

Capítulo

11

Muppets: Motion Puppets

André Lyra Fernandes e Victória Carolina Ferreira Da Silva

Abstract

The "Muppets: Motion Puppets" project is a teaching tool that, through computer vision (OpenCV and MediaPipe) in Python, allows Human-Computer Interaction (HCI) via hand gestures. Focused on creating a Graphical User Interface (GUI) based on Natural User Interfaces (NUI), the system transforms the hand into an intuitive control for classes. It offers a "transparent whiteboard" for drawing and an interactive menu, aiming at greater engagement and accessibility.

Resumo

O projeto "Muppets: Motion Puppets" é uma ferramenta didática que, por meio de visão computacional (OpenCV e MediaPipe) em Python, permite a interação Humano-Computador (IHC) via gestos manuais. Focado em criar uma Interface Gráfica do Usuário (GUI) baseada em Interfaces Naturais de Usuário (NUI), o sistema transforma a mão em um controle intuitivo para aulas. Ele oferece uma "lousa transparente" para desenhar e um menu interativo, visando maior engajamento e acessibilidade.

1.1. Introdução

Com o avanço das tecnologias em diversos contextos da vida, a área da educação também é afetada. Em um cenário onde alunos estão frequentemente expostos a múltiplos estímulos digitais, parte-se da hipótese que o método tradicional de ensino pode não ser o suficiente para manter o engajamento de todos os alunos. Desta maneira, é evidente o uso de recursos tecnológicos no contexto acadêmico, como o uso de projetores e computadores. No entanto, tais recursos, se mal utilizados, podem tornar a aula ainda mais monótona, com apenas slides passando, e perdendo o dinamismo de um professor escrevendo na lousa, onde o aluno tem a possibilidade de acompanhar e participar do raciocínio sendo construído ao poucos, e não apenas um aprendizado passivo, onde o mesmo é apenas exposto de forma massante a diversos slides em sequência já elaborados.

O projeto “Motion Puppets”, ou simplesmente “Muppets” pode ser de alta utilidade para fins didáticos neste contexto, com uso em aulas presenciais que já utilizam recursos tecnológicos para facilitar o ensino, como projetores e notebooks, ou simplesmente em transmissões remotas para ensino a distância. Além disso, é um recurso mais interativo do que os slides tradicionais, que pode cativar o interesse dos alunos e tornar todo ambiente de aprendizado mais lúdico.

O projeto em desenvolvimento tem como objetivo aplicar técnicas de visão computacional utilizando a linguagem Python, com o uso da biblioteca OpenCV (Open Source Computer Vision Library) e Media Pipe . Trata-se de uma aplicação com fins educativos, voltada principalmente para o meio acadêmico.

O sistema proposto permite a interação com o computador por meio de gestos das mãos captados por uma câmera. Dessa forma, o usuário poderá emular funções como o uso de um mouse, acesso a aplicativos, desenhar formas geométricas na tela, apenas com o movimento das mãos. A proposta é transformar a mão do usuário em uma interface de controle interativa, acessível e didática.

1.2. Escopo

Este projeto tem como objetivo o desenvolvimento de uma aplicação de visão computacional em Python que permita aos usuários interagirem com a tela de maneira intuitiva e sem contato físico, utilizando apenas gestos das mãos. A proposta está fortemente fundamentada em princípios da Interação Humano-Computador (IHC), especialmente no que se refere à usabilidade e interação natural.

A IHC é uma área multidisciplinar que estuda a forma como os seres humanos interagem com sistemas computacionais, buscando tornar essa interação mais eficiente, compreensível e

inclusiva. Nesse contexto, a aplicação se propõe a oferecer uma alternativa aos métodos tradicionais de interação (como mouse e teclado), proporcionando uma interface mais ergonômica e intuitiva, que respeita às limitações físicas e cognitivas dos usuários.

