

Expressões Faciais como Interface de Acessibilidade para Controle de Aplicações

ELOÍZA MARTINS PRIMO CAPELOCI
ENZO LUIZ TSUTSUMI DE ALMEIDA JOSÉ
LUIZ GUSTAVO FERNANDES SALVADOR
MARCEL SANTOS SILVA

Resumo

As expressões faciais são importantes ferramentas de comunicação humana, podendo apresentar uma infinidade de combinações. Para pessoas com ausência ou limitações de mobilidade, essa manifestação se torna ainda mais importante, sendo a única forma de se comunicar, em alguns casos. Este trabalho visa acentuar ainda mais essa importância, utilizando das expressões faciais para acesso às funcionalidades no uso de aplicações. A aplicação em questão será um software voltado para computadores pessoais, configurado de forma simples, altamente customizável, além de não depender de equipamento específicos. Para tanto, a identificação das articulações do rosto é feita por meio do mapeamento da face do usuário via imagem da *webcam*, utilizando da distância Euclidiana e de uma constante definida pelo usuário (*threshold*) para detectar a expressão. A ação a ser tomada de acordo com cada manifestação identificada é totalmente configurável pelo interessado, por meio de uma interface *web*, de fácil utilização. Uma vez que o objetivo deste trabalho é criar uma forma de acessibilidade para aplicativos baseada na identificação de expressões faciais, conclui-se que o propósito deste projeto foi alcançado com ressalvas. O principal ponto a ser considerado é a capacidade de adaptação do sistema, já que cada tipo de deficiência possui características únicas. Apesar disto, existem pontos que diminuem a usabilidade do *software* como consumo de recursos do computador, imprecisões e fadiga muscular após longos períodos.

Palavras-chave: Identificação de Expressões Faciais; Acessibilidade em aplicativos. Interação homem-máquina; Aplicações interativas.

Facial Expressions as an Accessibility Interface for Application Control

Abstract

Facial expressions are important tools for human communication, capable of presenting an infinite number of combinations. For people with absence or limitations of mobility, this manifestation becomes even more important, being the only way to communicate in some cases. This work aims to further accentuate this importance, using facial expressions to access functionalities in the use of applications. The application in question will be software aimed at personal computers, set in a simple way, highly customizable, besides not depending on specific equipment. To do so, the identification of facial joints is done by mapping the user's face via webcam image, using Euclidean distance and a user-defined constant (threshold) to detect the expression. The action to be taken according to each identified manifestation is fully configurable by the interested party, through an easy-to-use web interface. Since the objective of this work is to create a form of accessibility for applications based on the identification of facial expressions, it is concluded that the purpose of this project was achieved with reservations. The main point to be considered is the system's adaptability, as each type of disability has unique characteristics. Despite this, there are points that decrease the software's usability such as computer resource consumption, inaccuracies, and muscle fatigue after long periods.

Keywords: Facial Expression Recognition; Accessibility in applications. Human-computer interaction; Interactive applications.

1 INTRODUÇÃO

As expressões faciais humanas são ferramentas poderosas de comunicação não-verbal capazes de expressar toda uma gama de sentimentos (Revina; Emmanuel, 2021). Os

movimentos dos olhos, sobrancelhas, pálpebras, boca e do rosto com um todo geram uma infinidade de sentidos de acordo com as bases culturais de uma pessoa (Ekman, 1993). Nesse mesmo sentido, existem incontáveis combinações músculo-faciais que podem ser exploradas para uma nova era da comunicação: a virtual.

A definição de uma pessoa com deficiência é um indivíduo que possua alguma condição física, mental ou emocional que de alguma forma dificulte o funcionamento de uma célula, órgão ou membro desta pessoa, assim podendo gerar efeitos que variam desde dificuldade de visão até desmaios (Altman, 2014). De acordo com a Biblioteca Virtual em Saúde do Ministério da Saúde (s. d.) o número de deficientes no Brasil se aproxima de 45 milhões, sendo 28.8% deles deficientes físicos. Pessoas que sofrem com esse tipo específico de deficiência podem ter limitações ou serem impossibilitados de usarem os membros superiores, sendo estes a principal maneira de interagir com a tecnologia nos dias atuais (Ahmed, F.; Ahmed, H.; Miky, 2024). Assim se torna necessário o desenvolvimento de novas maneiras de acesso a tecnologia que integre esses indivíduos (Desmond *et al.*, 2018).

A interação entre seres humanos e a tecnologia atinge, cada vez mais, patamares inimagináveis, transformando a vida em sociedade de maneiras profundas e abrangentes (Fernandes *et al.*, 2021). À medida que as aplicações digitais desempenham um papel cada vez mais vital em nossas atividades diárias, a necessidade de interfaces intuitivas e acessíveis se tornou uma prioridade fundamental (Hott; Fraz, 2019). Nesse contexto, a identificação de expressões faciais emerge como uma promissora abordagem para otimizar a usabilidade e a acessibilidade de artifícios eletrônicos, possibilitando aos usuários o acesso a funcionalidades de forma natural e eficaz (Valentim; Barreto; Muchaluat-Saade, 2021).

Pesquisas que buscam traduzir movimentos faciais em interações em máquinas datam desde 1995. Ballard e Stockman (1995), por exemplo, tinham o objetivo de criar uma interface acessível para pessoas com deficiência, para tal utilizaram o plano formado pelos olhos e nariz para calcular a direção que um usuário estava olhando, usando este ângulo para controlar o movimento do *mouse*. Toyama (1998) possuía o mesmo objetivo, porém utilizou uma câmera 3D para fazer o mapeamento da face, assim atingindo uma movimentação em tempo real do *mouse*.

Embora existam diversos estudos que utilizam a face como interface homem-máquina, poucos buscam empregá-la em sistemas voltados ao usuário final (Ballard; Stockman, 1995), também há carências de sistemas que emulem ações como cliques do mouse (Toyama, 1998) e outros *softwares* com esse mesmo objetivo possuem limitações de customização (Deepateep; Vichitvejpaisal, 2020). Assim, pesquisas nesta área se tornam cada vez mais relevantes.

Dado o apresentado, o objetivo deste trabalho consiste no desenvolvimento de uma aplicação para identificar expressões faciais de forma simples, e utilizá-las como ferramenta de acessibilidade, tornando possível o uso de aplicativos por indivíduos com limitações físicas ou desafios de mobilidade, prezando pela satisfação do usuário. Esta pesquisa busca explorar como essa tecnologia pode ser aplicada de maneira eficaz, proporcionando uma experiência de usuário mais inclusiva e eficiente. A aplicação em questão será um *software* voltado para computadores pessoais, podendo ser configurado de forma simples, sendo altamente customizável, além de não depender de nenhum equipamento específico além de uma câmera comum.

2 DESENVOLVIMENTO

É clara a crescente relevância da identificação de expressões faciais como um meio de aprimorar a interação homem-máquina. À medida que a sociedade se torna mais digital e conectada, é essencial desenvolver soluções que não apenas simplifiquem a operação de

aplicativos, mas também considerem a diversidade de usuários, incluindo aqueles com necessidades especiais (Pirrone *et al.*, 2023).

No entanto, o desenvolvimento de um método preciso para identificação das expressões faciais é um desafio (Su *et al.*, 2009). Potikanya *et al.* (2016, p.201-204), por exemplo, baseou a face humana em um plano, dividindo-a em subconjuntos desse plano e, por meio da Análise de Componentes Principais¹, definiu os principais subconjuntos para processamento das expressões.

