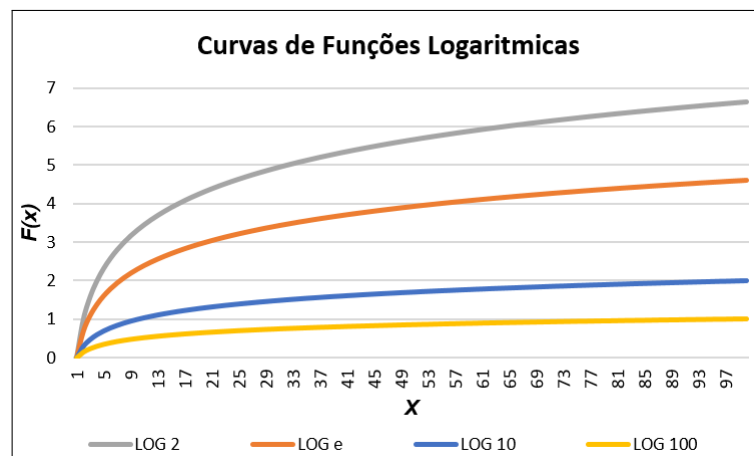


Figura 23: Curvas de Funções logarítmicas.

Conceitualmente, deixar a movimentação do cursor do mouse mais suave e precisa consiste em evitar que o cursor seja posicionado abruptamente entre os *frames*, forçando-o a seguir um caminho até a posição desejada. Assim, foi necessário que ao invés do posicionamento percorrer todo o caminho entre a posição atual, desejada pelo usuário, e

a posição anterior, percorresse apenas parte desse caminho. Analisando numericamente a função $f(x) = \log_e x$, foi constatado que para $x = 1$ temos que $f(x) = \ln(1) = 0$ e para $x = e$ temos que $f(x) = \ln(e) = 1$, ou seja, nessa função podemos considerar que o intervalo $\{1, \dots, e\}$ para o valor de x gera uma porcentagem, quando calculado o $\ln(x)$. Essa porcentagem pode ser utilizada para determinar quanto o cursor deve ser movimentado em 1 frame processado, tendo em vista que a movimentação por essa porcentagem terá o mesmo comportamento da curva da função logarítmica natural.

Para que a porcentagem de movimentação seja calculada, antes é necessário determinar o quanto a diferença Δd entre a posição desejada pelo usuário e a posição anterior, representa na escala do intervalo $\{1, \dots, e\}$. Tendo como base o intervalo $\{0, \dots, b\}$, onde o valor de b representa o limitador da escala de Δd e que na prática equivale a sensibilidade de movimentação, uma conversão de escalas é realizado. Para tanto a seguinte equação é aplicada para conversão entre escalas:

$$\frac{b - \Delta d}{b - 0} = \frac{e - \Delta d_e}{e - 1} \quad (18)$$

simplificando temos:

$$be - \Delta de - b + \Delta d = be - b\Delta d_e \quad (19)$$

$$-\Delta de - b + \Delta d = -b\Delta d_e \quad (20)$$

$$\Delta de + b - \Delta d = b\Delta d_e \quad (21)$$

$$\Delta d_e = \frac{\Delta de + b - \Delta d}{b} \quad (22)$$

Valores negativos de Δd resultam em uma conversão fora do intervalo $\{1, \dots, e\}$, logo um valor negativo de porcentagem. Para evitar que isso ocorra os valores de Δd devem

ser sempre positivos, assim deve ser considerado o módulo de Δd :

$$\Delta d_e = \frac{|\Delta d| e + b - |\Delta d|}{b} \quad (23)$$

A figura 24 apresenta de forma gráfica a correlação entre a porcentagem atribuída à movimentação do cursor e o limitador de escala b , comparando ainda com a correlação realizada de forma linear.

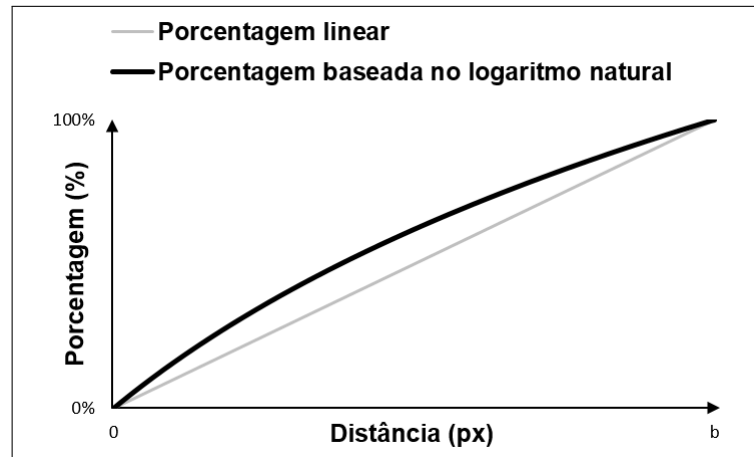


Figura 24: Curva logarítmica natural e linear da correlação entre escala de distância e porcentagem de movimentação.

O valor encontrado após o cálculo de Δd na escala $\{1, \dots, e\}$ é utilizado para determinar a porcentagem de movimentação do cursor do mouse, aplicando a função logarítmica $\ln(x)$. O valor em *pixels* dessa movimentação é calculado multiplicando a porcentagem por Δd , esse valor deve ser inteiro, razão pela qual foi necessário arredonda-lo. Valores positivos de movimentação indicam que o usuário deseja movimentar o cursor do mouse para cima ou para a direita, já valores negativos indicam movimentações para baixo ou para a esquerda. Por fim, a posição do cursor do mouse P' é calculada somando o valor de movimentação à última posição do cursor P . Assim, a função de transferência utilizada no sistema é representada pela equação:

$$P' = P + \text{round} \left(\Delta d \cdot \ln \left(\frac{|\Delta d| e + b - |\Delta d|}{b} \right) \right) \quad (24)$$

onde,

P' : Nova posição do cursor;

P : Posição atual do cursor;

Δd : Diferença entre a distância desejada pelo usuário e a posição atual do cursor;

e : Número de Euler;

b : Base de sensibilidade sobre a curva do logaritmo natural.

Com essa função, o usuário é capaz de posicionar o cursor do mouse de forma suave e precisa, além de eliminar a instabilidade ocasionada pela detecção dos pontos da face. O comportamento da movimentação do cursor segue o mesmo comportamento da curva do logaritmo natural, ou seja, para movimentações pequenas o sistema restringe a movimentação do cursor, assim o usuário consegue posicionar o cursor sobre áreas pequenas, como botões, já para movimentações maiores o sistema restringe menos a movimentação do cursor. A restrição que o sistema exerce sobre a movimentação do cursor do mouse é definida pela base de sensibilidade b , assim para uma base $b = 100$ uma movimentação de 50 px desejada pelo usuário representa em 1 *frame* processado 31 px de deslocamento, para uma base $b = 3000$ a mesma movimentação representa apenas 1 px de deslocamento em 1 *frame* processado. A figura 25 ilustra a comparação de movimentação do cursor do mouse sem a função de transferência e com a função de transferência. Pode ser observado que sem a função de transferência não seria possível utilizar o sistema.

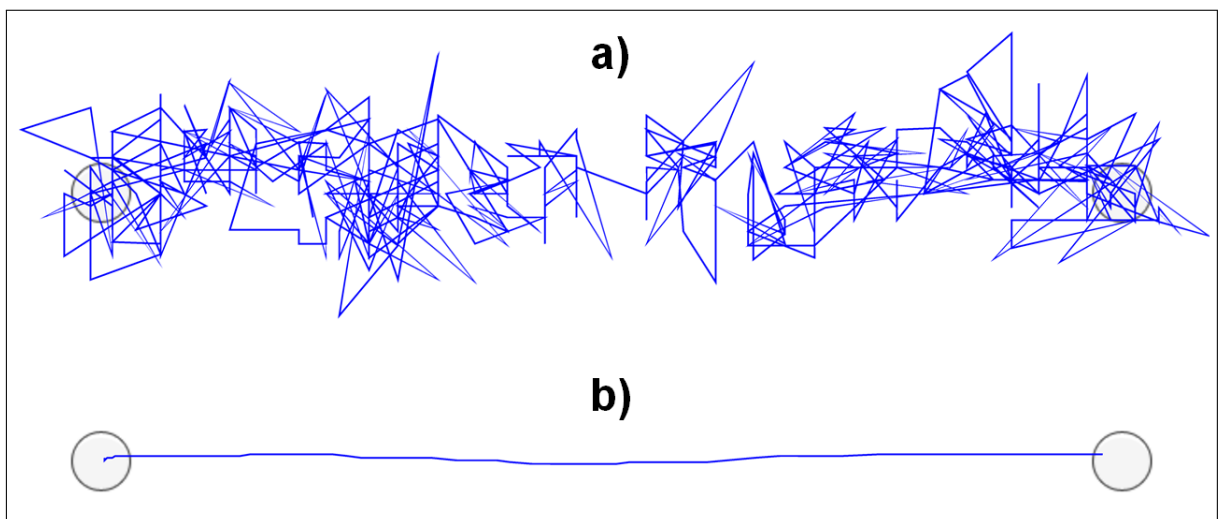


Figura 25: Comparação de posicionamento do cursor do mouse utilizando ou não a função de transferência. a) Posicionamento sem a função; b) Posicionamento com a função.

4.5 Acionamento de cliques e teclas

A função de acionamento de cliques dos botões do mouse e de teclas, realizadas pelo sistema desenvolvido é baseada na movimentação da face, dos olhos, e da boca. O sistema verifica o estado de abertura dos olhos e da boca e no caso da face é verificado a direção de posicionamento.

A definição da direção da face somente é realizada no modo *teclado* do sistema. O ponto do cursor de referência do modo *mouse*, antes da fase de suavização pela função de transferência é verificado. Caso o ponto de referência ultrapasse o retângulo magenta, apresentado na figura 21 (Pontos da face e regiões utilizados pelo sistema), o estado da direção da face é determinado, para cima, para baixo, para esquerda e para direita, essa definição pode conter 2 estados caso a face ultrapasse 2 direções, por exemplo para cima e esquerda. Numericamente, é medido o posicionamento do ponto 31, da figura 18 (Pontos da face detectados), em relação aos limites de eixo, pontos 40 e 43 para o eixo x e 1 e 4 para o eixo y . Caso essa posição esteja dentro de 20% em relação aos extremos a posição da face é determinada. Na prática o posicionamento é considerado caso o ponto de referência esteja dentro da zona morta representada na figura 22 (Área de movimentação de referência de eixo) para o eixo x , o que acontece de maneira análoga para o eixo y .

A definição do estado de abertura da boca é realizado analisando a distância entre os pontos 63 e 67, localizados nos lábios superiores e inferiores do usuário. A distância delimitadora de abertura é calculada por uma porcentagem do tamanho da face do usuário, encontrado pela distância entre os pontos 9 e 22, localizados no queixo e na sobrancelha direita do usuário, respectivamente. Por exemplo, considerando o tamanho da face de 150 px , e utilizando uma porcentagem delimitadora de 3%, a boca é considerada aberta se a distância entre os *pixels* 63 e 67 for maior que 5 px , caso a porcentagem delimitadora fosse de 5% a distância necessário seria de 8 px . Essa forma de determinar o estado da boca tem a vantagem de ser invariante à distância entre o usuário e o computador, entretanto tem comportamentos diferentes entre usuários, já que o tamanho de face pode variar.

Em um primeiro momento, foi realizada a definição do estados do olhos da mesma forma da definição do estado da boca, por distância entre *pixels*, entretanto a variação de distância entre os *pixels* da região dos olhos é muito pequena, não garantindo precisão.

Assim, outra técnica foi utilizada para realizar essa tarefa, a detecção de olhos abertos. Utilizando a mesma técnica da detecção facial, proposta por Viola e Jones (2001), mas com uma base de treinamento de olhos abertos, os olhos de forma isolada são detectados. Essa detecção acontece com a mesma frequência da detecção facial, a cada 3 *frames*, equivalente à 30 *ms*, e caso duas não detecções seguidas aconteçam o olho é considerado fechado. Fazer a dupla verificação de não detecção elimina movimentos involuntários de piscada de olhos.

Com a definição dos estados de olhos, face e boca, o sistema desenvolvido envia ao sistema operacional uma solicitação para executar a ação desejada, configurada pelo usuário por meio de uma tela de configuração.

4.6 Tela de configuração e funções do sistema

A última fase de implementação do sistema desenvolvido foi a construção de uma tela de configuração e funções auxiliares implementadas no desenvolvimento dessa tela. A figura 26 apresenta a tela de configuração do sistema desenvolvido, onde é possível verificar a presença das opções de configuração do sistema. O usuário pode configurar o modo de funcionamento; habilitar ou não as funcionalidades relacionadas aos olhos e boca; os cliques e teclas que serão acionados a partir dos movimentos da face, olhos e boca; a sensibilidade de movimentação do cursor do mouse; a sensibilidade da abertura da boca e o tempo de pausa do sistema quando essa função for executada. A escolha do modo de funcionamento do sistema determina quais opções podem ser configuradas.