A ferramenta tem foco educacional e é voltada para professores, permitindo que eles apontem, desenhem, grifem e manipulem elementos de apresentações, códigos ou qualquer conteúdo exibido na tela por meio de gestos manuais. Isso promove maior liberdade de movimento, engajamento com o conteúdo e inclusão de docentes que possam ter limitações no uso de periféricos físicos.

1.3.1. Objetivos

Desenvolver uma aplicação funcional baseada em visão computacional com detecção de gestos manuais em tempo real, voltada para uso em ambientes didáticos, tanto presenciais quanto online. A proposta busca oferecer uma interface intuitiva e acessível, que dispense a necessidade de especialização ou treinamento prévio por parte do usuário.

O desenvolvimento da aplicação está fundamentado nos princípios da Interação Humano-Computador (IHC), e no conceito de Interfaces Naturais de Usuário (NUI). O objetivo é proporcionar uma experiência de uso mais fluida, ergonômica e próxima das interações humanas naturais, permitindo que professores e demais usuários possam interagir com o conteúdo em tela de forma eficiente, utilizando apenas gestos das mãos, sem a necessidade de dispositivos periféricos tradicionais.

1.3.2. Objetivos Específicos

- Integrar bibliotecas de visão computacional como OpenCV e MediaPipe para detecção e rastreamento de mãos.
- Implementar funcionalidades de interação com a tela (overlay), como apontar, desenhar e desfazer ações.
- Criar um sistema de alternância entre modos (mouse/paint) com base em gestos.
- Desenvolver uma interface (GUI) se necessário.
- Testar a aplicação em cenários simulados de uso em sala de aula.

1.4. Bibliografia Inicial

- AI Virtual Painter | OpenCV Python | Computer Vision
- MediaPipe-Hands
- Real-Time Hand Tracking and Gesture Recognition with MediaPipe: Rerun Showcase
- OpenCV documentation
- OpenCV: OpenCV-Python Tutorials
- Python documentation
- Tkinter — Python interface to Tcl/Tk

1.5. Plano de Trabalho

Etapa	Descrição
1. Pesquisa Inicial	Levantamento de bibliotecas e estudos de caso similares
2. Setup de Ambiente	Configuração do Python e <code>yaml</code> (OpenCV, MediaPipe, etc)
3. Detecção de Mão	Implementação da detecção de mão
4. Mapeamento de Gestos	Tradução dos gestos em comandos (apontar, desenhar etc.)
5. Modo Overlay Paint	Implementar o modo de desenho/grifo com cores e desfazer
6. Integração com Tela	Sobreposição de interface em outras janelas ou slides
7. Testes e Ajustes	Testes práticos e refino dos gestos
8. Documentação e Apresentação	Finalização de documentação e material para seminário

Figura 1: Plano de trabalho inicial. Fonte: Os autores(2025).

2. Fundamentação Teórica

O projeto apresentado tem como objetivo utilizar técnicas de visão computacional para simular o uso do mouse a partir do reconhecimento de gestos captados por uma câmera.

De acordo com Cordeiro (2020), após a pandemia, os avanços nas tecnologias digitais de ensino impulsionaram o desenvolvimento do ensino remoto e ampliaram as possibilidades de transformar o ambiente educacional de maneira dinâmica, consolidando a tecnologia como um pilar fundamental na formação educacional contemporânea.

Nesse contexto, a interação por gestos das mãos surge como uma alternativa mais natural e intuitiva para o controle do computador, levando como base conceitos de Interação Humano Computador (IHC). Essa abordagem oferece maior flexibilidade, assim sendo mais intuitivo e dispensando a necessidade de dispositivos físicos tradicionais, não exigindo treinamento prévio do usuário e pode servir como uma ferramenta de apoio ao ensino, especialmente em ambientes acadêmicos que buscam promover acessibilidade, inovação e novas formas de engajamento educacional.

As técnicas de manipulação gestual variam conforme o contexto do problema. Algumas abordagens utilizam métodos baseados em Visão Computacional (VC), processando imagens dos gestos captados para interpretar e executar ações específicas no computador.