Já Li e Da (2012, p. 668-679), encontraram uma problemática na utilização do plano 2D no reconhecimento facial: a dificuldade de identificação das expressões com variações na pose do usuário e com a iluminação do ambiente. Para resolver esse problema, eles utilizaram um modelo 3D da face, definindo um mapa de deformações faciais de 6 regiões do rosto e, a partir da PSA¹, um classificador de eliminação é montado para rejeição de posições faciais e um reconhecimento mais eficiente. Outro exemplo de método é o proposto por Madhu e Amutha (2013, p.1464-1470), onde se utilizaram da Matriz de Concorrência de Nível de Cinza² para levantamento do nível de cinza dos *pixels* da imagem do usuário, utilizando as informações levantadas para classificação, com o uso de distância Euclidiana, de um conjunto das 10 imagens mais similares em uma base de dados. Por fim, a PSA e o Padrão Ternário Local³ são utilizados para se definir a expressão facial.

Para escolha do método a ser utilizado, esta pesquisa prezou por: i) Simplicidade e eficiência - interesse pela utilização de um método de baixa complexidade, permitindo o processamento rápido e eficiente, exigindo poucos recursos computacionais; ii) Baixa Demanda de Dados de Treinamento - diversas técnicas com uso de redes neurais são utilizadas na literatura para detecção de expressões faciais (Ridha *et al.*, 2019; Thomas, Mathew, 2012; Mudit, Dyvia, Joshi, 2022), no entanto, essas exigem grandes volumes de amostras de treinamento, o que foi evitado pelo autor; iii) Transparência Analítica - houve preferência por um método que determinasse com clareza a expressão apresentada pelo usuário; iv) Adaptabilidade - um sistema personalizável, de acordo com as necessidades e desejos do usuário, também foi uma das prerrogativas deste trabalho; e v) Resiliência a Perturbações de Iluminação e Fundo - Alguns dos métodos utilizados na literatura, como o apresentado por Madhu e Amutha (2013, p. 1464-1470), possuem uma forte dependência das texturas e cores das imagens. Esse trabalho prioriza um método mais robusto quanto à dependência de luminosidade e perspectiva.

Assim, através dos itens identificados acima, optou-se pela utilização de distâncias euclidianas propostas por Madhu e Amutha (2013, p. 1464-1470). Neste contexto, o sistema de identificação de expressões faciais permitiu a utilização de uma metodologia de análise que recai sobre a determinação das expressões faciais mediante a caracterização geométrica do espaço facial, ou seja, cada ponto da malha gerada da face do usuário possui coordenadas cartesianas em um plano bidimensional (x, y). O estabelecimento da presença ou ausência de uma expressão facial de interesse é então discernido por meio da avaliação das distâncias euclidianas entre pares de tais pontos, sobremodo ponderadas por uma constante predefinida, chamada de *Threshold*. A manifestação de uma expressão facial é, então, validada quando a magnitude resultante, em módulo, dessa ponderação ultrapassa essa constante arbitrariamente definida.

Desta forma, o programa é dividido em duas partes: a primeira parte captura a imagem gerada pela *webcam* e faz um mapeamento facial, gerando uma malha de pontos em

¹ Do inglês, Principal Component Analysis - PSA, um método usado para construir modelos lineares de forma e movimento, sendo amplamente utilizado para modelar a variação nos rostos das pessoas (TORRE, 2002).

² Do inglês, Gray Level Co-occurrence Matrix – GLCM.

³ Descritor de textura usado em aplicações de imagens, permitindo um cálculo mais fácil e reduzindo a complexidade na recuperação de imagens (VARGHESE *et al.*, 2011).

representação à face do usuário, limitada a regiões pré-definidas, sendo elas: sobrelancelha, boca, olhos e nariz; já a segunda parte é responsável por gerar a interface visual onde é possível configurar as ações que o código executará.

A estrutura de configuração da ação, por sua vez, é composta pelas expressões faciais esperadas para a execução. Cada expressão possui uma barra deslizante que define seu valor de *threshold*.

Para o teste do sistema final, foi optado por seguir uma metodologia baseada em tarefas. Nestes testes, os usuários têm o objetivo de concluir objetivos pré-definidos que envolvam o sistema em questão (Ji *et al.*, 2006). Devido ao curto prazo de execução da pesquisa optou-se pelos próprios pesquisadores testarem o sistema, além disso, o fato de um deles possuir deficiência física faz com que se encaixando no público alvo do *software*. Assim, foram escolhidas algumas tarefas do cotidiano, sendo elas: fazer uma pesquisa *online*, digitar uma pequena dissertação, assistir a um vídeo em qualquer rede social e jogar um jogo virtual. Dessa forma, todas as tarefas puderam ser executadas, todavia com algumas limitações:

- A depender do posicionamento do usuário em relação ao centro da imagem, a detecção de certas expressões pode ser limitada ou até impossível, devido a maiores angulações em relação ao dispositivo impedirem a visualização parcial ou completa de certas estruturas faciais;
- A presença de objetos na frente utilizador do sistema ou a baixa luminosidade podem dificultar ou impedir a análise da face;
- A qualidade da imagem da câmera presente na máquina hospedando o *software* pode gerar discretas imprecisões no uso do sistema;
- Certas características anatômicas podem interferir no uso do programa, por exemplo, sujeitos com olhos achatados podem experimentar dificuldades com a detecção do piscar dos olhos.

Além disso, dois pontos são dignos de serem citados. Primeiramente, observou-se que cada um dos pesquisadores se sentiu mais confortável com um conjunto diferente de expressões, devido à melhor detecção de alguns movimentos faciais, porém, isso era previsto, uma vez que essa foi a motivação de incluir a possibilidade de diferentes configurações. Outro ponto foi a necessidade de ajustes na configuração para as diferentes tarefas, o que embora não fosse obrigatório aumentou a fluidez no uso da aplicação.

2.2 Materiais e Métodos

Através de um análise quali-quantitativa foi desenvolvido um programa computacional utilizando a linguagem *Python*.

Para o desenvolvimento do programa computacional foi utilizada a linguagem de programação *Python*. Os motivos para tal são a simplicidade de utilização (TIEN *et al.*, 2013), por ser uma ferramenta de código aberto (*Open Source*) e versátil, apresentando uma imensa gama de bibliotecas (Singh *et al.*, 2022; Upala, Wong, 2019; Vadlapati, Velan, Varghese, 2021).

As bibliotecas utilizadas neste trabalho foram: *cv2* - A biblioteca *OpenCV*, frequentemente referida como "*cv2*" no *Python*, é uma biblioteca de código aberto amplamente utilizada para processamento de imagens e visão computacional (Opencv, 2023). Ela fornece uma ampla gama de funções e ferramentas para realizar tarefas relacionadas ao processamento de imagens e vídeos. No âmbito desta pesquisa, ela foi utilizada para capturar, exibir e desenhar formas nas imagens capturadas pela *webcam* do computador.

A *MediaPipe* é uma estrutura de desenvolvimento *cross-platform*, isto é, com funcionamento em diversas plataformas, que oferece ferramentas para processar dados

multimodais, como vídeo, áudio e outros tipos de séries temporais (Kukil, 2022). Neste trabalho, ela foi utilizada para processar imagens e fazer o mapeamento facial.

A *math* é uma ferramenta poderosa para realizar cálculos matemáticos. Ela oferece acesso às funções matemáticas definidas pelo padrão C⁴, sendo especialmente úteis para operações numéricas, como arredondamento, cálculo de fatorial, exponenciação e muito mais (AWARI, 2022). Para este trabalho, essa biblioteca foi útil para realizar os cálculos e descobrir a distância entre os pontos da face.