No modo *mouse* os movimentos da face são utilizados para controlar a movimentação do cursor do mouse. Piscar os olhos, caso habilitados, podem ser utilizados para acionar a digitação da tecla *Enter* e o clique com o botão direito do mouse. A abertura da boca realiza o clique com o botão esquerdo do mouse. O usuário pode configurar a sensibilidade do cursor do mouse, esse valor é interpretado para gerar uma base de conversão de escala b diferente durante o cálculo da função de transferência, variando entre os valores $\{2000, 3000, 5000, 7000, 9000\}$, quanto maior o valor de b menos sensível será a movimentação do mouse que acontece de forma mais lenta.

Já no modo *teclado* o usuário pode configurar quais teclas serão acionadas a partir

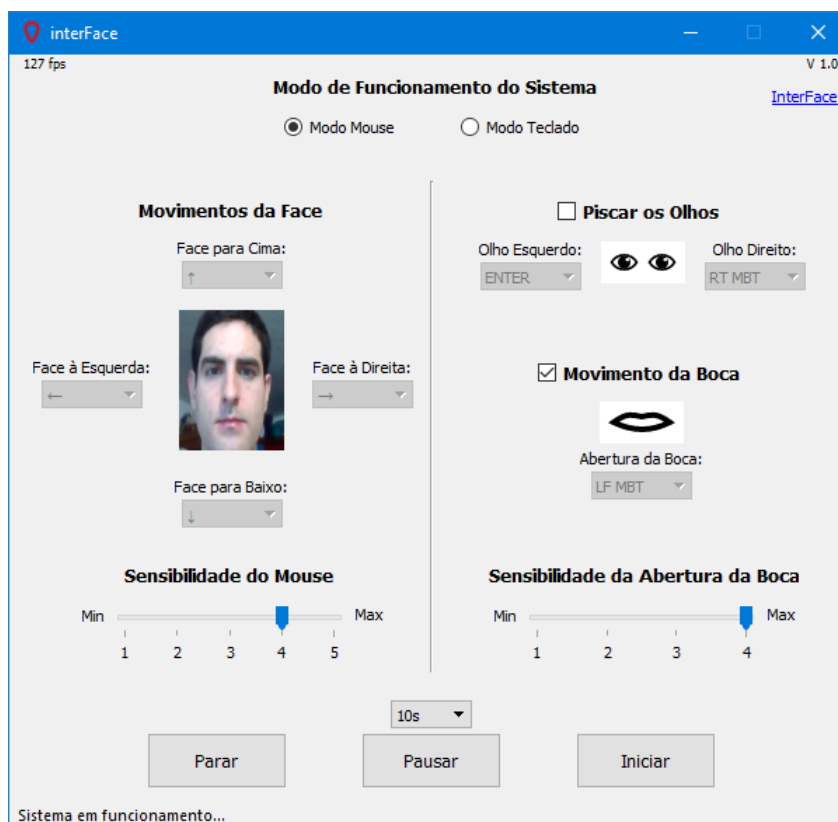


Figura 26: Tela de configuração do sistema desenvolvido.

da movimentação da face, dos olhos e boca. A lista de opções de acionamento possui 94 itens, desde comandos como *Shift*, *Alt* e *Enter*, números e letras até comandos de controle de volume, *media player* e *browser* de internet, possibilitando a utilização de inúmeras aplicações. O usuário tem a opção de regular a sensibilidade da abertura da boca para acionar alguma tecla. Anteriormente foi descrito que a quantidade de abertura para definir o estado da boca se baseia na diferença entre os pontos faciais 63 e 67, em relação aos pontos 9 e 22, com essa configuração o usuário varia a definição de abertura no intervalo de porcentagem {3%, 5%, 7%, 9%}. Essa configuração é importante devido a variabilidade entre os usuários da distância entre os pontos 9 e 22, que representa o tamanho da face do usuário, assim a medida de 3% para uma pessoa com a face pequena pode se tornar muito sensível, devendo o sistema ser configurado com valores maiores desse parâmetro.

Além das configurações apresentadas, o usuário tem recursos de controle do sistema, com as opções de Iniciar, Pausar e Parar seu funcionamento. O recurso de Iniciar/Parar permite que o sistema possa ser configurado e iniciado, ou parado para que o sistema ope-

racional possa ser utilizado normalmente. Já o recurso de Pausar, variando no intervalo $\{5 s, 10 s, 20 s, 30 s, 1 min, 2 min, 3 min, 5 min, 10 min\}$, é útil para o usuário quando este deseje, por exemplo, conversar com outra pessoa ou utilizar uma aplicação de reconhecimento de voz, já que o sistema utiliza a movimentação da boca. Neste último recurso, após o tempo de pausa ser atingido o sistema volta ao seu funcionamento normal de forma automática.

5 TESTES COM USUÁRIOS

A implementação do sistema desenvolvido seguiu procedimentos bem definidos e foram utilizadas técnicas eficientes e precisas, o que garantiu o bom desempenho e confiabilidade ao sistema. Para corroborar com as características apresentadas de desempenho, precisão e usabilidade, o sistema foi submetido à testes com usuários. Os testes realizados tinham, cada um, determinado objetivo de avaliação do sistema e permitiram verificar funcionalidades que podem ser utilizadas em sua real aplicação.

Os testes foram divididos por modo de utilização do sistema. Os testes 1, 2 e 3 avaliaram o modo *mouse* do sistema, especificamente a movimentação do cursor do mouse. Já os testes 4 e 5 avaliaram o modo *teclado* do sistema, especificamente a eficiência de acionamento de cliques e teclas com a face, olhos e boca. Por fim, foi realizada uma pesquisa de satisfação para avaliar de forma qualitativa a percepção que o usuário teve das funcionalidades do sistema. Assim, 50 usuários divididos em grupos, realizaram os seguintes testes:

Avaliação do Modo Mouse

1. Teste de movimentação do cursor do mouse
2. Teste comparativo entre sistemas
3. Teste de escrita em teclado virtual

Avaliação do Modo Teclado

4. Teste de acionamentos com olhos e boca
5. Teste de acionamentos com a face

Avaliação Qualitativa do Sistema

6. Pesquisa de satisfação

A figura 27 apresenta a divisão dos usuários em grupos para realizar os diferentes testes. Esse grupo de usuários foi formado por estudantes universitários, de diversas

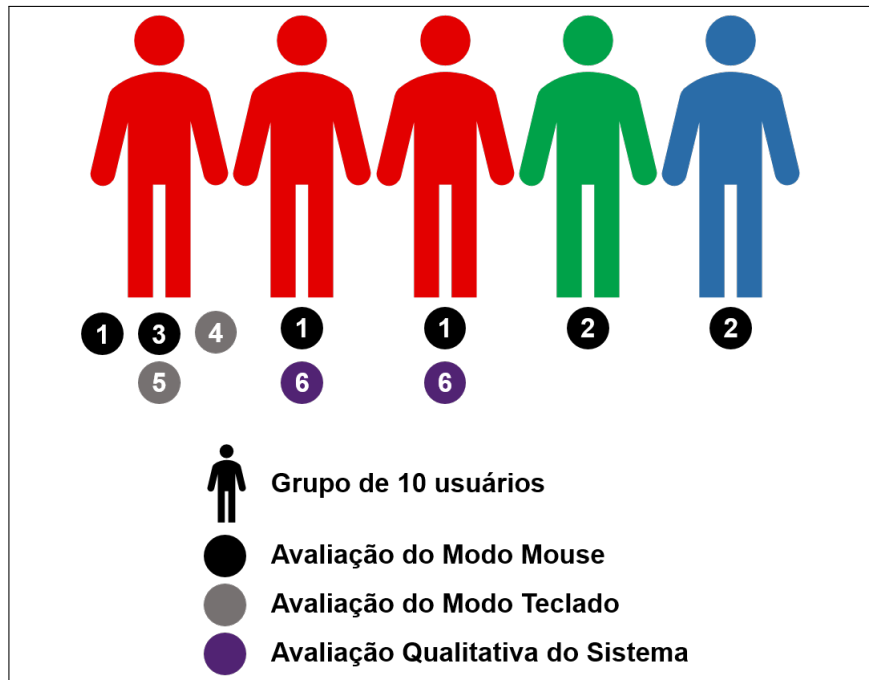


Figura 27: Divisão dos testes em grupos de usuários. 1) Teste de movimentação do cursor do mouse; 2) Teste comparativo entre sistemas; 3) Teste de escrita em teclado virtual; 4) Teste de acionamentos com olhos e boca; 5) Teste de acionamentos com a face; 6) Pesquisa de satisfação.

tonalidades de pele. 30 usuários realizaram o teste 1 (Teste de movimentação do cursor do mouse), devido sua estrutura. Desse grupo 10 usuários realizaram outros 3 testes, o teste 3 (Teste de escrita em teclado virtual), o teste 4 (Teste de acionamentos com olhos e boca) e o teste 5 (Teste de acionamentos com a face), para não sobrecarregar esses usuários a pesquisa de satisfação não foi realizada por eles. Assim, os outros 20 usuários realizaram o teste de satisfação após o teste 1. O teste 2 (Teste comparativo entre sistemas) foi realizados por 20 usuários, 10 para cada sistema comparado, não sendo aplicado para esses usuários a pesquisa de satisfação para evitar comparações entre sistemas, já que a pesquisa de satisfação teve como interesse apenas a avaliação do sistema desenvolvido.

O sistema foi parametrizado de forma igual à todos os usuários em todos os testes, e os recursos de *hardware* da máquina também foram os mesmos. Foi utilizado um *notebook* com processador *Intel Core i3-3110M dualcore* de 2.40Ghz, memória RAM de 8GB, sistema operacional *Windows 10* de 64 bits e *webcam* de 640x480 px de resolução. A seguir serão apresentados os objetivos, descrição e resultados dos testes de forma detalhada.

5.1 Teste de movimentação do cursor do mouse

A principal funcionalidade do sistema desenvolvido é a possibilidade do usuário de movimentar o cursor do mouse por meio dos movimentos faciais, assim essa funcionalidade foi avaliada para comparar seu desempenho. Nos trabalhos analisados, que tinham um objetivo similar ao proposto neste trabalho, diferentes testes foram realizados para avaliar essa funcionalidade, entretanto cada um tinha sua própria metodologia e procedimento de avaliação.

Bian et al. (2016) apresentaram uma metodologia padronizada de avaliar esse tipo de sistema, baseado na norma ISO/TS 9241-411. Essa norma define como a avaliação de dispositivos de apontamento deve ser realizada, determinando como o ambiente de teste deve ser estruturado e os cálculos que devem ser realizados.

O ambiente de teste é composto por regiões circulares, as quais o usuário deve posicionar o cursor do mouse, ao todo são 15 regiões organizadas em formato circular. Os parâmetros de avaliação são calculados em função da distância entre as regiões (D) e seu tamanho (W).

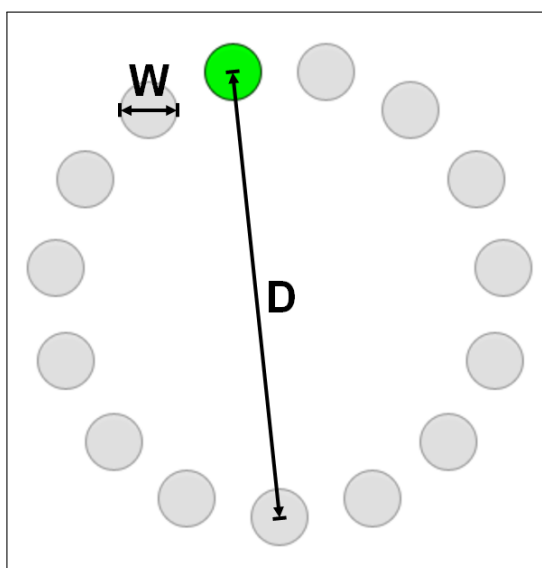


Figura 28: Modelo do ambiente de teste definido na norma ISO/TS 9241-411.

A figura 28 apresenta um modelo do ambiente de teste descrito. A partir das medidas

D e W é possível calcular o índice de dificuldade do ambiente, por meio da equação

$$ID = \log_2\left(\frac{D}{W} + 1\right) \quad (25)$$

Durante a realização do teste o sistema indicou ao usuário a sequência de cliques nas regiões do ambiente, mudando a cor da região para verde. O usuário deveria realizar movimentos opostos entre as regiões até clicar novamente na região inicial, como ilustrado na figura 29, essa movimentação foi armazenada e a rota percorrida pelo cursor do mouse foi apresentado ao final do teste para avaliar a suavidade de movimentação e a precisão nas regiões de clique.