A fundamentação do projeto baseia-se na revisão bibliográfica e na análise de projetos similares disponíveis no GitHub, plataforma de hospedagem de repositórios voltada para projetos de computação. Além disso, são consideradas as documentações oficiais das bibliotecas que serão utilizadas, vídeos tutoriais, palestras e demonstrações práticas sobre o

uso de aplicações similares. Entre as principais fontes teóricas destacam-se o trabalho de conclusão de curso de Favero (2024) intitulado “Reconhecimento e personalização de gestos das mãos para interação com páginas web por meio de visão computacional e aprendizado de máquina” e a publicação de NAOUM (2024) “Real-Time Hand Tracking and Gesture Recognition with MediaPipe: Rerun Showcase”, ou em tradução livre, “Rastreamento de mãos e reconhecimento de gestos em tempo real com MediaPipe: apresentação de reprises”, disponível no site Medium.

2.1. Tecnologias Utilizadas

A principal linguagem utilizada no desenvolvimento do projeto é o Python, na versão 3.10.16, compatível com as bibliotecas adotadas. A escolha por essa linguagem se justifica pela ampla quantidade de bibliotecas voltadas para visão computacional, como OpenCV e MediaPipe, que oferecem recursos robustos e eficientes para o processamento de imagens em tempo real. Além disso, o Python possui uma sintaxe simples e de fácil compreensão, o que facilita a leitura, manutenção e desenvolvimento do código. Atualmente, é considerada a linguagem mais utilizada em projetos de visão computacional e inteligência artificial, sendo adotada tanto na pesquisa quanto na indústria. Por fim, para armazenamento e versionamento do repositório, utilizou-se Git e Github.

2.1.2. Bibliotecas

As principais bibliotecas utilizadas no desenvolvimento deste projeto são o OpenCV, NumPy, MediaPipe e PyQt5. A configuração do ambiente foi realizada com base no arquivo “ambienteMuppets.yml”

A biblioteca OpenCV (Open Source Computer Vision Library), na versão 4.9.0.80, é uma biblioteca de código aberto voltada para aplicações de visão computacional. Inicialmente desenvolvida pela Intel, atualmente é mantida por uma ampla comunidade e possui suporte para diversas linguagens de programação, sendo o Python uma das mais utilizadas. Sua escolha se justifica pela vasta gama de funcionalidades que oferece, como detecção de objetos e rastreamento de movimento, entre outras operações.

Em conjunto com o OpenCV, utiliza-se o NumPy, na versão 1.26.4, uma biblioteca essencial para a manipulação eficiente de arrays e matrizes multidimensionais. Em visão computacional, as imagens são tratadas como matrizes de pixels, e o NumPy facilita operações matemáticas e estatísticas sobre esses dados, no caso do projeto, ao calcular as distâncias dos dedos para permitir os comandos.

A biblioteca MediaPipe, utilizada na versão 0.10.9, é desenvolvida pelo Google e representa uma plataforma de código aberto que facilita a criação de pipelines de processamento multimídia em tempo real, incluindo aplicações de visão computacional e aprendizado de máquina. Ela é amplamente utilizada para tarefas como rastreamento de mãos, reconhecimento de gestos, detecção facial, segmentação de fundo e muito mais. No contexto do projeto, a MediaPipe é especialmente valiosa para o rastreamento preciso e em tempo real dos movimentos das mãos, fornecendo modelos pré-treinados e otimizações que permitem capturar os gestos diretamente pela webcam.

Além dessas, utiliza-se a biblioteca PyQt5, versão 5.15.10, para construção da interface gráfica do sistema. Inicialmente tentou-se a instalação via Conta, porém foram encontrados erros relacionados ao plugin da plataforma Qt ("qt.qpa.plugin: Could not load the Qt platform plugin 'windows'..."). A solução adotada foi a instalação via pip.