A *genericpath* é parte do módulo *os.path*, e fornece uma implementação genérica para manipulação de caminhos de arquivos e diretórios. Ela contém funções e classes que lidam com operações comuns em caminhos, independentemente do sistema operacional (Os.Path, 2023). Neste trabalho, seu módulo *is.file* foi utilizado para reconhecer arquivos executáveis na função de abrir o aplicativo.

A *asyncio* é uma ferramenta essencial para lidar com operações assíncronas, como chamadas de rede, E/S de arquivos e operações em banco de dados, sem bloquear o fluxo principal do programa (Devmedia, 2018). No âmbito deste trabalho, ela foi utilizada para definir funções assíncronas para abrir e maximizar janelas na função de abrir aplicativo.

A *subprocess* é uma poderosa biblioteca de algoritmos para criar e gerenciar subprocessos a partir de um programa *Python*, permitindo que se execute comandos externos, conecte-se aos canais de entrada/saída/erro desses processos e obtenha seus códigos de retorno (Subprocess, 2023). Neste trabalho, ela foi utilizada para iniciar a execução da interface visual.

A *pyautogui* possibilita automatizar interações com a interface gráfica do usuário (GUI), possibilitando controlar o mouse e o teclado de forma programática, automatizar tarefas repetitivas, criar atalhos personalizados e até mesmo desenvolver *bots*⁵ simples (Sweigart, 2023). Para este trabalho, ela foi utilizada para fazer o controle do *mouse* e teclado.

O *os* é um módulo versátil para interagir com o sistema operacional, permitindo realizar várias ações específicas, como navegar por diretórios, criar novos diretórios, executar comandos no terminal e obter informações do sistema (Os, 2023). Aqui utilizada para executar comandos do sistema.

A *json* é essencial para trabalhar com dados no formato *JSON (JavaScript Object Notation)*, sendo esse um formato leve de troca de dados, amplamente utilizado para integração entre sistemas e comunicação entre aplicativos (Json, 2023). Servindo para salvar e carregar os arquivos json de configuração desta pesquisa.

A *threading* permite que os programadores criem, iniciem e gerenciem fluxos de um programa em execução, conhecidos como *threads* (Pantuza, 2017), em um programa *Python*, possibilitando criar *threads* independentes para executar tarefas específicas, permitindo que o programa continue executando outras tarefas em segundo plano (Threading, 2023). No presente estudo, o módulo *TIME* dessa biblioteca foi usado para criar as contagens regressivas na calibragem do sistema.

A biblioteca *PIL (Python Imaging Library)*, também conhecida como *Pillow*, é uma ferramenta essencial para processamento de imagens, oferecendo suporte a diversos formatos de arquivo, uma representação interna eficiente e capacidades poderosas de processamento de imagens (Lundh, 2023). Na implementação aqui descrita os módulos *ImageFont*, *ImageDraw* e *Image*, dessa biblioteca, foram utilizados para gerar os textos mostrados na câmera durante a calibragem.

A *numpy* é utilizada para computação científica em *Python*, oferecendo um objeto de matriz multidimensional, bem como uma variedade de rotinas para operações rápidas em

⁴ Funções de programações presentes na biblioteca matemática padrão da linguagem C.

⁵ Nome dado aos *softwares* desenvolvidos com o intuito de automatizar tarefas.

vetores de armazenamento (Numpy, 2023). Para a finalidade descrita, ela foi utilizada para conversão de imagens para lista de dados.

A *sys* é uma biblioteca essencial para interagir com o sistema operacional e controlar aspectos específicos do ambiente de execução (Sys, 2023). No âmbito deste trabalho, foi utilizada para parar a execução do código.

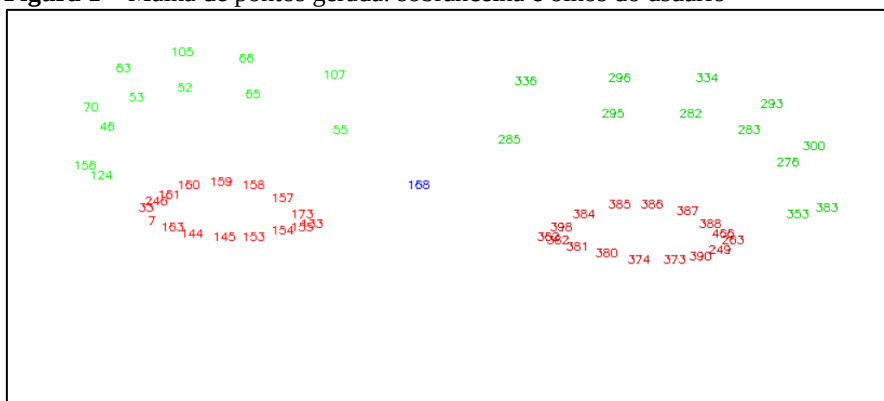
O *flask* é um *framework*⁶ leve para aplicação *web*, sendo projetado para facilitar o início rápido de projetos *web*, com a capacidade de escalar para aplicações mais complexas (Python, 2023). No caso específico desta aplicação, ele serviu de servidor local para a interface visual, modificando os arquivos de configuração e enviando o código da interface para o navegador.

Para configuração das ações que cada expressão facial representa, uma interface *web* foi construída. Dentre as vantagens para tal escolha, são passíveis de serem citadas: acessibilidade multiplataforma, facilidade de uso, navegação intuitiva e manutenção simplificada. Para sua elaboração foram utilizados *JavaScript*, *HTML* e *CSS*. Esses *softwares* são tecnologias padrão da *web* amplamente suportadas pelos diversos navegadores, além de possuírem uma vasta comunidade de desenvolvedores, bem como recursos, bibliotecas e *frameworks* disponíveis. O *HTML* é usado para estruturar o conteúdo da página, definindo elementos como cabeçalho, parágrafos, formulários, imagens e *links*, o que é essencial para a criação de uma interface *web* organizada (Duckett, 2011). O *CSS* permite a estilização da interface, definindo a aparência visual dos elementos *HTML*. Isso inclui cores, fontes, tamanhos, margens, posicionamento e outros aspectos visuais que contribuem para a usabilidade e a estética (Duckett, 2011). Por sua vez, o *JavaScript* é uma linguagem de programação poderosa para tornar a interface interativa. Ele permite a adição de comportamentos dinâmicos, validações de entrada, manipulação de eventos e comunicação com o servidor (Flanagan; Ferguson, 1998).

2.3 Resultados

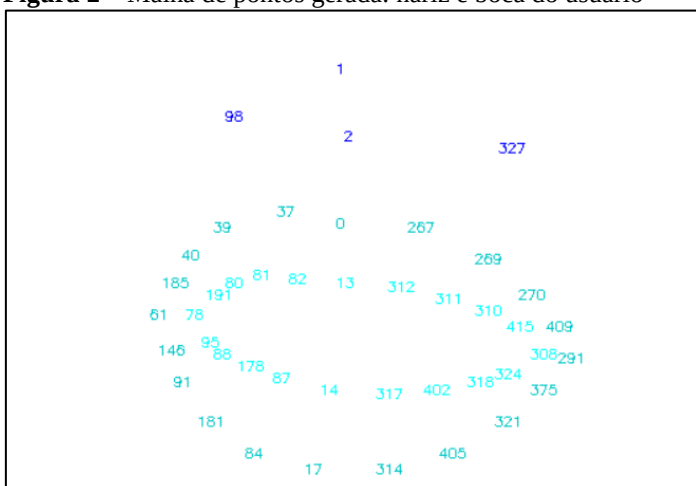
Seguindo a ordem de execução do algoritmo desenvolvido neste trabalho, o primeiro passo foi a aquisição de uma imagem, via *webcam* de *notebook*. A partir disso, o mapeamento da face do usuário foi realizado e está representado nas Figuras 1 e 2. A definição de cada ponto foi feita via algoritmo, com base em uma adaptação do repositório do GitHub do Annxingyuan (2020).

