

# Sistema de identificação de expressões faciais para acesso as funcionalidades no uso de aplicações

ENZO LUIZ TSUTSUMI DE ALMEIDA JOSÉ<sup>1</sup>; LUIZ GUSTAVO FERNANDES SALVADOR<sup>1</sup>; MARCEL SANTOS SILVA<sup>2</sup>; ELOÍZA MARTINS PRIMO CAPELOCI<sup>2</sup>.

<sup>1</sup> Discente em Big Data na FATEC Pompeia “Shunji Nishimura”, Pompeia-SP, [enzo.jose@fatec.sp.gov.br](mailto:enzo.jose@fatec.sp.gov.br); [luz.salvador@fatec.sp.gov.br](mailto:luz.salvador@fatec.sp.gov.br).

<sup>2</sup> Docentes do curso Big Data no Agronegócio, FATEC Pompeia, Pompeia-SP, [marcel.silva9@fatec.sp.gov.br](mailto:marcel.silva9@fatec.sp.gov.br); [eloiza.capeloci@fatec.sp.gov.br](mailto:eloiza.capeloci@fatec.sp.gov.br).

**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 incrementar ainda mais essa importância, utilizando das expressões faciais para acesso às funcionalidades no uso de aplicações. 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.

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

## 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 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 e 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).

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 Principais<sup>1</sup>, 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 *PSA*<sup>1</sup>, 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 Cinza<sup>2</sup> 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 top 10 imagens de uma base de dados. Por fim, a *PSA* e o Padrão Ternário Local<sup>3</sup> são utilizados para se definir a expressão facial.

Para escolha do método a ser utilizado, esta pesquisa prezou por:

---

<sup>1</sup> 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).

<sup>2</sup> Do inglês, Gray Level Co-occurrence Matrix – *GLCM*.

<sup>3</sup> 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).

- Simplicidade e eficiência - interesse pela utilização de um método de baixa complexidade, permitindo o processamento rápido e eficiente, exigindo poucos recursos computacionais;
- 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 e MATHEW, 2012; MUDIT, DYVIA e JOSHI, 2022), no entanto, essas exigem grandes volumes de amostras de treinamento, o que foi evitado pelo autor;
- Transparência Analítica - houve preferência por um método que determinasse com clareza a expressão apresentada pelo usuário;
- Adaptabilidade - um sistema personalizável, de acordo com as necessidades e desejos do usuário, também foi uma das prerrogativas deste trabalho;
- 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.

Dessa 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 representação à face do usuário, limitada a regiões pré-definidas, sendo elas: sobrancelha, boca, olhos e nariz<sup>4</sup>; já a segunda parte é responsável por gerar a interface visual onde é possível configurar as ações que o código executará.

No contexto do sistema de identificação de expressões faciais, adotou-se uma metodologia de análise que recai sobre a determinação das expressões faciais mediante a caracterização geométrica do espaço facial, i.e., 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, sobretudo ponderadas por um limiar predefinido, chamado de *Threshold*. A manifestação de uma expressão facial é, então, validada

---

<sup>4</sup> Foram escolhidas essas regiões da face por serem de fácil identificação para o algoritmo e por serem de fácil movimentação para o usuário.

quando a magnitude resultante, em módulo, dessa ponderação ultrapassa essa constante arbitrariamente definida.

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*.

O objetivo deste trabalho consiste no desenvolvimento uma aplicação para identificar expressões faciais de forma simples, e utilizá-las como ferramenta de melhoria à 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, com baixo requisito computacional, proporcionando uma experiência de usuário mais inclusiva e eficiente.

## **MATERIAL E MÉTODOS**

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 e WONG, 2019; VADLAPATI, VELA, e 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 deste trabalho, 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*, i.e., 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 ferramenta poderosa para realizar cálculos matemáticos. Ela oferece acesso às funções matemáticas definidas pelo padrão C, sendo especialmente úteis

para operações numéricas, como arredondamento, cálculo de fatorial, exponenciação e muito mais (FERREIRA, 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 ferramenta poderosa 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 simples (ROBIN, 2023). Para este trabalho, ela foi utilizada para fazer o controle do mouse e teclado;

A *os* é uma ferramenta 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). Utilizada neste trabalho 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). Utilizada neste trabalho para salvar e carregar os arquivos json de configuração;

A *threading* permite que os programadores criem, iniciem e gerenciem *threads*, i.e., fluxos de um programa em execução (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). Neste trabalho, 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). Neste trabalho, 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 vetores de armazenamento (NUMPY, 2023). Neste trabalho, 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;

A *flask* é a base para construção de softwares (*frameworks*) leve para aplicação *web*, sendo projetada para facilitar o início rápido de projetos *web*, com a capacidade de escalar para aplicações mais complexas (PYTHON, 2023). Neste trabalho, ela foi utilizada para servir 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 atualização 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



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 Sobancelha Esquerda	158; 145	Maior que o <i>threshold</i>
Erguer A Sobancelha 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.