2.1.3. Ambiente de desenvolvimento:

Por questões de praticidade e facilidade de uso, a IDE escolhida para o desenvolvimento do projeto foi o Visual Studio Code (VS Code). Essa plataforma de desenvolvimento possui suporte a múltiplas linguagens de programação, incluindo o Python e suas principais bibliotecas, além de contar com uma ampla variedade de extensões que facilitam o desenvolvimento e gerenciamento do ambiente.

Em conjunto com o VS Code, será utilizada a distribuição Anaconda, uma plataforma gratuita e de código aberto voltada para projetos de ciência de dados, aprendizado de máquina e computação científica. O Anaconda facilita a instalação, gerenciamento e atualização de bibliotecas e ambientes virtuais, sendo ideal para projetos que envolvem visão computacional.

2.2 Anatomia dos Gestos e Interpretação Biológica

Para compreender o funcionamento da aplicação desenvolvida, é necessário entender como a anatomia da mão humana se relaciona com os processos de interpretação computacional de gestos. A proposta parte da integração entre os conhecimentos de Interação Humano-Computador (IHC) e o design de Interfaces Gráficas com o Usuário (GUI), visando a construção de interações baseadas em gestos naturais e intuitivos.

A interface foi projetada para interpretar movimentos reais da mão, e sua eficácia depende diretamente da detecção precisa das estruturas anatômicas, como articulações, falanges e a palma. Essa associação entre forma biológica e leitura computacional é o que torna possível traduzir gestos físicos em comandos digitais de forma fluida e responsiva.

2.2.1 Estrutura Anatômica Funcional da Mão

A mão humana é o membro mais distante do membro superior (estrutura anatômica que compreende o ombro, braço, antebraço e mão). A mão é um conjunto de ossos sobre músculos e estruturas neurovasculares.

Os músculos são os principais responsáveis pelas nossas ações do dia a dia, como pegar um copo até um alpinista se segurando em rochas. Sendo divididos por músculos extrínsecos e músculos intrínsecos. Os músculos extrínsecos são o grupo muscular que começa desde o antebraço e vai até as pontas dos dedos, são os responsáveis pelos movimentos que exigem flexão e extensão da mão e dos dedos. Enquanto os músculos intrínsecos, são o grupo muscular que tem início mão, não estendendo até o antebraço, diferente do extrínsecos. São responsáveis pelos movimentos que necessitam de maior precisão, mais delicadeza.

Os ossos são divididos em três grupos sendo: ossos do carpo, ossos do metacarpo e falanges. Ossos do carpo, são um conjunto de 8 ossos que são o agrupamento que formam o pulso humano. Os ossos do metacarpo são a parte intermediária do esqueleto da mão, ligando as falanges e o carpo. São um conjunto de 5 ossos longos e finos.

Os ossos da falange são os que formam os dedos da mão, na mão cada dedo (exceto o polegar) tem três falanges: proximal, média e distal. O polegar possui apenas duas falanges: proximal e distal.



Figura 2: Conjunto de ossos da mão humana. Fonte: ResearchGate

2.2.2 Construção Natural dos Gestos

O ser humano é um ser multi-verbal, ou seja, capaz de se comunicar de diversas formas: por meio da linguagem verbal, não verbal, expressões artísticas, sinais corporais, entre outros. O foco desta pesquisa é a comunicação não verbal, especialmente aquela que ocorre por gestos, movimentos das mãos, cabeça ou outras partes do corpo que transmitem significados sem o uso da fala.

Segundo Lima e Santos (2012), “os gestos naturais constituem a primeira ferramenta de comunicação simbólica”, o que evidencia que, desde os primórdios, o ser humano utiliza gestos como forma de expressão. Desde o nascimento, já somos capazes de nos comunicar por meio de movimentos, antes mesmo do domínio da linguagem falada.