Figura 1 – Malha de pontos gerada: sobrancelha e olhos do usuário



Fonte: Elaborado pelos autores (2023)

⁶ Base, com ferramentas e padrões definidos, que serve como guia para simplificar e agilizar o desenvolvimento de *software*.

Figura 2 – Malha de pontos gerada: nariz e boca do usuário

Fonte: Elaborado pelos autores (2023).

Para discernimento das expressões do usuário, a Tabela 1 apresenta as definições das expressões, bem como os pontos utilizados para sua identificação.

Tabela 1 – Descrição das expressões, pontos de identificação e critério de discernimento.

Expressão	Pontos de comparação	Ativo quando
Erguer A Sobrancelha Esquerda	158; 145	Maior que o <i>threshold</i>
Erguer A Sobrancelha Direita	385; 374	Maior que o <i>threshold</i>
Piscar O Olho Esquerdo	65; 168	Menor que o <i>threshold</i>
Piscar O Olho Direito	282; 168	Menor que o <i>threshold</i>
Esticar Lado Esquerdo Da Boca	61; 2	Maior que o <i>threshold</i>
Esticar Lado Direito Da Boca	291; 2	Maior que o <i>threshold</i>
Abrir A Boca	13; 14	Maior que o <i>threshold</i>

Fonte: Elaborado pelos autores (2023).

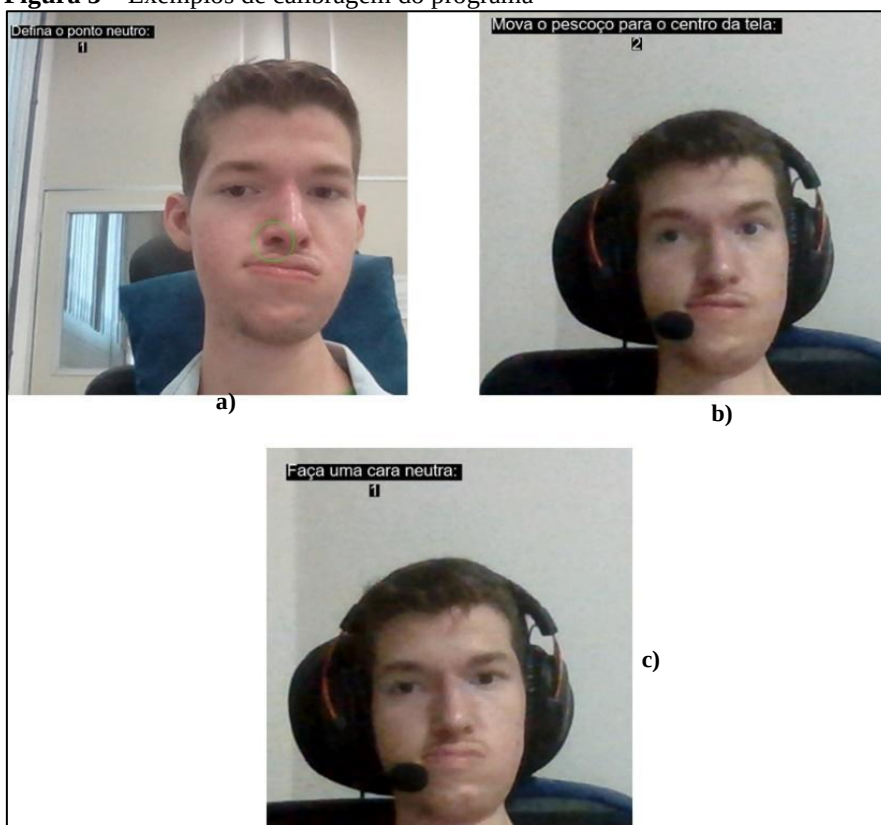
Para a configuração das ações, toma-se como base a Tabela 2, que dispõe as possíveis ações, suas respectivas descrições e os parâmetros necessários para definição da ação.

Tabela 2 – Ações, respectivos efeitos e parâmetros a serem definidos para sua execução.

Ação	Efeito	Parâmetros
Clicar	Efetua um clique de algum botão do <i>mouse</i>	Botão do <i>mouse</i> : esquerdo, meio ou direito
Mover O <i>Mouse</i>	Move o cursor do <i>mouse</i> para a posição definida	Posição X e Y
Segurar O Botão Do <i>Mouse</i>	Mantém algum botão do <i>mouse</i> pressionado	Botão do <i>mouse</i> : esquerdo, meio ou direito, e se o botão deve ser solto ao parar a expressão
Soltar O Botão Do <i>Mouse</i>	Solta algum botão do <i>mouse</i> pressionado	Botão do <i>mouse</i> : esquerdo, meio ou direito
Abrir Aplicativo	Abre o aplicativo especificado	Nome do aplicativo
Pesquisar	Pesquisa o texto informado	Texto
Apertar Teclas	Pressiona as teclas definidas	Uma ou mais teclas
Digitar Texto	Digita o texto informado	Texto

Fonte: Elaborado pelos autores (2023).

Para classificar as expressões, uma calibragem do programa é necessária, o que pode ser feito por meio da interface *web*. O compilado de imagens demonstradas na figura 3, apresenta exemplos dessa classificação. Durante esse procedimento, uma instrução é dada para o usuário, a qual aparece na tela. Embaixo desse texto, há um cronômetro, indicando o tempo de calibragem e tempo que o usuário possui para executar a orientação.

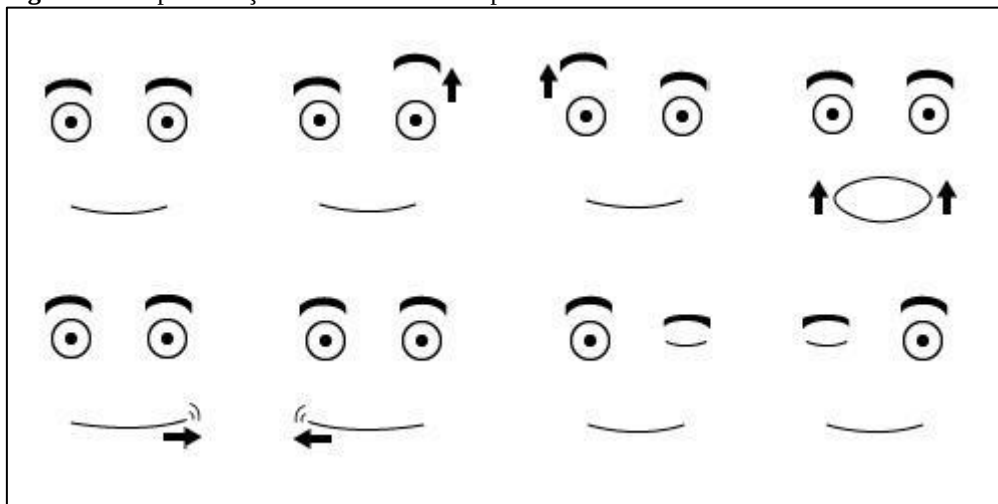
Figura 3 – Exemplos de calibragem do programa

Fonte: Elaborado pelos autores (2023)

A Figura 3.a e 3.b representam as instruções para calibração de ações associadas com o mouse. Já a Figura 3.c apresenta a instrução para o mapeamento da face do usuário.

Com o objetivo de tornar a interface mais intuitiva e amigável, representações de cada expressão foram desenhadas e são apresentadas na figura 4.