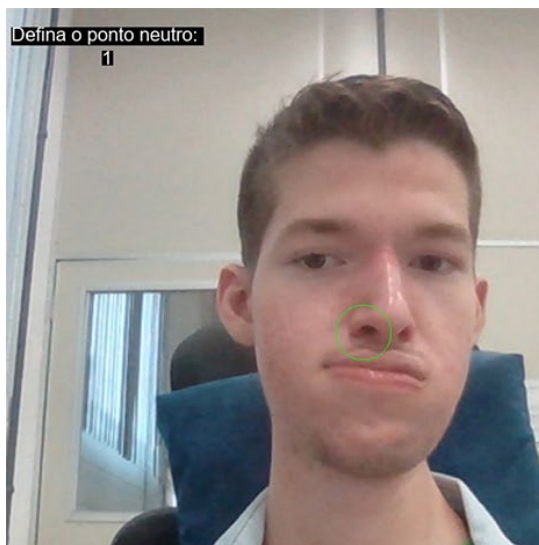
**Tabela 2. Ações, respectivos efeitos e parâmetros a serem definidos para sua execução.**

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

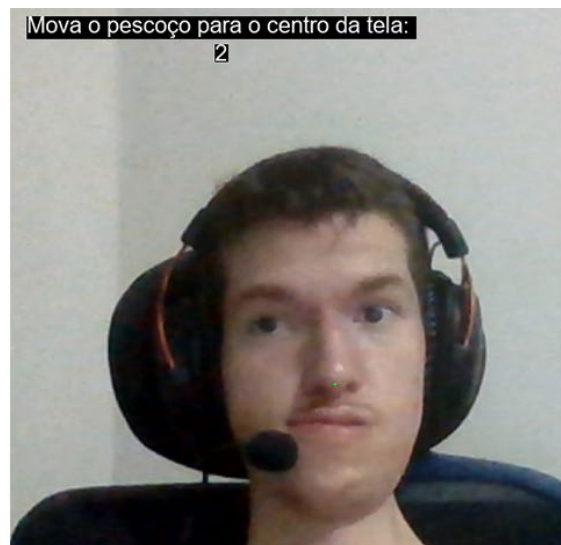
Fonte: Elaborado pelos autores (2023)

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**



a)



b)



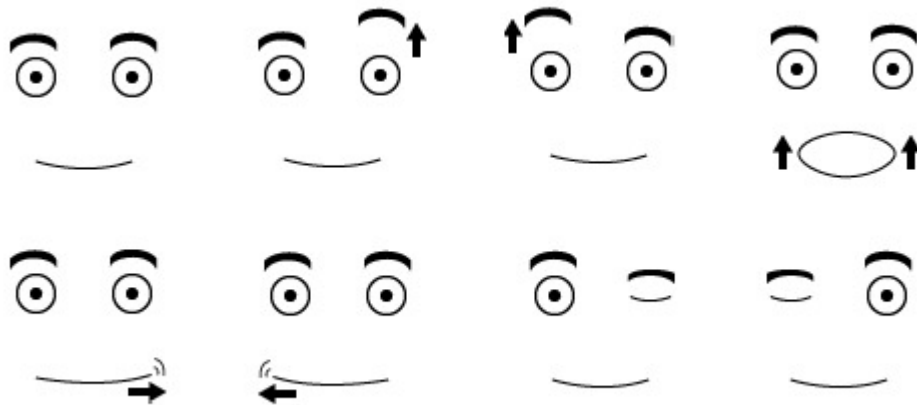
c)