Exemplos simples como o gesto de “joinha” ou o sinal de “ok” demonstram como os gestos se tornaram universais em certas culturas. Além disso, sistemas estruturados como a língua de sinais com variações específicas para cada país são fundamentais para garantir a comunicação de pessoas surdas, promovendo acessibilidade e inclusão social.

Os gestos, portanto, são uma forma natural e instintiva de expressão, presentes desde a infância até a vida adulta. A sociedade evoluiu e se organizou também graças à capacidade de transmitir informações por meio de gestos, tornando essa forma de comunicação uma base sólida para o desenvolvimento de interfaces computacionais baseadas em movimentos humanos.

2.2.3 Tradução de Gestos em Dados Digitais

A tradução dos gestos humanos em dados computacionais é realizada por meio da biblioteca MediaPipe, desenvolvida pelo Google. Essa biblioteca utiliza um pipeline de machine learning para realizar o rastreamento da região corporal desejada, permitindo identificar, analisar e interpretar gestos com precisão em tempo real.

O MediaPipe é capaz de detectar 33 pontos-chave do corpo humano com base em uma topologia inspirada no conjunto COCO (Common Objects in Context), frequentemente utilizado em tarefas de reconhecimento corporal. Essa técnica é conhecida como BlazePose, uma solução desenvolvida para oferecer alta precisão e eficiência na estimativa de poses humanas, mesmo com baixo custo computacional.

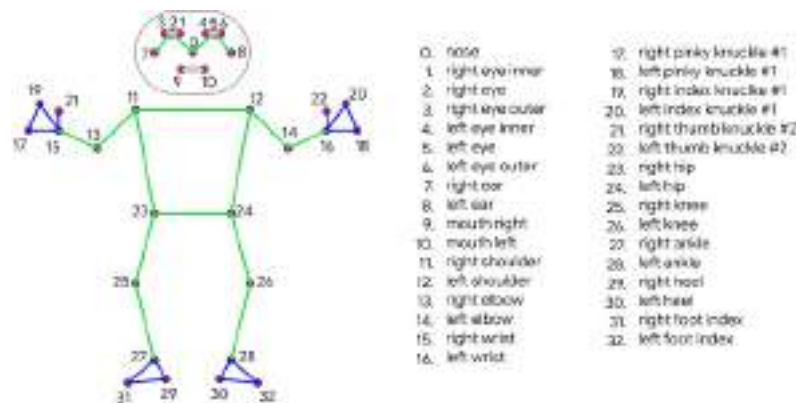


Figura 3: Topologia de 33 pontos-chave do BlazePose como superconjunto COCO (colorido em verde) Fonte: Bazarevsky e Grishchenko (2020).

A estrutura de 33 landmarks (pontos de referência) cobre regiões fundamentais do corpo, incluindo a cabeça, ombros, braços, quadris, joelhos e pés, permitindo uma compreensão completa da postura e dos movimentos do usuário. Pipeline de aprendizado de máquina em duas etapas sendo a detecção de pose: identifica rapidamente a presença do corpo humano na imagem. Rastreamento refinado: aplica um modelo mais preciso sobre a região detectada, analisando as posições e ângulos dos pontos de referência.

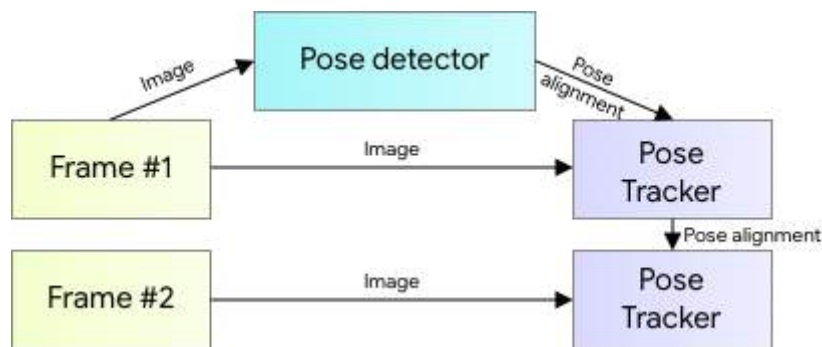


Figura 4: Visão geral do pipeline de estimativa de pose humana. Fonte: Bazarevsky e Grishchenko (2020).