Figura 4 – Representação em desenho das expressões

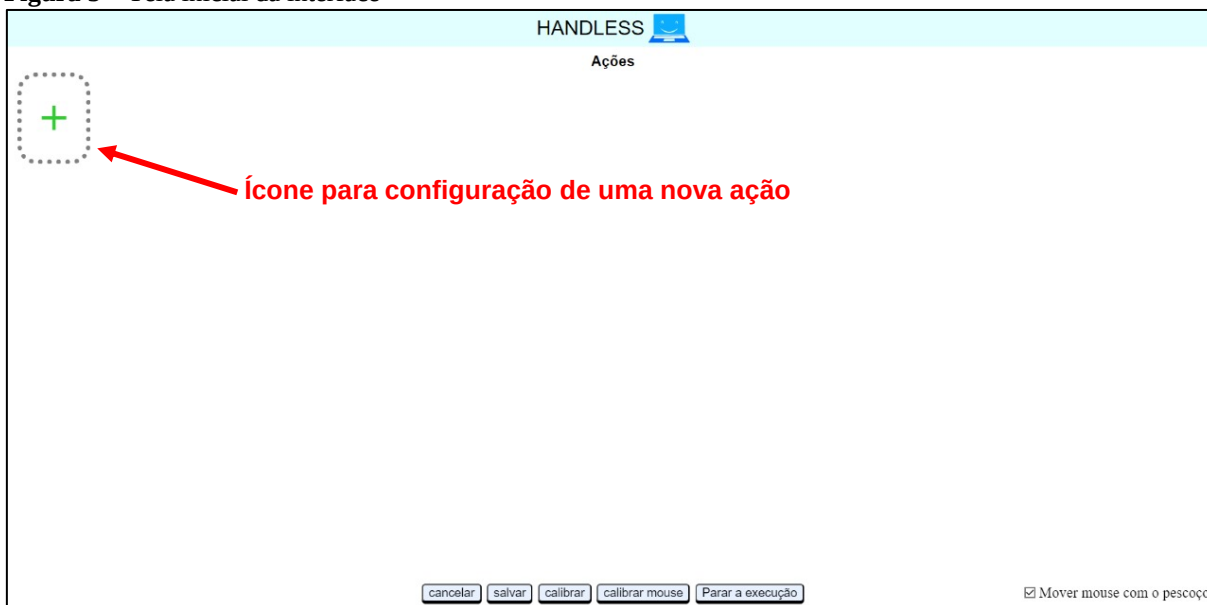


Fonte: Elaborado pelos autores (2023).

A figura 5, por sua vez, apresenta a tela inicial da interface. Nela estão presentes quatro botões: cancelar - que descartará as mudanças feitas e retornará ao último salvamento; salvar - que gravará as mudanças feitas; calibrar - que redefinirá os valores de meta para as expressões; calibrar mouse - que definirá a posição central do pescoço para o movimento do mouse; e há uma caixa de seleção que define se o mouse será ou não movido de acordo com a posição da cabeça.

A configuração da tomada de ações foi construída pensando na facilidade de uso e entendimento. Inicialmente é exibido um ícone de adição, que pode ser utilizado para criar uma nova ação.

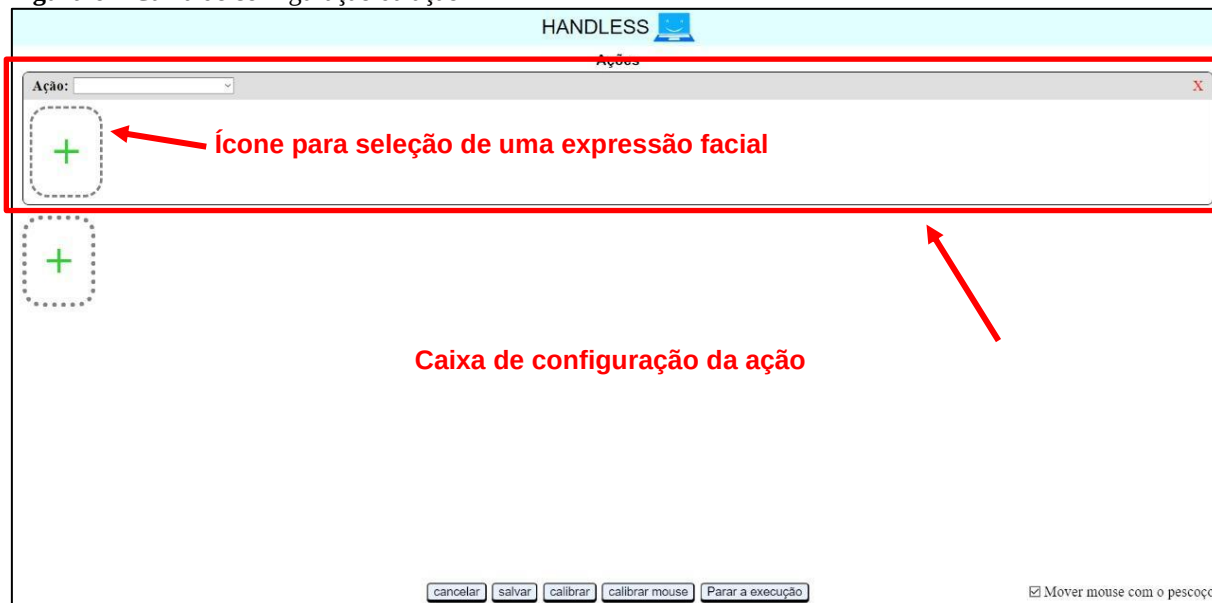
Figura 5 – Tela inicial da interface



Fonte: Elaborado pelos autores (2023).

Ao adicionar uma ação será criado uma caixa de configuração para a mesma, assim como mostra a Figura 6. Ela é composta por uma caixa de seleção para definir a função e a opção de adicionar expressões de ativação.

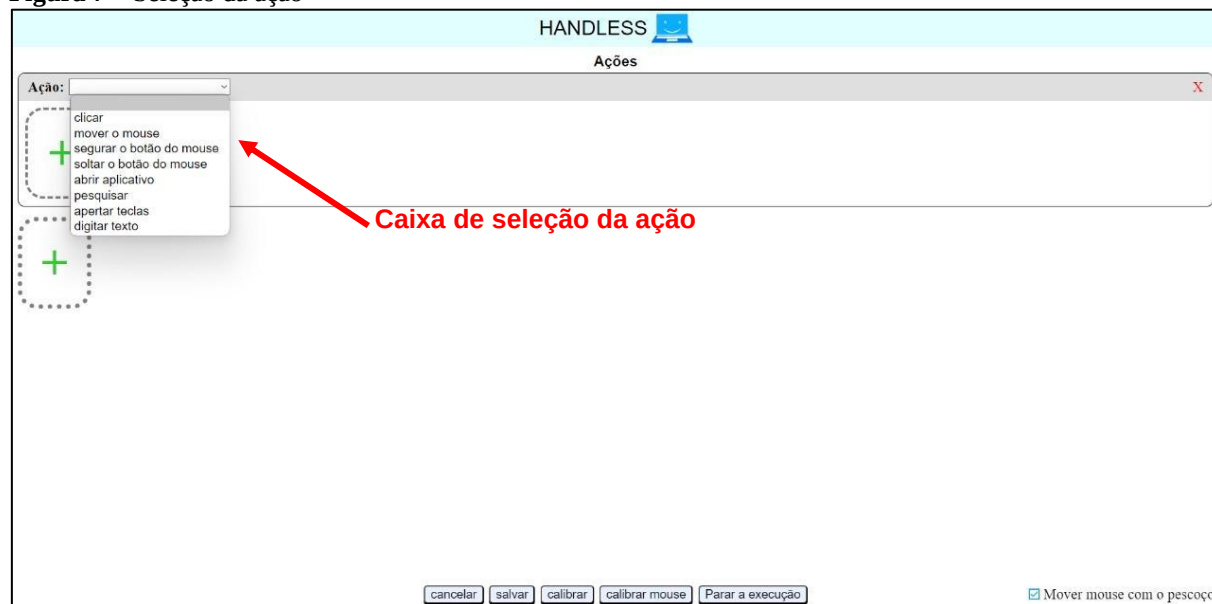
Figura 6 – Caixa de configuração da ação



Fonte: Elaborado pelos autores (2023).