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**

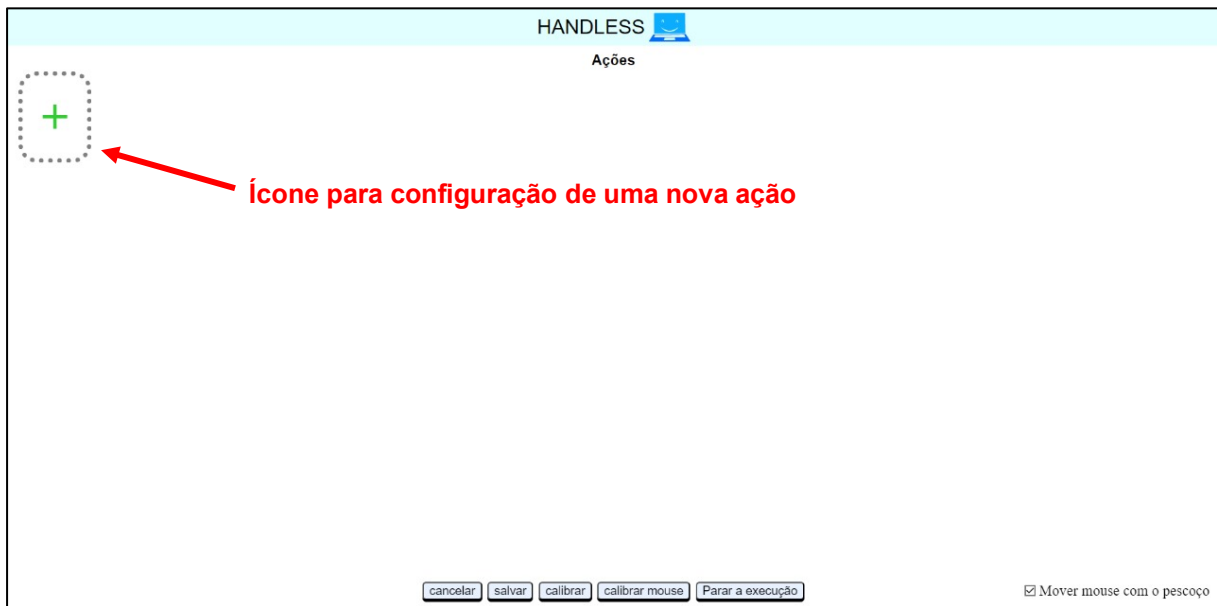


Fonte: Elaborado pelos autores (2023)

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.

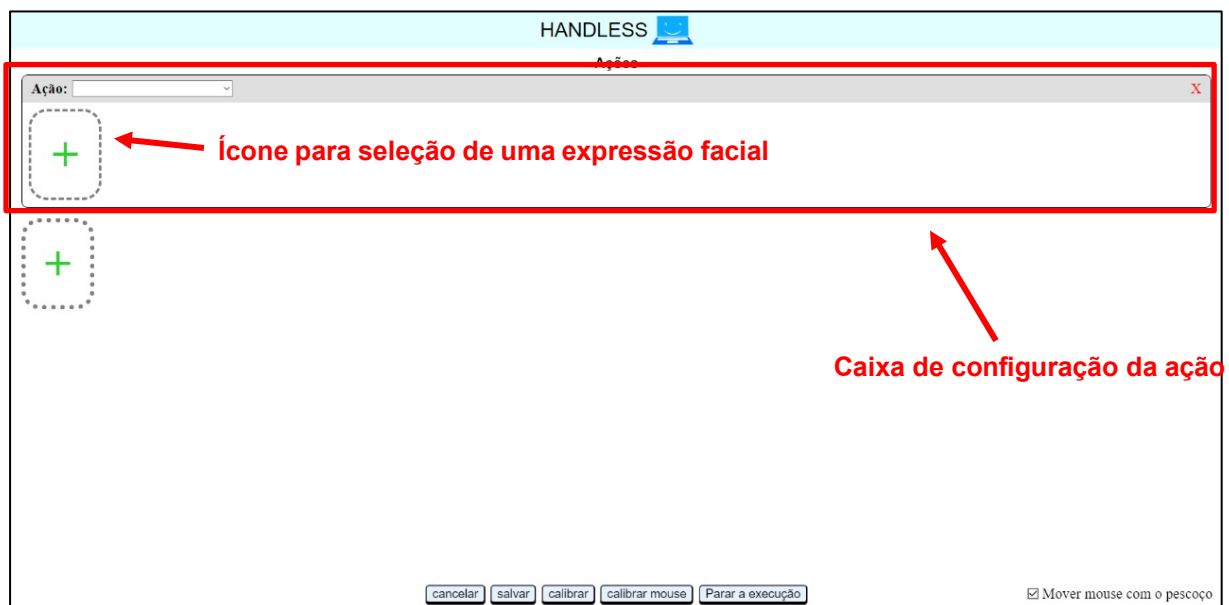
**Figura 5 – Tela inicial da interface**



Fonte: Elaborado pelos autores (2023)

Ao adicionar uma ação será criada 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.