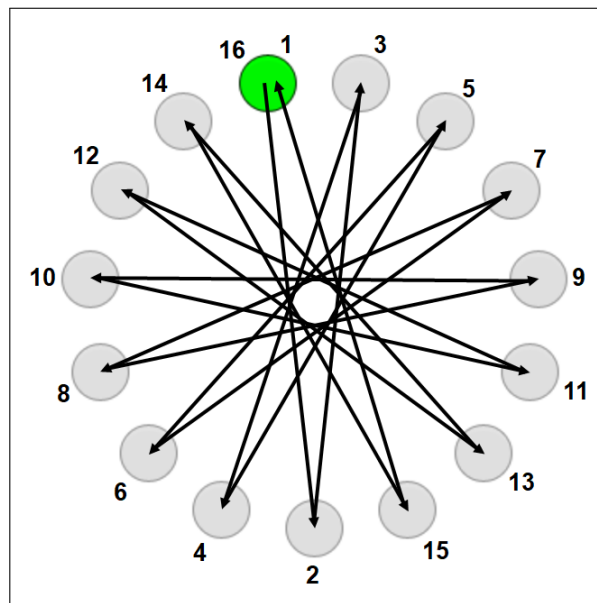


Figura 29: Sentido de movimentação realizado pelo cursor do mouse.

No decorrer do teste o sistema coletou o tempo de realização da tarefa, consequentemente o tempo médio de clique dos alvos (MT). Com essa medida de tempo foi possível calcular o indicador de desempenho *Throughput* (TP), dado em *bits/s*, esse indicador mensura o quanto de informação o usuário, por meio do sistema desenvolvido, conseguiu inserir no sistema operacional. Quanto maior o valor de *throughput* maior o volume de informação que o usuário consegue inserir no sistema operacional. O cálculo do *throughput* é encontrado com a equação

$$TP = \frac{ID}{MT} \quad (26)$$

Apesar da norma definir o ambiente de teste e os cálculos dos indicadores de desempenho, Soukoreff e MacKenzie (2004) apontaram alguns pontos de atenção na aplicação de testes baseados na norma ISO/TS 9241-411. Para os autores as medidas de tempo definidas pela norma, que desconsideram o tempo de clique na região alvo, produzem medições baseadas apenas na velocidade de movimentação do cursor, eliminando o fator precisão. Soukoreff e MacKenzie (2004) recomendaram que sejam realizadas análises nos testes executados, eliminando os testes que apresentarem comportamento apenas baseado em velocidade. Os testes realizados com o sistema desenvolvido, para evitar que apenas o fator velocidade fosse medido, considerou para medir o tempo médio de movimentação o tempo de movimentação e clique de um alvo para o outro nos ambientes de teste. Com essa medida foi possível considerar o fator precisão do sistema, garantindo a confiabilidade dos testes realizados.

Além das métricas relacionadas com o cursor do mouse, o teste foi utilizado para verificar a curva de aprendizado do sistema, assim 30 usuários realizaram 9 rodadas de teste cada um em 3 ambientes diferentes, sendo 3 rodadas para cada ambiente. A cada 10 usuários a ordem de execução dos ambientes era mudada, possibilitando avaliar a curva de aprendizado em diferentes ordens de execução. A variação da ordem dos ambientes de teste possibilitou medir o *throughput* real do sistema, já que a média de *throughput* dos ambientes pôde ser calculada considerando a utilização do ambiente em todas as ordens de uso. A tabela 1 apresenta os ambientes utilizados para a elaboração do teste. A figura 30 apresenta o resultado do teste, onde pode ser verificado a média do tempo de realização das rodadas e dos valores de *throughput*. A figura 31 apresenta o gráfico dos valores médios de *throughput* de cada ambiente, considerando sua execução em todas as ordens. A figura 32 apresenta o caminho percorrido pelo cursor do mouse, bem como o ponto de clique durante a realização do teste. Já a tabela 2 apresenta os valores numéricos dos gráficos da figura 30 e do gráfico da figura 31.

Tabela 1: Ambientes utilizados no teste de eficiência de movimentação do cursor do mouse

Ambiente	D	W	ID
1	534	76	3.00
2	534	57	3.37
3	305	57	2.67

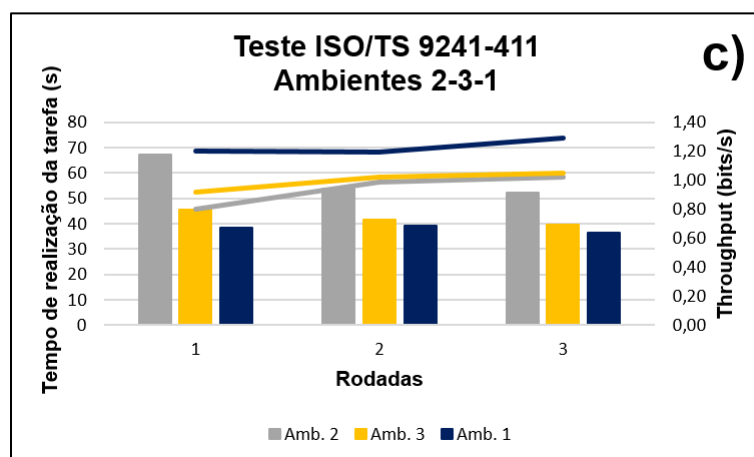
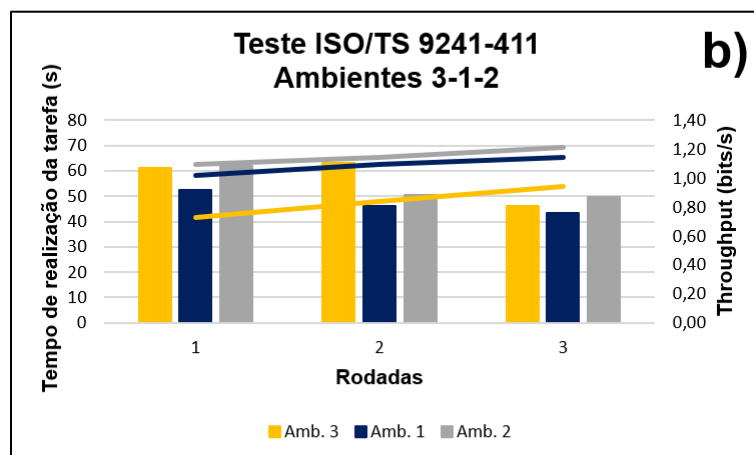
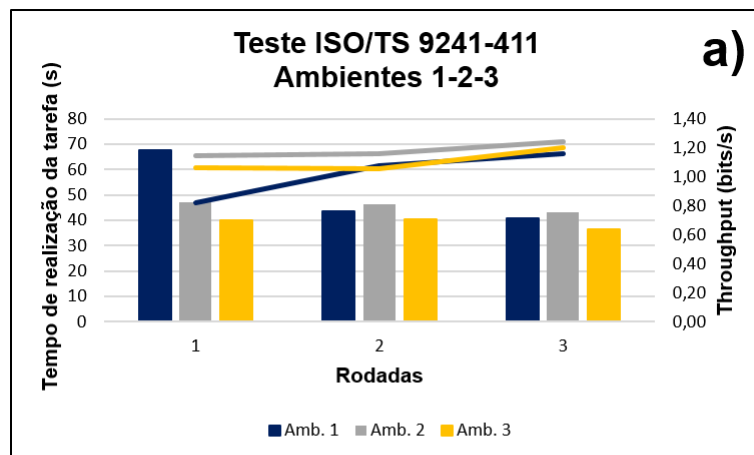


Figura 30: Resultados gráficos dos Testes ISO/TS 9241-411. a) Sequência de teste Amb. 1-2-3; b) Sequência de teste Amb. 3-1-2; c) Sequência de teste Amb. 2-3-1.

Tabela 2: Resultados numéricos do teste ISO/TS 9241-411

Sequência	Rodada	Ambiente								
		1			2			3		
		TCT	MT	TP	TCT	MT	TP	TCT	MT	TP
1-2-3	1	67,48	4,50	0,82	47,00	3,13	1,14	40,10	2,68	1,06
	2	43,68	2,91	1,07	46,24	3,08	1,16	40,34	2,69	1,06
	3	40,82	2,72	1,16	43,25	2,88	1,25	36,34	2,42	1,20
3-1-2	1	52,33	3,49	1,02	61,74	4,12	1,10	61,08	4,07	0,73
	2	46,16	3,08	1,09	50,59	3,37	1,14	62,97	4,20	0,84
	3	43,41	2,90	1,14	49,70	3,31	1,21	46,21	3,08	0,94
2-3-1	1	38,28	2,55	1,20	67,43	4,50	0,80	45,68	3,05	0,92
	2	39,33	2,62	1,19	53,92	3,60	0,99	41,69	2,78	1,02
	3	36,40	2,43	1,29	52,21	3,48	1,02	39,83	2,66	1,05
Média		45,32	3,02	1,11	52,45	3,50	1,09	46,03	3,07	0,98

A análise dos gráficos apresentados na figura 30 pode determinar algumas características do sistema desenvolvido. Em todos os gráficos a maior acentuação de queda do tempo de realização da tarefa e de aumento do *throughput* acontece durante a primeira rodada, em média aproximadamente 20%. Esse comportamento indica a rapidez de aprendizado do sistema desenvolvido, uma vez que o usuário necessita pouco tempo para se familiarizar e utiliza-lo de maneira efetiva. Independente da ordem de dificuldade dos ambientes de teste, os usuários em média conseguem finalizar o teste com valores de *throughput* próximos à $1,20 \text{ bits/s}$, indicando que os usuários conseguiram completar a tarefa com a mesma eficiência, não importando a dificuldade. Assim a ordem de realização dos testes entre os ambientes não influencia na aprendizagem do usuário, nem em seu desempenho em cumprir a tarefa. Entretanto, analisando a curva de *throughput* dos gráficos apresentados pode ser observado que as curvas do gráfico (Figura 30 b) possui comportamento constante de crescimento, sem as passagens abruptas vistos nos gráficos (Figura 30 a) e (Figura 30 c). No gráfico (Figura 30 b) também pode ser constatado que o tempo de realização da tarefa é o menor na primeira iteração do usuário com o sistema. Esses 2 fatores podem fazer com que pareça de forma mais aparente ao usuário que inicia pelo ambiente 3 que o sistema é fácil de utilizar, do que o usuário que inicia pelos outros

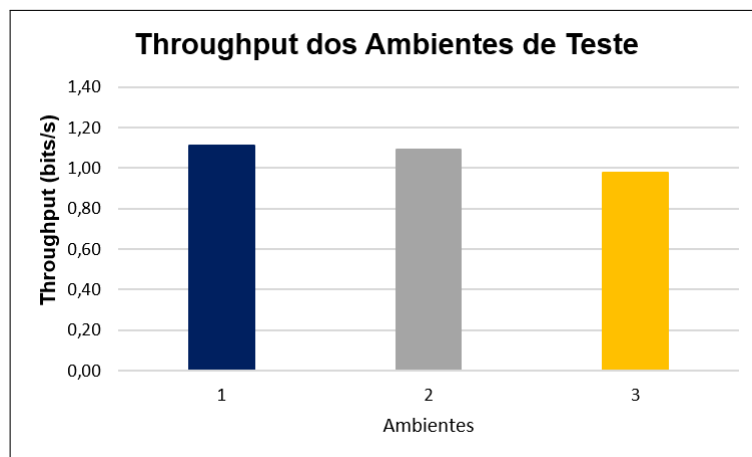


Figura 31: Média dos valores de *throughput* dos ambientes de teste.

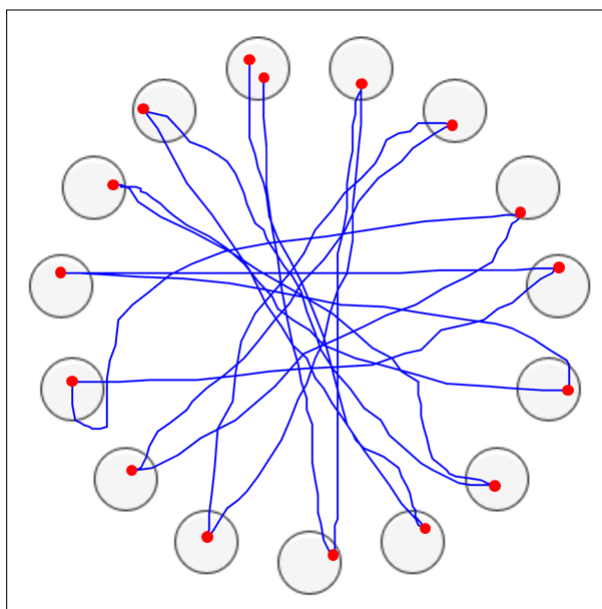


Figura 32: Caminho percorrido pela cursor do mouse no Amb. 1.

ambientes de teste.

Analisando a figura 31, que indica os valores médios de *throughput* dos ambientes de teste, foi possível identificar a faixa de *throughput* do sistema, que varia de 1,00 *bit/s* a 1,20 *bits/s*. O caminho percorrido pelo mouse, apresentado na figura 32, mostra que o índice calculado de *throughput* do sistema é acompanhado pela suavidade de movimentação e precisão de posicionamento, uma vez que o caminho se mostra contínuo e que nas regiões de clique são poucos os ajustes de posicionamento.