As opções da caixa de seleção podem ser vistas na Figura 7.

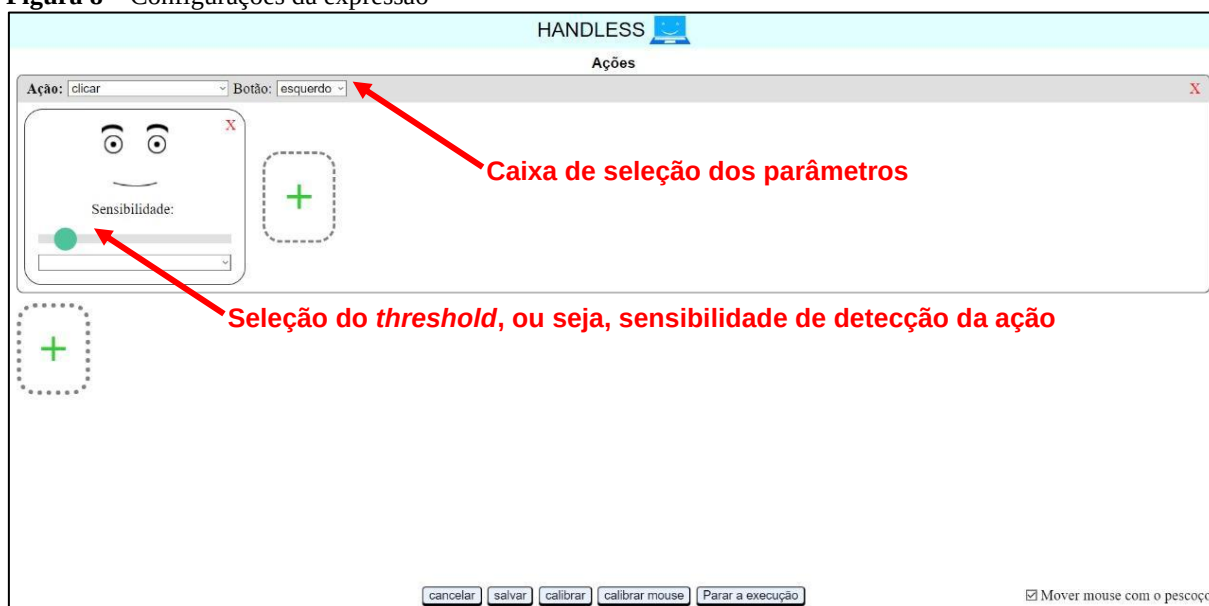
Figura 7 – Seleção da ação



Fonte: Elaborado pelos autores (2023).

Ao selecionar uma ação, os seus respectivos parâmetros serão disponibilizados para configuração. Adicionando uma expressão será mostrado uma caixa contendo uma imagem representativa da mesma, uma barra deslizante que define o valor do *threshold* e uma caixa de seleção com as opções de expressões, sendo tudo demonstrado na Figura 8.

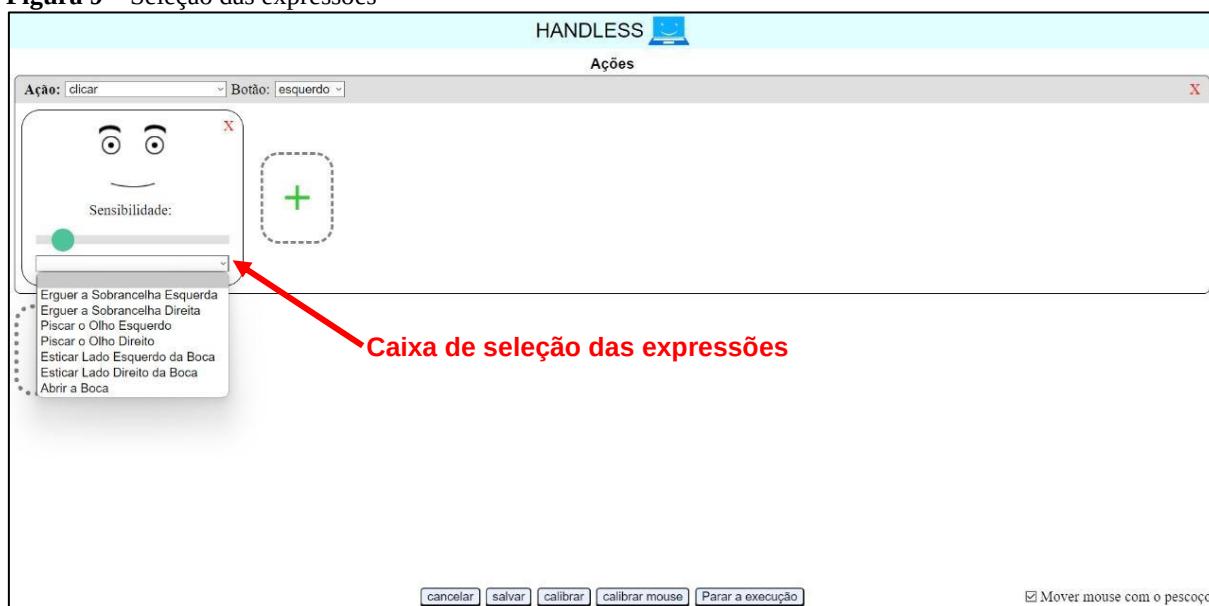
Figura 8 – Configurações da expressão



Fonte: Elaborado pelos autores (2023).

As opções possíveis de expressões estão exibidas na Figura 9.

Figura 9 – Seleção das expressões



Fonte: Elaborado pelos autores (2023).

Desse modo, o algoritmo e interface desenvolvidos são capazes de controlar o computador do usuário por meio das ações e respectivas expressões faciais definidas pelo mesmo, possibilitando o uso mais acessível de aplicativos para pessoas com ausência ou limitações de mobilidade de maneira fácil, intuitiva e amigável.

2.4 Discussões

Nesta pesquisa foi descrito um sistema capaz de utilizar expressões faciais para acessibilidade de computadores. A ferramenta criada demonstra grande utilidade, principalmente para pessoas com nenhuma mobilidade, porém também serve de ajuda para aqueles com mobilidade limitada, uma vez que, caso o usuário consiga utilizar o *mouse*, algumas funções da aplicação se tornam incômodas. O principal diferencial imbuído neste *software* está em sua customização, quando comparados ao projeto proposto por Deepateep e Vichitvejpaisal (2020), já que em sua pesquisa as ações eram travadas a uma só expressão. Em acréscimo a isso, não é necessário nenhum *hardware* específico, sendo utilizado apenas uma *webcam* comum, isso em conjunto a interface simples e intuitiva possibilita um maior alcance da solução. O programa foi executado com êxito em um dispositivo com uma com 16 *gigabytes* de memória *ram*, processador Intel i7 de 9ª geração. Dadas essas características, este dispositivo pode ser considerado um modelo mediano a avançado (Villinger, 2018).