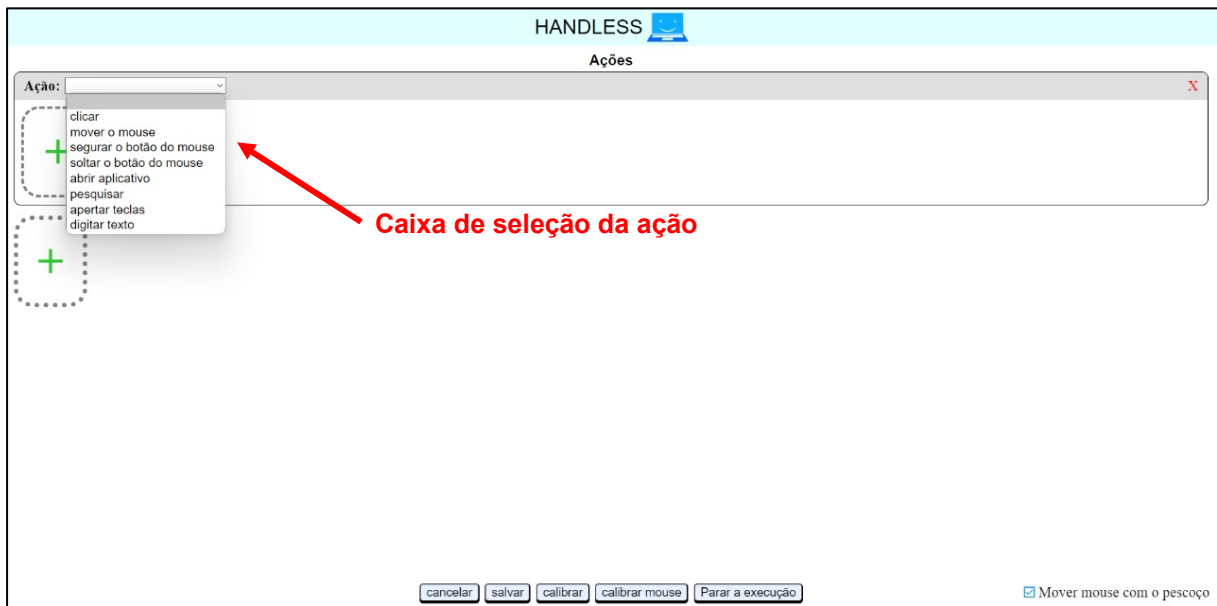
**Figura 6 – Caixa de configuração da ação**



Fonte: Elaborado pelos autores (2023)