A realização desse teste, além das medidas de eficiência apresentadas, possibilitou identificar 2 pontos de melhorias no sistema desenvolvido e que já foram implementados.

O primeiro foi o incremento de teclas que poderiam ser acionadas utilizando o sistema. Em um primeiro momento eram disponibilizados 45 itens, por sugestões dos usuários esse número foi aumentado para 94 itens, incluindo opções de digitação de teclas para controle de *media player*, *browser* de internet, entre outros. A segunda melhoria foi a inclusão da opção de regular a sensibilidade de abertura da boca, ou seja, o quanto o usuário deveria abrir a boca para que sua funcionalidade fosse executada. Com a parametrização fixa desse recurso, alguns usuários constataram muita sensibilidade, não possibilitando que a boca ficasse aberta nem para respirar, outros usuários constataram o oposto, a necessidade de abrir muito a boca para executar o comando, o que poderia causar fadiga. Com a parametrização desse recurso o usuário pode adequar o sistema da melhor forma possível para suas próprias necessidades.

5.2 Teste comparativo entre sistemas

O teste baseado na norma ISO/TS 9241-411, permitiu medir a eficiência de movimentação do cursor mouse e seus cliques. Obter de forma isolada o valor de *throughput*, sem nenhuma forma de comparação, torna difícil mensurar a qualidade desse valor. Assim foi necessário realizar um teste comparativo, onde os usuários utilizaram sistemas diferentes em um mesmo ambiente de teste. Os sistemas que foram comparadas ao sistema proposto foram o *CameraMouse*, proposto por Betke, Gips e Fleming (2002), e o *HeadMouse*, proposto por Pallejà et al. (2011). Esses sistemas estão disponíveis para download e utilizam os mesmos recursos que o sistema desenvolvido, apenas um computador e uma *webcam*.

Para a realização desse teste foram convidados 20 usuários, 10 para cada sistema. O ambiente de teste 1 do teste anterior foi escolhido, onde a distância entre os alvos (D) é de 534 *px* e o tamanho do alvo (W) é de 76 *px*. Para eliminar o fator aprendizado, 5 usuários iniciaram o teste com o sistema proposto desenvolvido, em seguida 5 usuários iniciaram com o sistema comparado. Durante o teste os usuários não tinham conhecimento dos sistemas que eram utilizadas. O sistema desenvolvido foi configurado da mesma forma que no teste anterior. O sistema *HeadMouse*, foi configurado de forma padrão, utilizando a abertura da boca para executar os cliques do mouse, nesse sistema foi necessário realizar uma calibração com a face do usuário antes de sua utilização. O sistema *CameraMouse*

também necessitou de calibração, onde o nariz do usuário foi escolhido para movimentar o cursor do mouse, o modo de clique *dwell* foi configurado para 0,5 s, como recomendado pela norma. A figura 33 apresenta o resultado do teste comparativo.

Os gráficos (Figura 33 a) e (Figura 33 b) apresentam de forma isolada a comparação entre os sistemas, produzida com 10 usuários. As barras da esquerda representam o *throughput* médio da utilização pelos usuários que começaram o teste pelo sistema proposto, já as barras da direita indicam a mesma medida mas dos usuários que utilizaram o sistema proposto por último. Para um melhor entendimento dos valores de *throughput*, uma medida de eficiência externa ajuda a posicionar esse valor em sistemas similares ao proposto. Bian et al. (2016) indicaram que a medida de *throughput* para o mouse tradicional varia de 4,00 *bit/s* a aproximadamente 4,50 *bits/s*. A comparação do sistema proposto com o *CameraMouse* (Figura 33 a), apresentou resultados similares, já que os dois sistemas variam o *throughput* entre 1,00 *bit/s* e 1,20 *bits/s*. O sistema *CameraMouse*, por utilizar a função de cliques em modo *dwell*, onde um clique é executado a cada 0,5s, pode acumular valores baseados em velocidade, desconsiderando precisão, entretanto não foi excluído nenhuma medida do teste. Durante o teste com o sistema *CameraMouse* por 2 vezes, com 2 usuários diferentes, a rastreabilidade foi perdida, sendo necessário recalibrar o sistema para continuar o teste. A comparação do sistema proposto com o *HeadMouse* (Figura 33 b), mostra que o sistema proposto apresenta eficiência muito superior a *HeadMouse*, praticamente o dobro do valor de *throughput*, independente da ordem de utilização dos sistemas. O grande problema apresentado pelo sistema *HeadMouse* foi o baixo desempenho de processamento, muito enfatizado na construção do sistema proposto, o que dificultou ao usuário ter um controle preciso do cursor do mouse e a sensibilidade adequada para ativar os cliques com a abertura da boca. Por vezes o usuário passava do alvo, por o sistema estar atrasado em seu processamento, ou seja, havia um *delay* que onerava a precisão de posicionamento, no caso dos cliques várias tentativas eram realizadas pelos usuários, que muitas vezes alterava o posicionamento do cursor para fora da região de clique, o que acabou atrasando ainda mais o cumprimento da tarefa. O gráfico (Figura 33 c) apresenta uma comparação geral com as medidas dos 3 sistemas, a medida do sistema proposto apresentada nesse gráfico é a média entre os 20 usuários que participaram do teste comparativo.

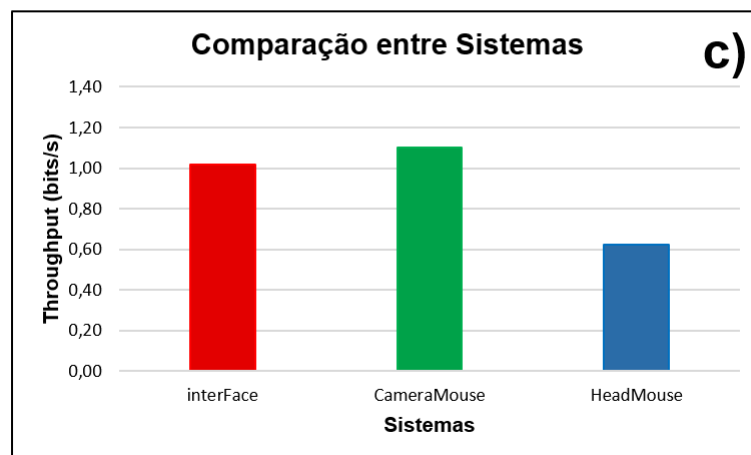
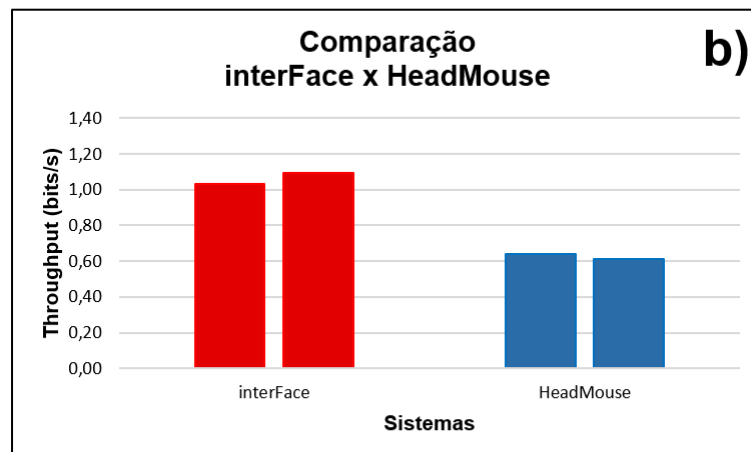
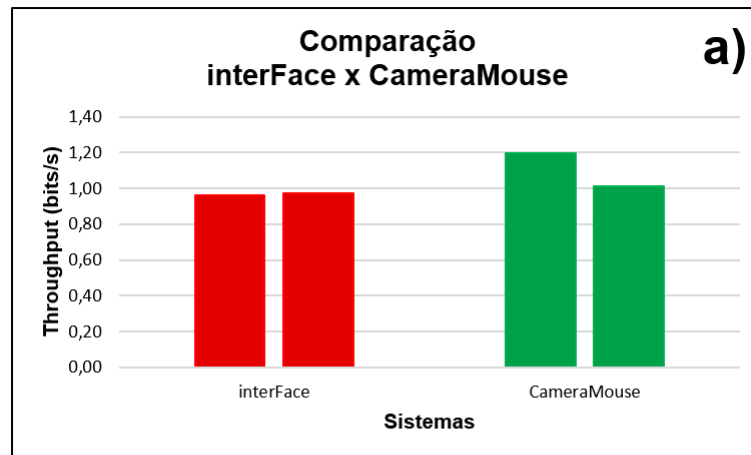


Figura 33: Teste comparativo entre sistemas. a) Comparação interFace (proposto) X *CameraMouse*; b) Comparação interFace (proposto) X *HeadMouse*; c) Comparação das médias de *throughput*.

5.3 Teste de escrita em teclado virtual

Uma das principais atividades realizadas durante a utilização de computadores é a redação de textos. Navegar na internet, escrever *e-mails*, enviar mensagens em redes sociais, são atividades cotidianas e muito comuns dos usuários, portanto essa é uma função que o sistema desenvolvido deveria atender, possibilitando aos usuários utilizar esses recursos.

Para avaliar o sistema quanto ao cumprimento dessa funcionalidade foi realizado um teste de escrita. Por meio de um teclado virtual, apresentado na figura 34, 10 usuários foram solicitados a escrever 3 frases, sendo medido o tempo necessário para escreve-las. Essa escrita deveria ser realizada com o sistema no modo *mouse*, ou seja, por meio da movimentação do cursor do mouse e cliques nas teclas do teclado virtual. As teclas desse teclado virtual tinham dimensões de aproximadamente 30x30 *px*, e durante a realização do teste o usuário não precisaria digitar caracteres especiais como acentuação e espaços, caso algum caractere fosse digitado errado não era necessário apaga-lo. As 3 frases solicitadas para escrita foram "Bom Dia", "Visão Computacional" e "Vamos assistir ao jogo". Para a escolha dessas frases foi considerado apenas a quantidade de caracteres que deveriam ser digitados, a frase "Bom Dia" possui 6 caracteres e serviu como um aprendizado da tarefa, as outras frases possuem 3 vezes esse valor, 18 e 19 caracteres respectivamente.



Figura 34: Teclado virtual utilizado no teste de escrita.

Ao final da realização do teste, os usuários apresentaram em média 30 s para a escrita da frase "Bom Dia", e 80 s para as frases "Visão Computacional" e "Vamos assistir ao jogo", como pode ser visto na figura 35 e na tabela 3. Assim, a taxa média de escrita foi

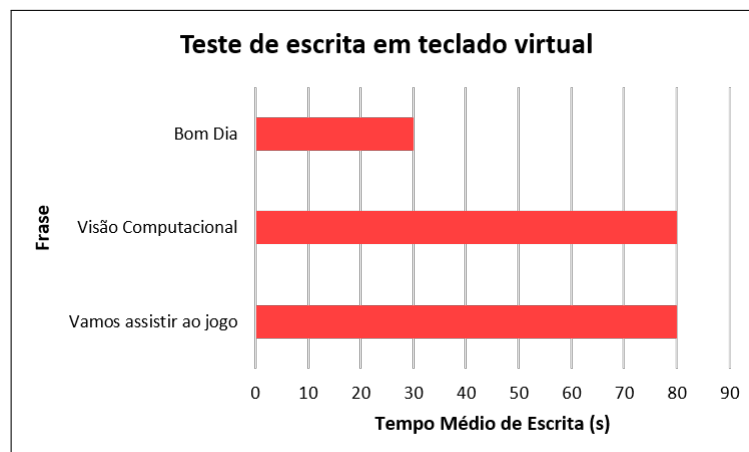


Figura 35: Resultado do teste de escrita.

Tabela 3: Resultado do teste de escrita em teclado virtual

Frase	Caracteres	Tempo (s)
Bom Dia	6	30
Visão Computacional	18	80
Vamos assistir ao jogo	19	80

de 1 caractere a cada 5 s. Esse tempo pode ser reduzido com tempo de uso do sistema, mas mesmo com essa taxa de escrita, redigir expressões curtas para pesquisas na internet ou mensagens em redes sociais não exige muito tempo, entretanto para redações mais complexas o usuário demandaria mais tempo. Essa funcionalidade do sistema de permitir a utilização de teclados virtuais durante sua utilização no modo *mouse*, supre a limitação de 7 acionamentos de teclas do modo *teclado*. Para essa funcionalidade o usuário poderia utilizar sistemas de reconhecimento de voz, o que agilizaria a atividade.

5.4 Teste de acionamentos com olhos e boca

Os testes apresentados anteriormente apresentaram a avaliação de funcionalidades relacionadas ao modo *mouse* do sistema desenvolvido. As funcionalidades do modo *teclado* também foram avaliadas. O teste de eficiência de acionamento de cliques e teclas com olhos e boca teve como objetivo verificar se os usuários conseguiam utilizar esses recursos e fazer uma comparação entre os cliques acionados com a boca e os realizados com os olhos, uma vez que são executados por meio de técnicas diferentes.