Mesmo assim, alguns pontos são dignos de nota. É necessária uma primeira configuração, portanto, caso o usuário não possua mobilidade dos membros superiores será preciso um assistente para que ele faça um primeiro acesso. Além disso, o fato do *software* se basear exclusivamente em movimentos faciais pode causar exaustão para tarefas mais intensas. Por fim, conforme descrito no item 2, em alguns momentos a aplicação pode apresentar algumas imprecisões, o que pode se provar frustrante em alguns momentos, porém isso não impede o uso da aplicação, apenas sendo necessário tentar novamente.

3 CONCLUSÃO

Dado o objetivo proposto por esta pesquisa, de criar uma forma de acessibilidade para aplicativos, baseada na identificação de expressões faciais, conclui-se que o propósito deste projeto foi alcançado com algumas ressalvas. O reconhecimento de expressões do rosto do usuário foi realizado a partir de um algoritmo para mapeamento da face, utilizando apenas uma câmera simples para captura de imagens. Já a definição da ação associada à articulação do rosto foi realizada por meio de uma interface web, intuitiva e amigável. Um dos principais pontos a ser considerado é a capacidade de adaptação do sistema, já que cada tipo de deficiência possui características únicas, e a capacidade de mutação de programa permite se adequar a elas.

Apesar disto, existem pontos que diminuem a usabilidade do *software* embora o arquivo do código seja relativamente leve possuindo apenas 377 megabytes, ele consome diversos recursos do computador, sendo necessário uma máquina relativamente avançada para executá-lo de maneira satisfatória. Acima disso certas imprecisões causaram desconforto no uso da aplicação. Outra consequência do uso de movimentos faciais foi a fadiga muscular após longos períodos.

Como sugestão para trabalhos futuros, é importante a caracterização de mais expressões faciais para tornar o programa ainda mais robusto. Além disso, vale pontuar, como possível melhoria, uma calibração automática do *threshold* do sistema, com o objetivo de otimizar a detecção das expressões. Também seria de grande valia o aumento da precisão da aplicação em conjunto com a carga de uma primeira configuração, aumentando a independência dos adeptos ao *software*.

REFERÊNCIAS

- AHMED, F. G.; AHMED, H. I.; MIKY, S. F Effect of Physical Disabilities on Quality of life for Soldiers. **Egyptian Journal of Health Care**, v. 15, n. 2, p. 214-227, 2024.
- ALTMAN, B. M. Definitions, concepts, and measures of disability. **Annals of epidemiology**, v. 24, n. 1, p. 2-7, 2014.
- ANNXINGYUAN. **Facemesh**: Face-landmarks-detection [facemesh]. Commit 838611c. São Francisco, 2020. Repositório do GitHub, out. 2020. Disponível em: <https://github.com/tensorflow/tfjs-models/blob/838611c02f51159afdd77469ce67f0e26b7bbb23/face-landmarks-detection/src/mediapipe-facemesh/keypoints.ts>. Acesso em: 01 nov. 2023.
- AWARI. **Matemática com Python**: Aprenda a utilizar a biblioteca math. Blog Awari. Jundiaí, 2022. Disponível em: https://awari.com.br/matematica-com-python-aprenda-a-utilizar-a-biblioteca-math/?utm_source=blog&utm_campaign=projeto+blog&utm_medium=Matemática%20com%20Python:%20Aprenda%20a%20utilizar%20a%20biblioteca%20math. Acesso em 30 out. 2023.
- BALLARD, P.; STOCKMAN, G. C. Controlling a computer via facial aspect. **IEEE Transactions on Systems, Man, and Cybernetics**, v. 25, n. 4, p. 669-677, 2002.
- BIBLIOTECA VIRTUAL EM SAÚDE MINISTÉRIO DA SAÚDE. **11/10 – Dia da Pessoa com Deficiência Física**. [S. l.], [S. d.]. Informação Online. Disponível em: <https://bvsmms.saude.gov.br/o-que-e-a-bvs-ms/>. Acesso em: 22 mai. 2025
- DEEPATEEP, C.; VICHITVEJPaisal, P. Facial Movement Interface for Mobile Devices Using Depth-sensing Camera. In: INTERNATIONAL CONFERENCE ON KNOWLEDGE AND SMART TECHNOLOGY (KST), 12., 2020, Pattaya. **Proceedings [...]**. Nova Iorque: IEEE, 2020. p. 115-120.
- DESMOND, D.; Layton, N.; Bentley, J.; Boot, F. H.; Borg, J.; Dhungana, B. M.; Gallagher, P.; Gitlow, L.; Gowran, R. J.; Groce, N.; Mavrou, K.; Mackeogh, T.; McDonald R.; Pettersson, C.; Scherer, M. J. Assistive technology and people: a position paper from the first global research, innovation and education on assistive technology (GREAT) summit. **Disability and Rehabilitation: Assistive Technology**, v. 13, n. 5, p. 437-444, 2018.
- DEVMEDIA. **Programação assíncrona com Python**. Tutorial publicado em site. Jacarepaguá, 2018. Disponível em: <https://www.devmedia.com.br/programacao-assincrona-com-python/40258>. Acesso em 30 out. 2023.
- DUCKETT, J. **Html & Css: Design and Build Websites**. Indianápolis, Chichester, Inglaterra: Ed. Wiley distributor, 2011.
- EKMAN, P. Facial expression and emotion. Washington, D.C., **American Psychologist**, v. 48, n. 04, p. 384-392, 1993.
- FERNANDES, L. R.; FONTANA, S. S.; FIGUEIREDO, A. da M.; MATTIOLI, P. de S.; SILVA, P. C. D. Human-machine interaction and forms of human communication. **Research, Society and Development**, v. 10, n. 14, p. e90101420777, 2021. Disponível em: <https://rsdjournal.org/index.php/rsd/article/view/20777>. Acesso em: 25 oct. 2023.

FLANAGAN, D.; FERGUSON, P. **JavaScript: The Definitive Guide**. Estados Unidos: Ed. O'Reilly & Associates, 3. ed., 1998.

HOTT, D. F. Martins; FRAZ, J. N. Acessibilidade, tecnologia assistiva e unidades de informação: articulações à realidade da inclusão. **Perspectivas em Ciência da Informação**. v. 24, n. 04, 2019. Disponível em: <https://doi.org/10.1590/1981-5344/4194>. Acesso em: 25 out. 2023.

JI, Y. G.; PARK, J. H.; LEE, C.; YUN, M. H. A Usability Checklist for the Usability Evaluation of Mobile Phone User Interface. **International Journal of Human-Computer Interaction**. v. 20, n. 3, p. 207-231, 2006.

KUKIL. Introduction to MediaPipe. **LearnOpenCV**, 2022. Disponível em: <https://learnopencv.com/introduction-to-mediapipe/>. Acesso em: 30 de outubro de 2023.

LI, X.; DA, F. Efficient 3D face recognition handling facial expression and hair occlusion. **Image and Vision Computing**, v. 30, p. 668-679, 2012. Disponível em: <https://doi.org/10.1016/j.imavis.2012.07.011>. Acesso em 01 nov. 2023.

LUNDH, F.; **Pillow**. Documentação 10.1.0. [S. l.], 2023. Documentação de código computacional da biblioteca Pillow. Disponível em: <https://pillow.readthedocs.io/en/stable/>. Acesso em: 30 out. 2023.

MADHU, M.; AMUTHA, R. A combinatorial approach to human face recognition and expression identification. **Scientific Research and Essays**, v. 8, p. 1464-1470, 2013. Disponível em: <https://doi.org/10.5897/SRE2013.5599>. Acesso em: 01 nov. 2023.