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

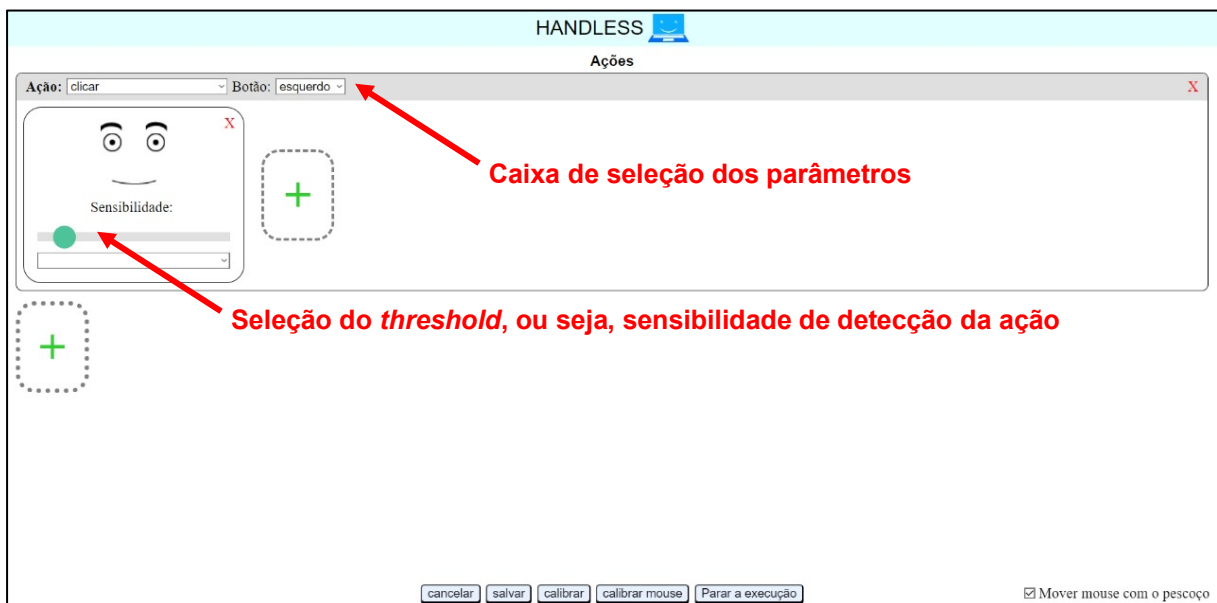
**Figura 7 – Seleção da ação**



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.

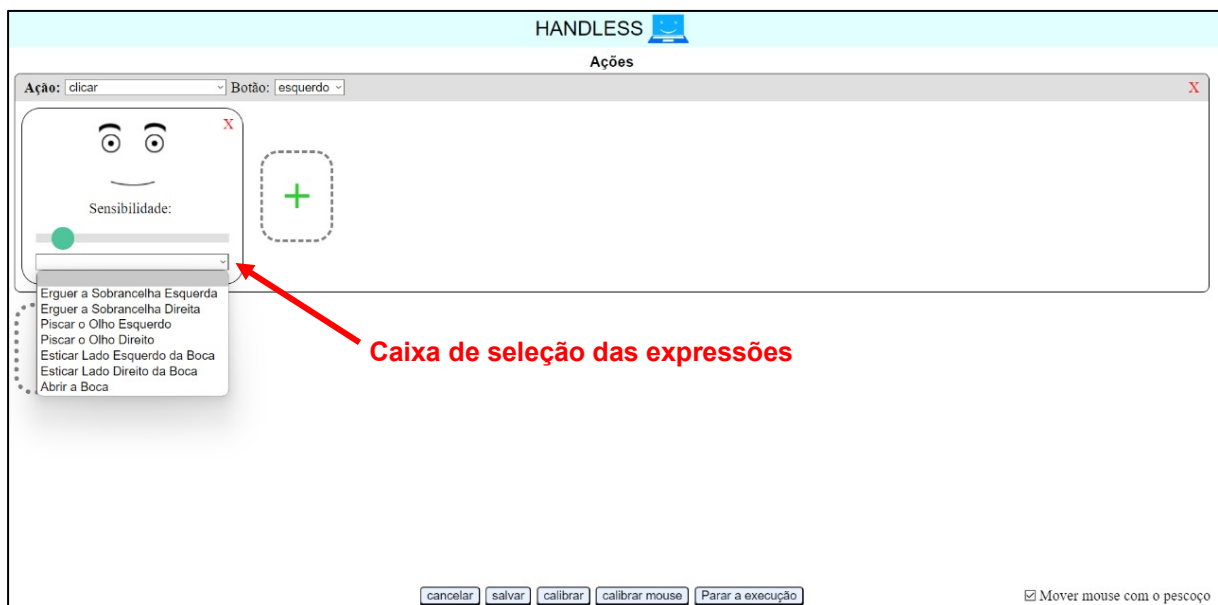
**Figura 8 – Configurações da expressão**



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 definidas pelo mesmo, possibilitando o uso mais acessível de aplicativos para pessoas com ausência ou limitações de mobilidade de maneira fácil, intuitiva e amigável.

## CONCLUSÕES

Dado o objetivo proposto por esta pesquisa de criar uma forma de acessibilidade para aplicativos multiplataformas, baseada na identificação de expressões faciais, conclui-se que o propósito deste projeto foi alcançado. O reconhecimento de expressões do rosto do usuário foi realizado, de forma eficiente, a partir de um algoritmo para mapeamento da face. Já a definição da ação associada à articulação do rosto é realizada por meio de uma interface *web*, intuitiva e amigável. Além das presentes constatações, também é possível alegar a adaptabilidade do programa desenvolvido, uma vez que fica totalmente a cargo do usuário a definição da sensibilidade para detecção de uma ação e a definição de qual função a expressão será responsável.

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.