2.2.4. MediaPipe Hands: Detecção de Gestos Manuais

Para a detecção de gestos manuais, o MediaPipe Hands utiliza uma abordagem baseada em machine learning, capaz de capturar 21 pontos-chave (landmarks) tridimensionais (3D) de uma mão, a partir de apenas um único frame de vídeo.

Esses pontos representam articulações e extremidades dos dedos, permitindo a modelagem precisa da posição e curvatura da mão em tempo real. Cada ponto é definido por uma coordenada espacial (x, y, z), tornando possível rastrear movimentos com alta precisão.



Figura 5: Pontos-chaves na mão pelo MediaPipe Hands. Fonte: Medium (2021).

O processo começa com o modelo BlazePalm, uma rede neural especializada na detecção de palmas. Este modelo localiza rapidamente a região da mão na imagem, fornecendo um recorte eficiente para o mapeamento detalhado dos 21 pontos.

Após a detecção da palma, o sistema aplica um modelo de regressão 3D de landmarks, responsável por prever as posições exatas das articulações e extremidades dos dedos.

Com esse mapeamento, o sistema consegue:

- Reconhecer gestos específicos com base na posição relativa entre os dedos.

- Traduzir gestos em comandos, como apontar, clicar, desenhar, apagar, entre outros.
- Determinar ações específicas por meio da análise de coordenadas e distâncias entre os pontos.

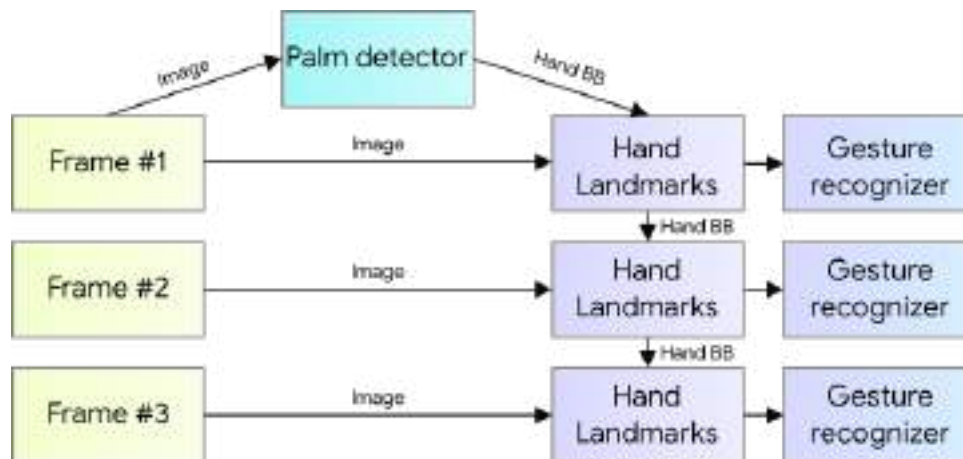
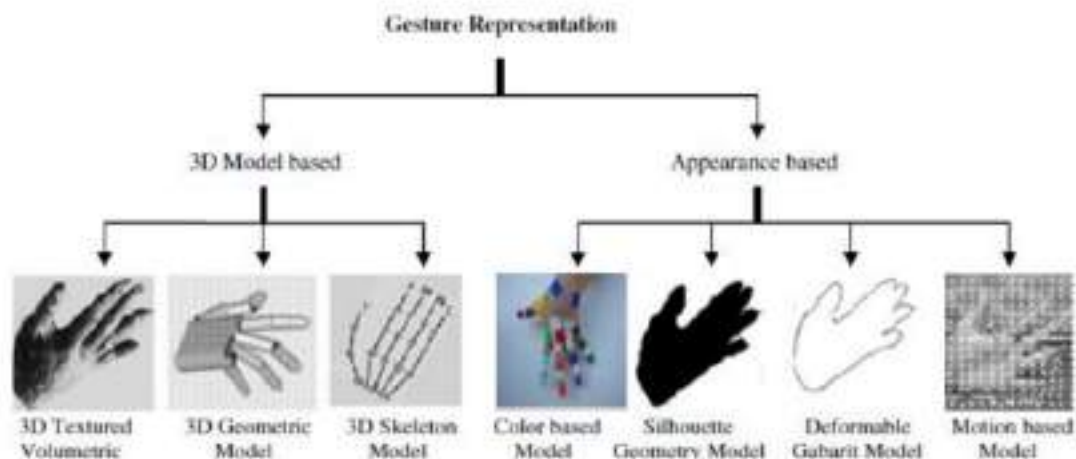