MUDIT, K.; DIVYA, S., K.; JOSHI, S. V. Facial Expression Detection using Convolutional Neural Network. In: 2022 International Conference on Cyber Resilience (ICCR), 2022, Dubai, Emirados Árabes Unidos. **Proceedings [...]**, Nova Iorque: IEEE, 2022. p. 1-5.

NUMPY DEVELOPERS. **Numpy reference**. Versão 1.26. [S. l.], 2023. Documentação de código computacional. Disponível em: <https://numpy.org/doc/stable/reference/>. Acesso em: 30 out. 2023.

OPENCV. **Introduction OpenCV**. Versão 4.8.1. [S. l.], 2023. Documentação de código computacional. Disponível em: <https://docs.opencv.org/4.8.0/d1/dfb/intro.html>. Acesso em: 30 out. 2023.

PANTUZA, G. **O que são e como funcionam as threads**. Blog Pantuza. [S. l.], 2017. Disponível em: <https://blog.pantuza.com/artigos/o-que-sao-e-como-funcionam-as-threads>. Acesso em: 30 out. 2023.

PIRRONE, M.; CENTORRINO, M.; GALLETTA, A.; SICARI, C.; VILLARI, M. Digital Humanities and disability: A systematic literature review of cultural accessibility for people with disability. **Digital Scholarship in the Humanities**, v. 38, n. 1, p. 313-329, abr. 2023.

POTIKANYA, S.; LERTPITHAKSOONTHORN, T.; MEECHAI, A.; MUNGAUAMKLANG, R.; KEAOKAO, P.; SUKJAMSRI, C.; CHIANRABUTRA, C.; CHAROENPONG, T. Development of facial expression recognition by significant sub-region. In: International Conference on Knowledge and Smart Technology (KST). 8., 2016, Chiang Mai, Tailândia. **Proceedings [...]**, Nova Iorque: IEEE. 2016 p. 201-204.

PYTHON PACKAGE INDEX. **A simple framework for building complex web applications**. Versão 3.0.0. [S. l.], 2023. Documentação de código computacional. Disponível em: <https://pypi.org/project/Flask/>. Acesso em: 30 out. 2023.

PYTHON SOFTWARE FOUNDATION. **json** - Codificador e decodificador JSON. Documentação Python 3.12.0. Wilmington, 2023. Disponível em: <https://docs.python.org/pt-br/3/library/json.html>. Acesso em: 30 out. 2023.

PYTHON SOFTWARE FOUNDATION. **OS** - Diversas interfaces de sistema operacional. Documentação Python 3.12.0. Wilmington, 2023. Documentação de código computacional. Disponível em: <https://docs.python.org/pt-br/3/library/os.html>. Acesso em: 30 out. 2023.

PYTHON SOFTWARE FOUNDATION. **OS.PATH** - Common pathname manipulations. Documentação Python 3.12.0. Wilmington, 2023. Disponível em: <https://docs.python.org/3/library/os.path.html>. Acesso em: 30 out. 2023.

PYTHON SOFTWARE FOUNDATION. **Subprocess** - Gerenciamento de subprocessos. Documentação Python 3.12.0. Wilmington, 2023. Documentação de código computacional. Disponível em: <https://docs.python.org/pt-br/dev/library/subprocess.html>. Acesso em: 30 out. 2023.

PYTHON SOFTWARE FOUNDATION. **Sys** - System-specific parameters and functions. Documentação Python 3.12.0. Wilmington, 2023. Documentação de código computacional. Disponível em: <https://docs.python.org/3/library/sys.html>. Acesso em: 30 out. 2023.

PYTHON SOFTWARE FOUNDATION. **Threading** - Paralelismo baseado em Thread. Documentação Python 3.12.0. Wilmington, 2023. Disponível em: <https://docs.python.org/pt-br/3/library/threading.html>. Acesso em: 30 out. 2023.

REVINA, M. I.; EMMANUEL, W. R. S. A Survey on Human Face Expression Recognition Techniques. **Journal of King Saud University - Computer and Information Sciences**, v. 33, n. 6. p. 619-628, 2021.

RIDHA, I.; BELADGHAM, M.; MERIT, K.; TALEB, A.; ALOUANI, I. Facial Expression Recognition Based on DWT Feature for Deep CNN. In: International Conference on Control, Decision and Information Technologies (CoDIT), 6., 2019. Paris, França. **Proceedings [...]**. Nova Iorque: IEEE, 2019. p. 344-348.

SINGH, G.; GUPTA, I.; SINGH, J.; KAUR, N. Face Recognition using Open Source Computer Vision Library (OpenCV) with Python. In: International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO), 10., 2022. Noida, Índia. **Proceedings [...]**. Nova Iorque: IEEE, 2022. p. 1-6.

SU, Y.; SHAN, S.; CHEN, X.; GAO, W.; Hierarchical ensemble of global and local classifiers for face recognition. **IEEE Trans Image Process**. v. 18, n. 8. p. 1885-1896, 2009.

Sweigart A. **Pyautogui Documentation**. [S. l.]. [S. d.]. Documentação de código computacional. Disponível em: <https://pyautogui.readthedocs.io/en/latest/index.html/>. Acesso em 30 out. 2023.

TIEN, M. Z.; SYDYKOVA, D. K.; MEYER, A. G.; WILKE, C. O. Peptide Builder: A simple Python library to generate model peptides. **PeerJ**, v. 1, p. e80, 2013. Disponível em: <https://doi.org/10.7717/peerj.80>. Acesso em: 26 out. 2023.

THOMAS, N.; MATHEW, M. Facial expression recognition system using neural network and MATLAB. In: 2012 International Conference on Computing, Communication and Applications (ICCCA), 2012. Dindigul, Índia. **Proceedings [...]**. Nova Iorque: IEEE, 2012. p. 1-5.

TORRE, F; BLACK, M. Robust Parameterized Component Analysis: theory and applications to 2d facial appearance models. **Computer Vision and Image Understanding**, v. 91, n. 1-2, p. 53-71, 2003.

TOYAMA, K. Look, ma-no hands! hands-free cursor control with real-time 3d face tracking. **PUI98**, v. 24, 1998.

UPALA, M.; WONG, W. K. IoT Solution for Smart Library Using Facial Recognition. **IOP Conference Series: Materials Science and Engineering**, v. 495, n. 1, p. 012030, abr. 2019. Disponível em: <https://dx.doi.org/10.1088/1757-899X/495/1/012030>. Acesso em: 26 out. 2023.

VADLAPATI, J.; VELAN, S. S.; VARGHESE, E. Facial Recognition using the OpenCV Libraries of Python for the Pictures of Human Faces Wearing Face Masks during the COVID-19 Pandemic. In: International Conference on Computing Communication and Networking Technologies (ICCCNT), 12., 2021, Kharagpur, India. **Proceedings [...]**, Nova Iorque: IEEE. p. 1-5, 2021.

VALENTIM, P. A.; BARRETO, F.; MUCHALUAT-SAADE, D. C.. Towards Affective TV with Facial Expression Recognition. In: Life Improvement In Quality By Ubiquitous Experiences Workshop (LIQUE), 1., 2021, New York. **Anais [...]**. Porto Alegre: Sociedade Brasileira de Computação, 2021.

VARGHESE, A.; VARGHESE, R.; KANNAN, B.; PAUL, J. Identification of Region of Interest using Local Binary Pattern with Ternary Encoding. **Digital Image Processing**, v. 3, p. 1050-1052, 2011.

VILLINGER, S. **De quanta potência de computador você realmente precisa?**. Blog AVG Signal. [S. l.], 2018. Disponível em: <https://www.avg.com/pt/signal/how-powerful-a-computer-do-you-really-need>. Acesso em: 25 mai. 2025.