## AGRADECIMENTOS

Primeiramente, gostaria de agradecer a Deus por todas as portas que me abriu e por toda a ajuda quando eu precisei. Em segundo lugar, gostaria de agradecer à minha família, por toda a força e motivação para eu conseguir chegar até aqui. Por fim, agradeço aos professores e amigos que tive o prazer de encontrar na FATEC Pompeia.

## REFERÊNCIAS

ANNXINGYUAN. **Facemesh**: Face-landmarks-detection [facemesh]. 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.

DUCKETT, J. **Html & Css: Design and Build Websites**. Indianápolis, Chichester, Inglaterra: Ed. Wiley distributor, 2011.

EKMAN, P. **Facial expression and emotion**. *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.

FERREIRA, P; **Matemática com Python**: Aprenda a utilizar a biblioteca math; Awari. Ago. 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](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.

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, [S.l.], v. 24, n. 04, 2019. Disponível em: <<https://doi.org/10.1590/1981-5344/4194>>. Acesso em: 25 out. 2023.

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

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

**LUNDH, F.; Pillow.** Documentação 10.1.0. 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.** 2022 International Conference on Cyber Resilience (ICCR). Dubai, Emirados Árabes Unidos, p. 1-5, 2022.

**NUMPY reference. NumPy Developers.** Versão 1.26. Set. 2023. Disponível em: <<https://numpy.org/doc/stable/reference/>>. Acesso em: 30 out. 2023.

**OPENCV. Introduction OpenCV.** 2023. Disponível em: <<https://docs.opencv.org/4.8.0/d1/dfb/intro.html>>. Acesso em: 30 out. 2023.

**OS - Diversas interfaces de sistema operacional.** Python Software Foundation. Documentação Python 3.12.0. 2023. Disponível em: <<https://docs.python.org/pt-br/3/library/os.html>>. Acesso em: 30 out. 2023.

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

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

**PYTHON PACKAGE INDEX. A simple framework for building complex web applications.** Versão 3.0.0. Set. 2023. Disponível em: <<https://pypi.org/project/Flask/>>. 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**. 2016 8th International Conference on Knowledge and Smart Technology (KST). p. 201-204, 2016.

DEVMEDIA. **Programação assíncrona com Python**. DevMedia. 2018. Disponível em: <<https://www.devmedia.com.br/programacao-assincrona-com-python/40258>>. 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**. P. 344-348, 2019. Disponível em: <<https://doi.org/10.3390/ELECTRONICS8030324>>. Acesso em: 01 nov. 2023.

ROBIN, A.; **Pyautogui Documentation?** Top 6 Best Answers. BrandisCrfts. Ago. 2023. Disponível em: <<https://brandiscrafts.com/pyautogui-documentation-top-6-best-answers/>>. Acesso em 30 out. 2023.

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

SU, Y.; SHAN, S.; CHEN, X.; GAO, W;. **Hierarchical ensemble of global and local classifiers for face recognition**. IEEE Trans Image Process. p. 1885-1896, ago. 2019.

SUBPROCESS - **Gerenciamento de subprocessos**. Python Software Foundation. Documentação Python 3.12.0. 2023. Disponível em: <<https://docs.python.org/pt-br/dev/library/subprocess.html>>. Acesso em: 30 out. 2023.

SYS - **System-specific parameters and functions**. Python Software Foundation. Documentação Python 3.12.0. 2023. Disponível em: <<https://docs.python.org/3/library/sys.html>>. Acesso em: 30 out. 2023.

THOMAS, N.; MATHEW, M. **Facial expression recognition system using neural network and MATLAB**. 2012 International Conference on Computing, Communication and Applications, p. 1-5, 2012. Disponível em: <<https://doi.org/10.1109/ICCCA.2012.6179169>>. Acesso em: 01 nov. 2023.

THREADING - **Paralelismo baseado em Thread**. Python Software Foundation. Documentação Python 3.12.0. 2023. Disponível em: <<https://docs.python.org/pt-br/3/library/threading.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.

TORRE, F; BLACK, M. **Robust Parameterized Component Analysis.** P. 653-669, 2002. Disponível em: <[https://doi.org/10.1007/3-540-47979-1\\_44](https://doi.org/10.1007/3-540-47979-1_44)>. Acesso em: 01 nov. 2023.

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: 12th International Conference on Computing Communication and Networking Technologies (ICCCNT), 2021, Kharagpur, India. Proceedings..., 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), 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.

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