Figura 6: Visão geral do pipeline de percepção manual. Fonte: Bazarevsky e Zhang (2019).

Essa estrutura permite que a aplicação interprete a mão do usuário como uma interface interativa, transformando gestos simples em ações funcionais dentro do sistema computacional.

2.3 Técnicas de Modelagem para Reconhecimento de Mãos

As tecnologias de visão computacional aplicadas ao reconhecimento de gestos manuais se baseiam, em sua maioria, em duas abordagens principais: a modelagem baseada em aparência e a modelagem baseada em modelos tridimensionais (3D). Ambas têm sido amplamente estudadas e aplicadas em sistemas interativos, com vantagens e limitações próprias.



2.3.1 Modelagem baseada em aparência

Segundo Pereira (2022), a modelagem baseada em aparência utiliza atributos visuais extraídos diretamente de imagens 2D como cor, silhueta, posição e movimento para criar uma representação da mão e inferir os gestos. Essa abordagem é considerada mais simples, pois trabalha com propriedades bidimensionais das imagens capturadas pela câmera.

Entre suas principais vantagens estão:

- Facilidade de implementação.
- Baixo custo computacional.
- Boa performance em aplicações em tempo real.

No entanto, essa técnica é mais suscetível a ruídos visuais, como mudanças de iluminação, fundos complexos e oclusões parciais da mão, o que pode comprometer a acurácia em ambientes não controlados.

2.3.2 Modelagem baseada em modelos 3D

De forma distinta, a modelagem baseada em modelos 3D utiliza uma estrutura visual tridimensional para representar a mão de forma mais detalhada. Os parâmetros de movimento, como a posição da palma e os ângulos das articulações, são estimados a partir da comparação entre as imagens de entrada e projeções 2D do modelo 3D da mão.

Essa abordagem proporciona maior precisão e é capaz de representar uma variedade mais ampla de gestos. No entanto, tende a ser mais complexa e custosa em termos de processamento.

Os modelos tridimensionais podem ser classificados em dois tipos principais:

- Modelos volumétricos: representam a aparência completa da mão em três dimensões. São detalhados, mas altamente complexos, exigindo processamento elevado devido à grande quantidade de parâmetros envolvidos.
- Modelos esqueléticos: representam apenas a estrutura interna da mão, como um esqueleto digital. Essa simplificação reduz significativamente a complexidade, tornando o método mais viável para aplicações em tempo real.

3. Metodologia

O projeto será desenvolvido com base no cronograma de entregas definido metodologia da disciplina, representado na figura 8, o qual estrutura as etapas de execução. Além disso, será guiado pela tabela da figura 1: Plano de trabalho inicial, que apresenta um detalhamento inicial das fases do projeto. Ao término dos estudos, será realizada a integração

entre o cronograma e os planos anteriores, resultando na chamada “Metodologia Final”, que representa uma adaptação e consolidação das metodologias previamente propostas.