O ambiente de teste para realizar essa avaliação foi composto de apenas 1 elemento, um botão medindo 200x200 *px* o qual o usuário realizou cliques utilizando a boca e os olhos. O próprio ambiente de teste indicava ao usuário como o clique deveria ser realizado e contava quantos cliques foram executados em um intervalo de 20 s. 10 usuários realizaram esse teste.

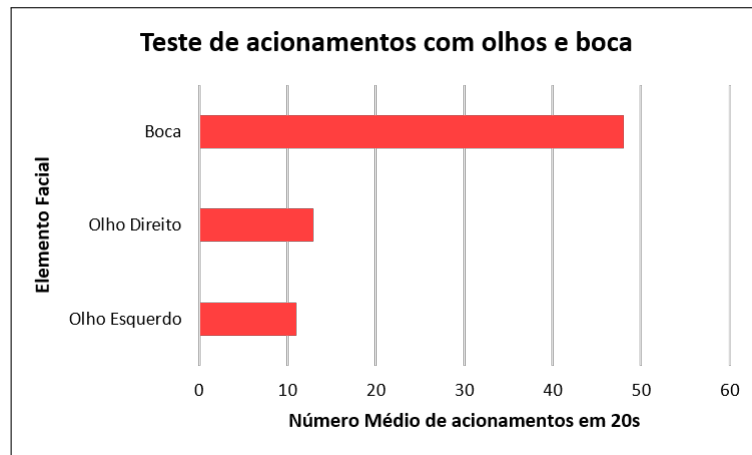


Figura 36: Resultado do teste de acionamentos com olhos e boca.

Tabela 4: Resultado do teste de acionamentos com olhos e boca

Elemento Facial	Acionamentos em 20s
Olho Esquerdo	11
Olho Direito	13
Boca	48

Pode ser observado na figura 36 e na tabela 4, que ao final do teste, em média, os usuários conseguiram acionar o botão em 20 s, 48 vezes com a boca, 13 vezes com o olho direito e 11 vezes com o olho esquerdo. Assim os acionamentos com a boca tiveram uma frequência de 2,4 acionamentos por segundo, valor 4 vezes maior que os acionamentos executados com os olhos, que apresentaram frequência de 0,8 acionamentos por segundo. Essa maior eficiência de execução de comandos por meio da abertura da boca, em relação aos olhos, acontece devido à técnica envolvida nessa execução. A boca utiliza a simples diferença de distância entre pontos, o que não foi possível de realizar na região dos olhos, que devem ser detectados 2 vezes seguidas para determinar sua não abertura.

5.5 Teste de acionamentos com a face

De maneira análoga ao teste realizados para avaliar o acionamento de cliques e teclas com os olhos e boca, o teste de eficiência dessa funcionalidade por meio dos movimentos da face foi realizado. 10 usuários foram submetidos a executar o acionamento de teclas direcionando a face para cima, para baixo, para esquerda ou para a direita.

O ambiente de teste foi composto de 4 indicadores, em formato de cruz, que mudavam da cor cinza para a cor verde, mostrando ao usuário em qual direção ele deveria movimentar a face. Para que o indicador verde fosse acionado, sistemicamente, uma tecla deveria ser digitada, assim foram configuradas as teclas A, D, S e W no sistema proposto para serem acionadas com o movimento da face. Uma sequência de 20 indicações foi elaborada, 5 para cada direção. A figura 37 ilustra o ambiente de teste utilizado. Nesse teste foi medido o tempo de realização da tarefa, com a qual foi possível determinar a frequência de acionamento e avaliar se esse recurso poderia ser utilizado em aplicações reais.

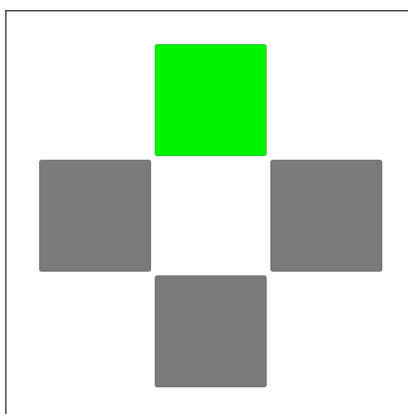


Figura 37: Ambiente de teste de acionamento de teclas com a face.

Os usuários conseguiram em média realizar a tarefa em 23 s, o que representa uma taxa média de acionamento de 1,15 teclas por segundo. Muitos usuários conseguiram realizar em uma frequência abaixo de 1 acionamento por segundo. Essa característica permite que inúmeras aplicações sejam utilizadas por meio desse recurso, como jogos simples e tabelas de solicitação. Uma tabela de solicitação é uma aplicação voltada à pessoas com deficiência, onde diversos sentimentos e desejos como são representados em figuras, como berber água, ver televisão, entre outros, navegáveis por teclas. O usuário então navega entre essas figuras por meio de teclas, posicionando um indicador sobre uma dessas figuras. Usuários com restrições mais severas de movimentação do pescoço,

impossibilitadas de utilizar o modo *mouse* do sistema proposto, poderiam utilizar o modo *teclado* para fazer uso de tabelas de solicitação.

Esse teste mostrou novamente eficiência de determinar a direção da face pela posição dos pontos faciais, como realizado no posicionamento do cursor do mouse, e na verificação de abertura da boca. Podendo ser um recurso de utilização em diversas outras aplicações.

5.6 Pesquisa de satisfação de usabilidade

Os testes apresentados anteriormente tiveram objetivos quantitativos, a fim de avaliar aspectos técnicos do sistema desenvolvido. Entretanto essas medidas não apresentam insumos para avaliar características como facilidade de utilização do sistema, utilidade de recursos implementados, conforto durante a utilização ou perspectivas de uso. Assim uma pesquisa de satisfação de usabilidade foi conduzida com 20 usuários, os quais realizaram o teste de eficiência de movimentação do cursor do mouse, baseado na norma ISO/TS 9241-411. Após a realização desse primeiro teste, o sistema foi detalhadamente descrito, as funções e opções do sistema foram explicadas e o público alvo do sistema foi descrito, em seguida os usuários responderam à pesquisa.

A pesquisa contou com perguntas que envolviam 8 temas de interesse, com os quais seria possível mapear a percepção do usuário sobre o sistema, perspectivas de uso e possíveis melhorias:

1. Perfil do Usuário
2. Tela de Configuração
3. Opções do Sistema
4. Funcionamento do Sistema
5. Facilidade de Uso e Aprendizado
6. Conforto de Utilização
7. Aplicação do Sistema Desenvolvido
8. Satisfação

Com exceção dos temas de Perfil do Usuário e Satisfação, que continham perguntas diretas como sim ou não, as respostas deveriam ser classificadas em 5 níveis pelo usuário,

demonstrando sua concordância ou discordância para uma dada afirmação: 1) Discorda Totalmente; 2) Discorda; 3) Não Concorda nem Discorda; 4) Concorda; 5) Concorda Totalmente. O questionário completo está descrito no Apêndice A – Questionário da Pesquisa de Satisfação

5.6.1 Perfil do Usuário

O mapeamento de perfil dos usuários possibilitou identificar informações como sexo, idade e experiência com informática. Pode ser observado nas figuras 38 e 39, que o público foi composto de 80% usuários homens, logo 20% usuários mulheres. 90% dos usuários eram jovens com idade entre 17 e 30 anos. 85% utilizam *e-mail* diariamente, 75% utilizam redes sociais diariamente e todos utilizam internet diariamente.

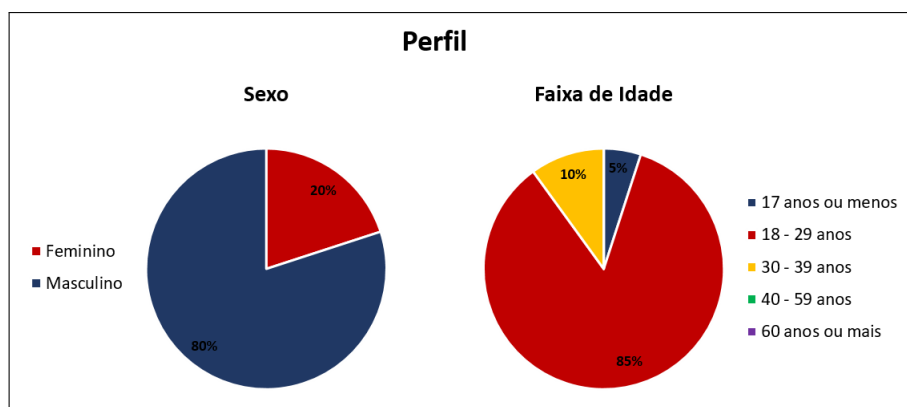


Figura 38: Pesquisa de Satisfação - Perfil.

5.6.2 Tela de Configuração

A avaliação da tela de configuração teve como objetivo avaliar os elementos presentes na tela de configuração do sistema, como tamanho de fontes e a localização das opções do sistema. A figura 40 mostra que 90% dos usuários concordaram que os tamanhos das fontes estão adequados e todos concordam que a localização das opções do sistema estão adequados. Tanto a figura 40 quanto as similares, que demonstram as opiniões dos usuários, agrupam as respostas "Discorda Totalmente" e "Discorda" em Discorda e as respostas "Concorda" e "Concorda Totalmente" em Concorda.



Figura 39: Pesquisa de Satisfação - Experiência em Informática.

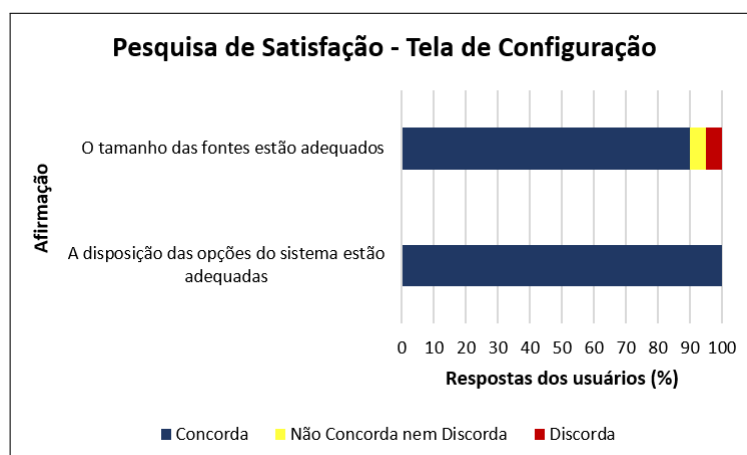


Figura 40: Pesquisa de Satisfação - Tela de Configuração.

5.6.3 Opções do Sistema

Nessa seção foi perguntado aos usuários sobre a utilidade das opções do sistema e se essas opções tornam o sistema flexível para realizar diversas tarefas. A figura 41 apresenta a opinião dos usuários sobre as opções do sistema. Todos os usuários concordaram que determinar o que o movimento da face, dos olhos e da boca realizam no sistema operacional é útil. 95% concordaram que ativar e desativar as funcionalidades acionadas pelos olhos

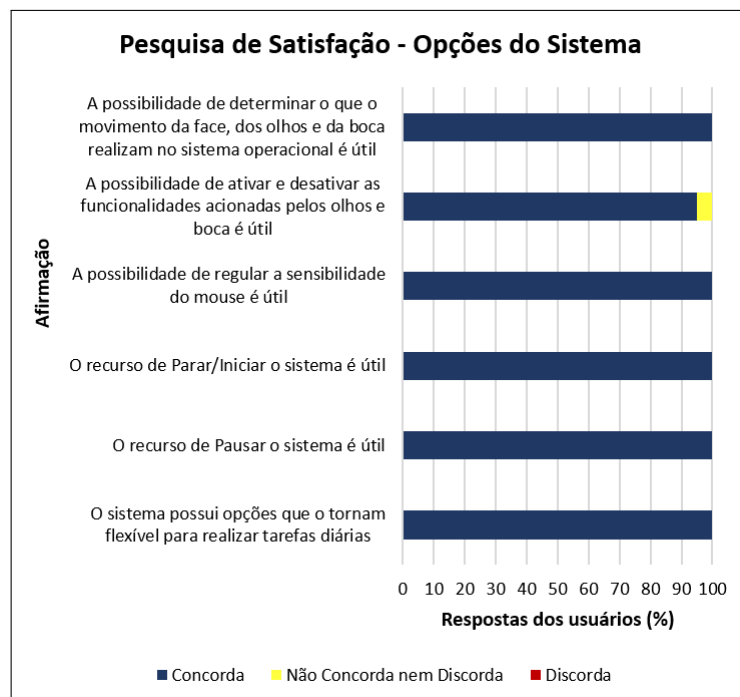


Figura 41: Pesquisa de Satisfação - Opções do Sistema.

e boca é útil. Todos concordaram que regular a sensibilidade do mouse é útil. Todos concordaram que Parar/Iniciar e Pausar o sistema é útil. 95% indicaram que as opções do sistema, o tornam flexível para realizar tarefas computacionais diárias.

5.6.4 Funcionamento do Sistema

As perguntas relacionadas ao funcionamento do sistema visaram avaliar se o sistema tem boa precisão e velocidade de resposta. A figura 42 mostra que 80% dos usuários apontaram que o movimento do cursor do mouse por meio dos movimentos da face é suave, 85% indicaram que esse movimento e o posicionamento do cursor são precisos. 75% dos usuários indicaram que o clique por meio da abertura da boca é preciso, avaliação que levou a melhorar essa funcionalidade. Para alguns usuários o clique com a boca estava muito sensível, causado pela variação do tamanho do rosto, impedindo até mesmo que o usuário pudesse respirar pela boca. Assim foi desenvolvido a opção de regular a sensibilidade de abertura da boca possibilitando esse ajuste para cada usuário. Por fim, todos os usuários indicaram que a velocidade de resposta do sistema aos seus comandos foi boa.

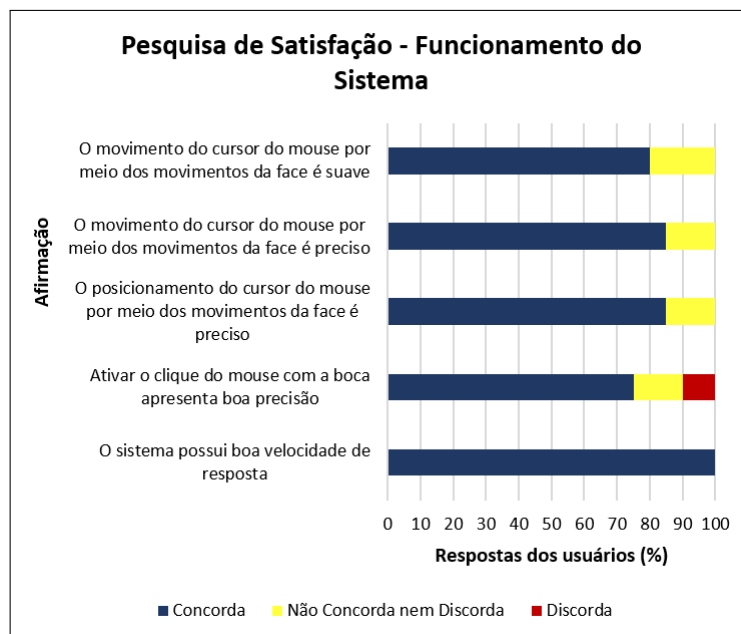


Figura 42: Pesquisa de Satisfação - Funcionamento do Sistema.

5.6.5 Facilidade de Uso e Aprendizado

Nessa seção foi perguntado aos usuários percepções de facilidade e de aprendizado, bem como se o sistema poderia ser utilizado em algumas funções ou atividades. A figura 43 apresenta as respostas dos usuários. Todos os usuários concordaram que o sistema é de fácil aprendizado. 95% disseram que o sistema é fácil de usar. 95% concordaram que o sistema permite selecionar de forma fácil ícones grandes em tela. 85% concordaram que o sistema permite selecionar de forma fácil ícones pequenos em tela. 95% dos usuários concordaram que caso utilizassem o sistema cotidianamente conseguiriam realizar muitas atividades, ou seja, concordaram que o sistema precisa de adaptação e treinamento, mas que após algum tempo de uso sua utilização é mais natural e eficiente.

5.6.6 Conforto de Utilização

Esse grupo de perguntas buscou avaliar a percepção de cansaço, conforto e tempo de utilização, as respostas são apresentadas na figura 44. 90% dos usuários concordaram que a utilização do sistema é confortável. 75% concordaram que poderiam passar muito tempo utilizando o sistema. 80% indicaram que não sentiram cansaço durante a utilização do sistema. Os dois últimos itens de tempo de uso e cansaço, estão muito relacionados com o estado do usuário durante o teste, no sentido de pressão, relaxamento ou objetivo

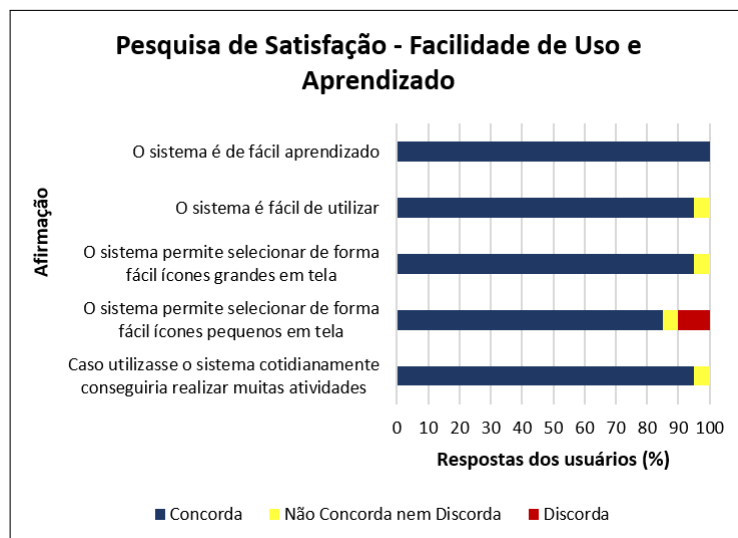


Figura 43: Pesquisa de Satisfação - Facilidade de Uso e Aprendizado.

da tarefa, usuários que por algum motivo se sentiram pressionados a cumprir a tarefa, ou ficaram impacientes para terminar a tarefa poderiam estar pré-dispostos a se sentir mais cansado ao fim do teste.

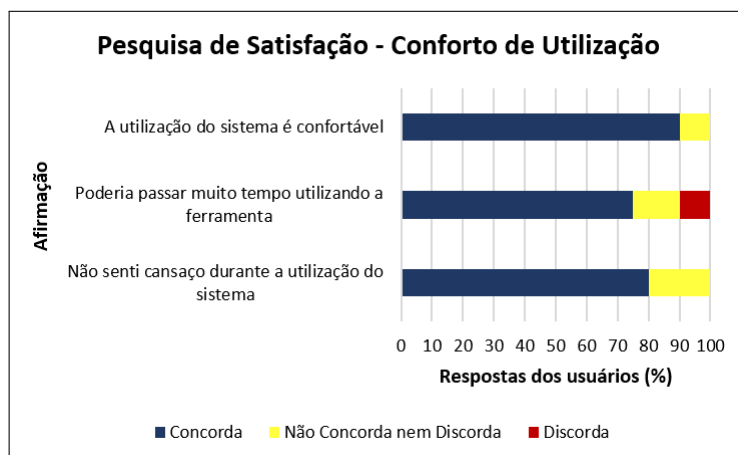


Figura 44: Pesquisa de Satisfação - Conforto de Utilização.

5.6.7 Aplicação do Sistema Desenvolvido

As perguntas relacionadas à aplicação do sistema desenvolvido tiveram o objetivo de mensurar possíveis públicos e atividades que podem ser realizadas.

A figura 45 apresenta a distribuição das respostas dos usuários. No que diz respeito a atividades, 85% dos usuários concordaram que o sistema pode ser utilizado para navegar na internet, 75% concordaram em sua utilização nas redes sociais e 85% concordaram que

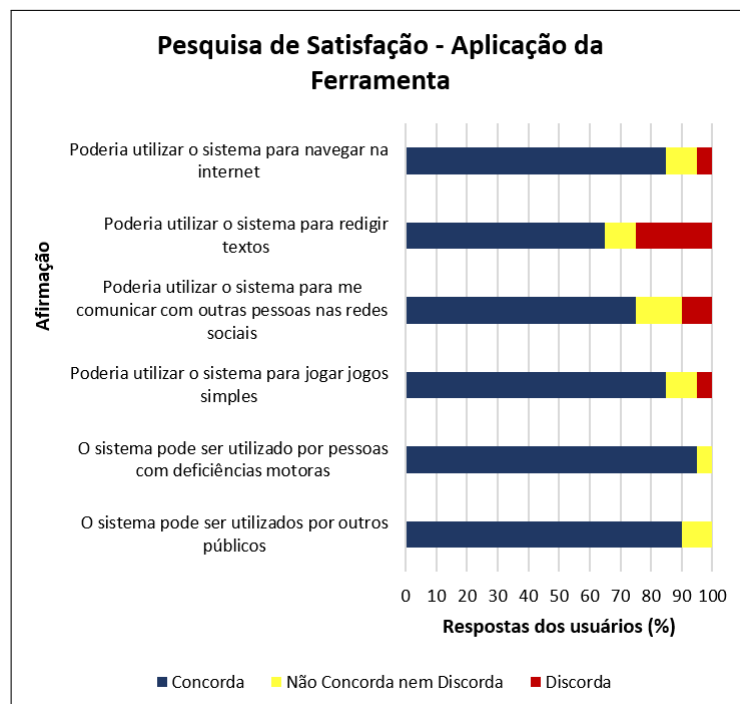


Figura 45: Pesquisa de Satisfação - Aplicação do Sistema Desenvolvido.

o sistema pode ser utilizado para jogar jogos simples. A utilização do sistema para redigir textos foi o item com o qual os usuários menos concordaram que poderia ser realizado com o sistema desenvolvido, apenas 65%. Esse resultado está relacionado, além desse grupo de usuários não terem realizado a tarefa de escrever em teclado virtual e terem que imaginar a possibilidade de executar essa tarefa, à área restrita que o usuário teria que controlar o cursor do mouse para poder, por exemplo, clicar nas teclas de um teclado virtual. Essa funcionalidade apenas seria possível após um tempo maior de treinamento do sistema pelo usuário. Quanto ao público alvo, 95% concordaram que o sistema pode ser utilizado por pessoas com deficiência e 90% concordaram que poderia ser utilizado por outros públicos.

5.6.8 Satisfação

Nessa seção foi perguntado aos usuários sobre perspectivas de uso futuro do sistema proposto e foi solicitado comentários livres sobre o sistema. Todos os usuários disseram que utilizariam o sistema novamente e que indicariam para amigos. De maneira geral os usuários tiveram uma boa percepção do sistema, como pode ser visto na figura 46 que ilustra a visão geral da satisfação dos usuários com o sistema proposto. Nessa figura,

pode ser observado a consolidação da satisfação dos usuários por item avaliado, onde a porcentagem indica a concordância que determinado item está adequado e satisfaz as expectativas dos usuários. Nos comentários ao final do questionário, os usuários indicaram que o sistema desenvolvido pode ajudar muitas pessoas com dificuldades motoras a utilizarem o computador e que é um sistema de fácil aprendizado. Alguns usuários comentaram que após certo tempo de aprendizado muitas aplicações podem ser utilizadas ampliando o alcance de uso.

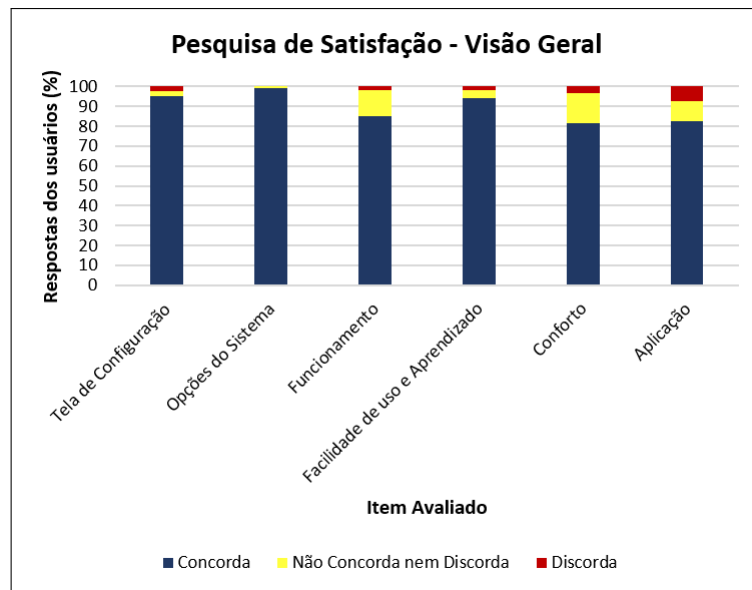


Figura 46: Pesquisa de Satisfação - Visão Geral.

6 CONCLUSÃO

O presente trabalho teve por objetivo desenvolver um sistema que possibilita a interação com o computador apenas utilizando os movimentos faciais do usuário. Seu desenvolvimento envolveu diversas técnicas de visão computacional, sendo sua utilização intuitiva e natural. Entre outras aplicações, essa forma de interação humano-computador pode ser utilizada por pessoas com deficiência motora como uma tecnologia assistiva.

O sistema desenvolvido foi construído utilizando técnicas e praticas eficientes, analisadas em trabalhos similares, bem como novas técnicas desenvolvidas e testadas. A detecção da face do usuário, realizada utilizando a técnica proposta por Viola e Jones (2001), foi precedida por pré-processamentos para reduzir a área de busca de face por meio de segmentação de pele. A detecção dos pontos faciais, utilizando a técnica proposta por Kazemi e Sullivan (2014), possibilitou que a movimentação do cursor do mouse, a determinação de abertura da boca e a posição dos olhos, pudessem ser realizada pela interpretação do posicionamento desses pontos. A implementação de uma função de transferência própria, possibilitou a movimentação do cursor de forma suave e precisa. A tela de configuração do sistema apresenta diversas opções, tornando-o flexível tanto do ponto de vista de usuário quanto de aplicação. Essa configuração, em casos onde o usuário possui deficiências graves como tetraplegia, deve ser realizada por outra pessoa, geralmente alguém que acompanha o usuário em suas atividades cotidianas. Todos esses elementos contribuíram para o desenvolvimento de uma ferramenta com real utilização e desempenho elevado, com processamento de 100 *fps*.

A realização de diferentes tipos de teste permitiram visualizar se os objetivos do sistema desenvolvido foram atingidos e se suas funcionalidades apresentam o desempenho esperado. O teste de eficiência de movimentação do cursor do mouse, baseado na norma ISO/TS 9241-411, mostrou que o sistema permite aos usuários movimentar o cursor e realizar os cliques do mouse de forma suave e precisa com *throughput* de 1,20 *bits/s*, sendo ainda um sistema de rápido aprendizado. Os tempos medidos e os índices calculados nesse teste provaram que o sistema desenvolvido possui desempenho igual ou superior a outras ferramentas, e não possui inúmeras desvantagens apresentadas nessas ferramentas. Os testes de escrita e de digitação com os movimentos da face, olhos e boca, comprovou que essas funcionalidades podem ser desempenhadas em aplicações reais por meio do

sistema desenvolvido, sendo possível realizar 1,15 acionamentos com a face por segundo. Já a pesquisa de satisfação mostrou que o sistema desenvolvido é de fácil utilização, não é desconfortável de utilizar, não causa cansaço e que pode ser utilizado em inúmeras aplicações como acessar a internet e jogar jogos simples, tanto por pessoas com deficiência motora quanto por outros públicos.

Trabalhos futuros poderiam ser desenvolvidos tendo em vista 2 vertentes. A primeira de melhorias no próprio sistema, ou interfaces computacionais voltadas às pessoas com deficiência. O ponto que apresentou o menor desempenho do sistema desenvolvido foi a definição do estado de abertura e fechamento dos olhos. Sistemas que utilizam recursos de baixo custo, como *webcam* de baixa resolução, podem apresentar dificuldade de identificar movimentos em regiões pequenas, o que acontece na região dos olhos, trabalhos futuros poderiam apresentar soluções para esse problema, abrindo muitas outras aplicações dessa possível solução. Adicionalmente a melhorias técnicas, testes com os usuários deficientes poderiam ser realizados, ajustando o sistema à suas necessidades e identificando pontos de melhorias específicos. A outra vertente seria o desenvolvimento desse tipo de aplicação para outras plataformas, como *mobile*. Hoje sistemas *mobile* utilizam sistemas baseados em imagens para aplicações limitadas, como desbloqueio do dispositivo por rastreamento facial, ou para modificação do fotos. Outras aplicações utilizando interfaces baseadas nos movimentos faciais poderiam ser desenvolvidas, como por exemplo controle de jogos ou aplicativos de digitação.

Independente de futuros desenvolvimentos de interfaces baseadas em imagens, é certo que novas interfaces devem continuar a serem desenvolvidas. Entretanto, pesquisas e desenvolvimentos voltados à usabilidade das pessoas com deficiência devem sempre continuar sendo realizados, uma vez que a sociedade deve garantir a esse público o acesso às tecnologias e sua inclusão digital e social.

REFERÊNCIAS BIBLIOGRÁFICAS

AL-MOHAIR, H. K.; MOHAMAD-SALEH, J.; SUANDI, S. A. **Color space selection for human skin detection using color-texture features and neural networks.** In: *2014 International Conference on Computer and Information Sciences (ICCOINS)*. : IEEE, 2014.

ARYA, K.; ADARSH, A. **An Efficient Face Detection and Recognition Method for Surveillance.** In: *2015 International Conference on Computational Intelligence and Communication Networks (CICN)*. : IEEE, 2015.

ATADJANOV, I.; LEE, S. **Bilateral symmetry detection based on scale invariant structure feature.** In: *2015 IEEE International Conference on Image Processing (ICIP)*. : IEEE, 2015.

BALLARD, P.; STOCKMAN, G. **Computer operation via face orientation.** In: *[1992] Proceedings. 11th IAPR International Conference on Pattern Recognition*. : IEEE Comput. Soc. Press, 1992.

BARBOSA, S. D. J. *Interacao Humano-computador*. : cap. 2 e 4. Elsevier, 2010.

BAY, H. et al. **Speeded-Up Robust Features (SURF).** *Computer Vision and Image Understanding*, Elsevier BV, v. 110, n. 3, p. 346–359, 2008.

BENJAMIM, X. C. et al. **Visual identification of medicine boxes using features matching.** In: *2012 IEEE International Conference on Virtual Environments Human-Computer Interfaces and Measurement Systems (VECIMS) Proceedings*. : IEEE, 2012.

BERSCH, R. *Introdução à Tecnologia Assistiva*. : Assistiva - Tecnologia e Educação, 2013. Disponível em: <<http://www.assistiva.com.br/tassistiva.html>>.

BETKE, M.; GIPS, J.; FLEMING, P. **The Camera Mouse: Visual Tracking of Body Features to Provide Computer Access for People With Severe Disabilities.** *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Institute of Electrical and Electronics Engineers (IEEE), v. 10, n. 1, p. 1–10, 2002.

- BIAN, Z.-P. et al. **Facial Position and Expression-Based Human–Computer Interface for Persons With Tetraplegia**. *IEEE Journal of Biomedical and Health Informatics*, Institute of Electrical and Electronics Engineers (IEEE), v. 20, n. 3, p. 915–924, 2016.
- BLAKE, J. *Natural User Interfaces in .NET*. : cap. 1. Manning Publications, 2011.
- BRASIL. *Tecnologia Assistiva*. : cap. 2 e 3. SDHPR - Secretaria Nacional de Promoção dos Direitos da Pessoa com Deficiência, 2009.
- CASS, S. *The 2016 Top Programming Languages*. 2016. IEEE Spectrum. Acesso em 02 de outubro de 2016. Disponível em: <http://docs.opencv.org/trunk/d7/d8b-tutorial_py_face_detection.html>.
- DIX, A. et al. *Human-Computer Interaction (3rd Edition)*. : cap. 1, 2 e 3. Pearson, 2003.
- DLIB. *Página Principal*. 2017. Acesso em 31 de maio de 2017. Disponível em: <<http://dlib.net/>>.
- EL-BIALY, A. **Towards a Complete Computer Dental Treatment System**. In: *2008 Cairo International Biomedical Engineering Conference*. : IEEE, 2008.
- FREUND, Y.; SCHAPIRE, R. E. **A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting**. *Journal of Computer and System Sciences*, Elsevier BV, v. 55, n. 1, p. 119–139, 1997.
- FU, Y.; HUANG, T. **hMouse: Head Tracking Driven Virtual Computer Mouse**. In: *2007 IEEE Workshop on Applications of Computer Vision*. : IEEE, 2007.
- GAO, F. et al. **A vision-based user interface of a mobile robotic arm for people with severe disabilities**. In: *2015 International Conference on Intelligent Informatics and Biomedical Sciences (ICIIBMS)*. : IEEE, 2015.
- GOEL, P.; AGARWAL, S. **An Illumination Invariant Robust and Fast Face Detection, Feature Extraction Based Face Recognition System**. In: *2012 Third International Conference on Computer and Communication Technology*. : IEEE, 2012.

GOR, A. K.; BHATT, M. S. **Fast scale invariant multi-view face detection from color images using skin color segmentation & trained cascaded face detectors**. In: *2015 International Conference on Advances in Computer Engineering and Applications*. : IEEE, 2015.

GRFOOD. **Telas**. 2016. GrSoft. Acesso em 11 de agosto de 2016. Disponível em: <<http://www.grfood.com.br/telas.php>>.

HJELMÅS, E.; LOW, B. K. **Face Detection: A Survey**. *Computer Vision and Image Understanding*, Elsevier BV, v. 83, n. 3, p. 236–274, 2001.

IBGE. **Censo Demográfico 2000 - Características gerais da população - Resultados da amostra**. Rio de Janeiro: cap. 1.4. Instituto Brasileiro de Geografia e Estatística - IBGE, 2003.

IBGE. **Censo Demográfico 2010 - Características gerais da população, religião e pessoas com deficiência - Características urbanísticas do entorno dos domicílios**. Rio de Janeiro: cap. 1.3. Instituto Brasileiro de Geografia e Estatística - IBGE, 2012.

IBM. **Página sobre o IBM Watson**. 2017. Acesso em 22 de maio de 2017. Disponível em: <<https://www.ibm.com/watson/>>.

IBUG. **300 Faces In-the-Wild Challenge**. 2017. Acesso em 07 de novembro de 2017. Disponível em: <<https://ibug.doc.ic.ac.uk/resources/300-W/>>.

ITS. **Tecnologia Assistiva nas Escolas - Recursos básicos de acessibilidade sócio-digital para pessoas com deficiência**. São Paulo: cap. 1 e 2. Instituto de Tecnologia Social (ITS Brasil), 2008.

JACKO, J. A. **Human Computer Interaction Handbook: Fundamentals, Evolving Technologies, and Emerging Applications, Third Edition (Human Factors and Ergonomics)**. : cap. 6. CRC Press, 2012.

JAIN, J.; LUND, A.; WIXON, D. **The future of natural user interfaces**. In: *Proceedings of the 2011 annual conference extended abstracts on Human factors in computing systems - CHI EA 11*. : ACM Press, 2011.

- Ji, S.; LU, X.; XU, Q. **A fast face detection method combining skin color feature and AdaBoost**. In: *2014 International Conference on Multisensor Fusion and Information Integration for Intelligent Systems (MFI)*. : IEEE, 2014.
- JIN, J.; LI, W. **A Survey of the Information Fusion in MMHCI**. In: *2010 International Conference on Machine Vision and Human-machine Interface*. : IEEE, 2010.
- KAZEMI, V.; SULLIVAN, J. **One millisecond face alignment with an ensemble of regression trees**. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2014. p. 1867–1874.
- KHAN, A. A.; IQBAL, M. **A Motion Detection Based Surveillance System (MDSS)**. In: *2011 First International Conference on Informatics and Computational Intelligence*. : IEEE, 2011.
- KJELDSEN, R. **Improvements in vision-based pointer control**. In: *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility - Assets 06*. : ACM Press, 2006.
- KRAICHAN, C.; PUMRIN, S. **Face and eye tracking for controlling computer functions**. In: *2014 11th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*. : IEEE, 2014.
- KRIG, S. *Computer Vision Metrics: Survey, Taxonomy, and Analysis*. : Apress, 2014.
- KRISHNA et al. **Automated traffic monitoring system using computer vision**. In: *2016 International Conference on ICT in Business Industry & Government (ICTBIG)*. : IEEE, 2016.
- LIANG, D. et al. **Video2Cartoon: A System for Converting Broadcast Soccer Video into 3D Cartoon Animation**. *IEEE Transactions on Consumer Electronics*, Institute of Electrical and Electronics Engineers (IEEE), v. 53, n. 3, p. 1138–1146, 2007.
- LOWE, D. G. **Distinctive Image Features from Scale-Invariant Keypoints**. *International Journal of Computer Vision*, Springer Nature, v. 60, n. 2, p. 91–110, 2004.

- MA, Y.; PENG, T.; ZHANG, T. **A fast and robust face detection and tracking algorithm.** In: *2014 IEEE International Conference on Signal Processing, Communications and Computing (ICSPCC)*. : IEEE, 2014.
- MANN, S. *Intelligent Image Processing (Adaptive and Cognitive Dynamic Systems: Signal Processing, Learning, Communications and Control)*. : cap. Intr. e 2. Wiley-IEEE Press, 2008.
- MARKUŠ, N. et al. **High-performance face tracking.** In: *Proceedings of the 3rd Symposium on Facial Analysis and Animation - FAA 12*. : ACM Press, 2012.
- MARKUŠ, N. et al. **Fast localization of facial landmark points.** *arXiv preprint arXiv:1403.6888*, 2014.
- MICROSOFT. *Página sobre desenvolvimento*. 2017. Acesso em 20 de fevereiro de 2017. Disponível em: <<https://developer.microsoft.com/en-us/windows/desktop/develop>>.
- MITCHELL, S. C. *Binary Features for Object Detection and Landmarking*. 2018. Acesso em 13 de fevereiro de 2018. Disponível em: <<https://www.slideshare.net/stevencharlesmitchell/binary-features-for-object-detection-and-landmarking>>.
- MUTTU, Y.; VIRANI, H. G. **Effective face detection, feature extraction & neural network based approaches for facial expression recognition.** In: *2015 International Conference on Information Processing (ICIP)*. : IEEE, 2015.
- NAIZHONG, Z.; JING, W.; JUN, W. **Hand-free head mouse control based on mouth tracking.** In: *2015 10th International Conference on Computer Science & Education (ICCSE)*. : IEEE, 2015.
- NIYOMUGABO, C.; HONG, K. J.; KIM, T. Y. **A fast face detection method based on skin color and Adaboost.** In: *The 18th IEEE International Symposium on Consumer Electronics (ISCE 2014)*. : IEEE, 2014.
- OPENCV. *Página Principal*. 2017. Acesso em 31 de maio de 2017. Disponível em: <<http://opencv.org/>>.

- OSMAN, T. et al. **Intelligent traffic management system for cross section of roads using computer vision.** In: *2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC)*. : IEEE, 2017.
- OUANAN, H.; OUANAN, M.; AKSASSE, B. **Facial landmark localization: Past, present and future.** In: *2016 4th IEEE International Colloquium on Information Science and Technology (CiSt)*. : IEEE, 2016.
- PALLEJÀ, T. et al. **Implementation of a robust absolute virtual head mouse combining face detection, template matching and optical flow algorithms.** *Telecommunication Systems*, Springer Nature, v. 52, n. 3, p. 1479–1489, 2011.
- PAPAGEORGIOU, C.; OREN, M.; POGGIO, T. **A general framework for object detection.** In: *Sixth International Conference on Computer Vision IEEE*. : Narosa Publishing House, 1998.
- PARMAR, K.; MEHTA, B.; SAWANT, R. **Facial-feature based Human-Computer Interface for disabled people.** In: *2012 International Conference on Communication, Information & Computing Technology (ICCICT)*. : IEEE, 2012.
- PING, S. T. Y.; CHUN, H. W.; BOONPING, L. **Face Detection through Template Matching and Color Segmentation.** *Stanford University. EE 368*, 2003.
- PRINCE, S. J. D. **Computer vision: models, learning and inference.** : cap. Intr. e 13. Cambridge University Press, 2012.
- QURESHI, A. et al. **Performance analysis of skin classifiers in RGB and YCbCr color space.** In: *17th IEEE International Multi Topic Conference 2014*. : IEEE, 2014.
- RENOUST, B. et al. **A Social Network Analysis of Face Tracking in News Video.** In: *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. : IEEE, 2015.
- ROCHA, H. V. da; BARANAUSKAS, M. C. C. **Design e Avaliação de Interfaces Humano-Computador.** Campinas: cap. 1. Instituto de Computação - Universidade Estadual de Campinas, 2003.

- SABBAGH, R. *Scrum: Gestão ágil para projetos de sucesso*. São Paulo: Casa do Código, 2014.
- SAFFER, D. *Designing Gestural Interfaces: Touchscreens and Interactive Devices*. : cap. 1 e 4. O'Reilly Media, 2008.
- SAGONAS, C. et al. **300 Faces In-The-Wild Challenge: database and results**. *Image and Vision Computing*, Elsevier BV, v. 47, p. 3–18, mar 2016.
- SAGONAS, C. et al. **300 Faces in-the-Wild Challenge: The First Facial Landmark Localization Challenge**. In: *2013 IEEE International Conference on Computer Vision Workshops*. : IEEE, 2013.
- SAGONAS, C. et al. **A Semi-automatic Methodology for Facial Landmark Annotation**. In: *2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*. [S.l.]: IEEE, 2013.
- SATO, N.; YODA, I.; INOUE, T. **Shoulder gesture interface for operating electric wheelchair**. In: *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*. : IEEE, 2009.
- SCHWABER, K.; SUTHERLAND, J. *Guia do Scrum: Um guia definitivo para o Scrum: As regras do jogo*. : , 2013.
- SEMIENIUTA, O.; DRANSFELD, S.; FALKMAN, P. **Vision-based robotic system for picking and inspection of small automotive components**. In: *2016 IEEE International Conference on Automation Science and Engineering (CASE)*. : IEEE, 2016.
- SHRIVASTAVA, M.; MATLANI, P. **A smoke detection algorithm based on K-means segmentation**. In: *2016 International Conference on Audio, Language and Image Processing (ICALIP)*. : IEEE, 2016.
- SIRILUCK, W. et al. **Blink and click**. In: *Proceedings of the 1st international convention on Rehabilitation engineering & assistive technology in conjunction with 1st Tan Tock Seng Hospital Neurorehabilitation Meeting - i-CREAtE 07*. : ACM Press, 2007.

SOUKOREFF, R. W.; MACKENZIE, I. S. **Towards a standard for pointing device evaluation, perspectives on 27 years of Fitts' law research in HCI.** In: . [S.l.]: Elsevier BV, 2004. v. 61, n. 6, p. 751–789.

SZELISKI, R. **Computer Vision: Algorithms and Applications.** : cap. 1, 3, 5 e 14. Springer, 2010.

TAKAHASHI, M.; FUJII, M.; YAGI, N. **Automatic Pitch Type Recognition from Baseball Broadcast Videos.** In: *2008 Tenth IEEE International Symposium on Multimedia.* : IEEE, 2008.

TAKAHASHI, Y. et al. **Human interface using PC display with head pointing device for eating assist robot and emotional evaluation by GSR sensor.** In: *Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation.* : IEEE, 2001.

TU, J.; HUANG, T.; TAO, H. **Face as Mouse Through Visual Face Tracking.** In: *The 2nd Canadian Conference on Computer and Robot Vision (CRV 05).* : IEEE, 2005.

VILLAROMAN, N. H.; ROWE, D. C.; HELPS, R. G. **Design and evaluation of face tracking user interfaces for accessibility.** In: *Proceedings of the 2nd annual conference on Research in information technology - RIIT 13.* : ACM Press, 2013.

VIOLA, P.; JONES, M. **Rapid object detection using a boosted cascade of simple features.** In: *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001.* : IEEE Comput. Soc, 2001.

VIOLA, P.; JONES, M. J. **Robust Real-Time Face Detection.** In: . Hingham, MA, USA: Kluwer Academic Publishers, 2004. v. 57, n. 2, p. 137–154. ISSN 0920-5691.

WANG, M. **Skin Color Detection using OpenCV.** 2017. Acesso em 15 de fevereiro de 2017. Disponível em: <<https://ccw1986.blogspot.com.br/2012/12/opencvhsv-skin-color-detection.html>>.

WIGDOR, D.; WIXON, D. **Brave NUI World: Designing Natural User Interfaces for Touch and Gesture.** : cap. 1 e 2. Morgan Kaufmann, 2011.

- WU, L.; WU, P.; MENG, F. **A Fast Face Detection for Video Sequences.** In: *2010 Second International Conference on Intelligent Human-Machine Systems and Cybernetics.* : IEEE, 2010.
- XU, G.; WANG, Y.; FENG, X. **A Robust Low Cost Virtual Mouse Based on Face Tracking.** In: *2009 Chinese Conference on Pattern Recognition.* : IEEE, 2009.
- YADAV, S.; NAIN, N. **Fast Face Detection Based on Skin Segmentation and Facial Features.** In: *2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS).* : IEEE, 2015.
- YANG, G.; HUANG, T. S. **Human face detection in a complex background.** *Pattern Recognition*, Elsevier BV, v. 27, n. 1, p. 53–63, 1994.
- YANG, M.-H.; KRIEGMAN, D.; AHUJA, N. **Detecting faces in images: a survey.** *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Institute of Electrical and Electronics Engineers (IEEE), v. 24, n. 1, p. 34–58, 2002.
- YOW, K. C.; CIPOLLA, R. **Feature-based human face detection.** *Image and Vision Computing*, Elsevier BV, v. 15, n. 9, p. 713–735, 1997.
- ZHANG, C.; ZHANG, Z. **A Survey of Recent Advances in Face Detection.** In: *Technical Report MSR-TR-2010-66.* : Microsoft Corporation - Microsoft Research, 2010.
- ZHANG, H.; YE, C. **An indoor navigation aid for the visually impaired.** In: *2016 IEEE International Conference on Robotics and Biomimetics (ROBIO).* : IEEE, 2016.

APÊNDICE

Apêndice A – Questionário da Pesquisa de Satisfação

Título: Pesquisa de Satisfação - Sistema InterFace

Descrição da Pesquisa: Essa pesquisa tem por objetivo entender o que os usuários acharam da utilização do sistema de interface natural interFace, proposto pelo aluno de Mestrado Gustavo Scalabrini Sampaio na dissertação "DESENVOLVIMENTO DE UMA INTERFACE COMPUTACIONAL NATURAL PARA PESSOAS COM DEFICIÊNCIA MOTORA BASEADA EM VISÃO COMPUTACIONAL". Ao responder essa pesquisa tente se colocar no lugar de uma pessoa com deficiência motora que não possui os movimentos de membros superiores, apenas o movimento da cabeça. Suas respostas ajudarão a melhorar o sistema e proporcionar ao público alvo uma maneira eficiente de utilizar computadores.

Perfil

- Informe seu sexo:
 - Masculino
 - Feminino

- Informe sua faixa de idade:
 - 17 anos ou menos
 - 18 - 29 anos
 - 30 - 39 anos
 - 40 - 59 anos
 - 60 anos ou mais

Experiência em Informática

Opções de resposta:

- Diariamente
- Semanalmente
- Eventualmente
- Nunca

Perguntas:

- Com que frequência você utiliza e-mail?
- Com que frequência você utiliza a internet?
- Com que frequência você utiliza as Redes Sociais?

Avaliação do Sistema - Tela

Opções de resposta:

1. Discorda Totalmente
2. Discorda
3. Não Concorda nem Discorda
4. Concorda
5. Concorda Totalmente

Afirmações:

- O tamanho das fontes estão adequados.
- A disposição das opções do sistema estão adequadas.

Avaliação do Sistema - Opções do Sistema

Opções de resposta:

1. Discorda Totalmente
2. Discorda
3. Não Concorda nem Discorda
4. Concorda
5. Concorda Totalmente

Afirmações:

- A possibilidade de determinar o que o movimento da face, dos olhos e da boca realizam no sistema operacional é útil.
- A possibilidade de ativar e desativar as funcionalidades acionadas pelos olhos e boca é útil.
- A possibilidade de regular a sensibilidade do mouse é útil.
- O recurso de Parar/Iniciar o sistema é útil.
- O recurso de Pausar o sistema é útil.
- O sistema possui opções que o tornam flexível para realizar tarefas diárias.

Avaliação do Sistema - Funcionamento

Opções de resposta:

1. Discorda Totalmente
2. Discorda
3. Não Concorda nem Discorda
4. Concorda
5. Concorda Totalmente

Afirmações:

- O movimento do cursor do mouse por meio dos movimentos da face é suave.
- O movimento do cursor do mouse por meio dos movimentos da face é preciso.
- O posicionamento do cursor do mouse por meio dos movimentos da face é preciso.
- Ativar o clique do mouse com a boca apresenta boa precisão.
- O sistema possui boa velocidade de resposta.

Avaliação do Sistema - Facilidade de uso e Aprendizado

Opções de resposta:

1. Discorda Totalmente
2. Discorda
3. Não Concorda nem Discorda
4. Concorda
5. Concorda Totalmente

Afirmações:

- O sistema é de fácil aprendizado.
- O sistema é fácil de utilizar.
- O sistema permite selecionar de forma fácil ícones grandes em tela.
- O sistema permite selecionar de forma fácil ícones pequenos em tela.
- Caso utilizasse o sistema cotidianamente conseguiria realizar muitas atividades.

Avaliação do Sistema - Conforto

Opções de resposta:

1. Discorda Totalmente
2. Discorda
3. Não Concorda nem Discorda
4. Concorda
5. Concorda Totalmente

Afirmações:

- A utilização do sistema é confortável.
- Poderia passar muito tempo utilizando a ferramenta.
- Não senti cansaço durante a utilização do sistema.

Avaliação do Sistema - Aplicação

Opções de resposta:

1. Discorda Totalmente
2. Discorda
3. Não Concorda nem Discorda
4. Concorda
5. Concorda Totalmente

Afirmações:

- Poderia utilizar o sistema para navegar na internet.
- Poderia utilizar o sistema para redigir textos.
- Poderia utilizar o sistema para me comunicar com outras pessoas nas redes sociais.
- Poderia utilizar o sistema para jogar jogos simples.
- O sistema pode ser utilizado por pessoas com deficiências motoras.
- O sistema pode ser utilizados por outros públicos.

Avaliação do Sistema - Satisfação

Opções de resposta:

- Sim
- Não

Perguntas:

- Você usaria novamente o sistema?
- Você indicaria o sistema para um amigo?

Avaliação do Sistema - Comentários

Deixe seu comentário sobre o sistema:

- O usuário poderia deixar livremente seu comentário.