Metodologia da disciplina:

Data	Entrega	Descrição
07/05	Proposta do Seminário	Escopo, objetivos, bibliografia inicial e plano de trabalho
21/05	Documento Parcial	Introdução e Fundamentação teórica completas, Metodologia parcial
11/06	Protótipo prático	Metodologia completa, Demonstração inicial da implementação prática
24/06	Apresentação prévia	Envio dos slides e relatório final
—/—	Entrega Final	Envio da aplicação prática e apresentação

Figura 8: Plano da disciplina. Fonte: Marcelino (2025) .

Após a análise de planos de projetos anteriores e da metodologia da disciplina, foi realizado um estudo que resultou na divisão da metodologia deste projeto em seis etapas principais:

1-Proposta do projeto: Consiste na apresentação da ideia inicial e no detalhamento dos objetivos, incluindo uma pesquisa prévia sobre o contexto do problema e uma análise introdutória da bibliografia.

2-Fundamentação teórica: Abrange a apresentação parcial da metodologia, a configuração do ambiente de desenvolvimento e a implementação inicial da detecção da mão por meio de visão computacional

3-Metodologia completa e demonstração inicial: Detalha a metodologia final do projeto, acompanhada de uma primeira demonstração prática, com a tradução dos gestos da mão em comandos interpretados pelo sistema.

4-Interpretação de figuras geométricas e aprimoramento da detecção: Envolve a implementação da interpretação de formas geométricas e o aperfeiçoamento da detecção dos gestos, possibilitando a sobreposição de páginas e a integração com a tela e o Hand Mouse.

5-Apresentação prévia: Consiste na elaboração de uma apresentação intermediária utilizando recursos como o PowerPoint para exibição dos resultados parciais e validação do progresso do projeto.

6-Documentação final: Envolve a revisão, consolidação e finalização do projeto, com a produção da documentação completa contendo todas as etapas, resultados e aprendizados obtidos.

Em todas as etapas serão realizados testes para validação e aprimoramento contínuo do projeto, garantindo sua evolução e eficácia conforme os objetivos propostos.

O projeto consiste em uma pesquisa qualitativa aplicada, cujo objetivo é desenvolver uma solução tecnológica baseada em técnicas de visão computacional para a simulação do uso do mouse por meio do reconhecimento de gestos captados por uma webcam. Além de, utilizar conceito de IHC, que de acordo com Fávero (2024) “A preocupação principal da Interação Ser Humano-Computador (IHC) é garantir uma usabilidade flexível, produtos fáceis de usar e que proveem uma experiência agradável para usuários de sistemas digitais, isso é feito por meio de ferramentas que tornam o uso do computador mais adaptável ao usuário”

A abordagem qualitativa permite compreender, analisar e adaptar a tecnologia com base na interação do usuário e na usabilidade prática do sistema. Por se tratar de uma pesquisa aplicada, busca-se não apenas a produção de conhecimento, mas também sua utilização prática, especialmente em contextos educacionais que demandam novas formas de acessibilidade e interação digital.

4. Aplicação

4.1.1. Detecção da mão

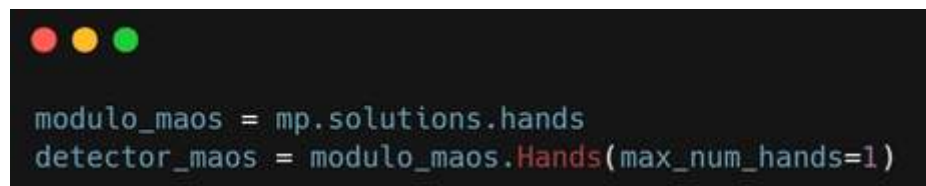
A detecção da mão é realizada por meio de uma webcam, utilizando a biblioteca MediaPipe. Essa ferramenta marca 21 pontos (landmarks) na mão, permitindo a leitura e o reconhecimento de padrões com base nessas posições. A função de detecção pode ser visualizada na Figura 1.

A biblioteca MediaPipe identifica 21 pontos principais em cada mão, o que possibilita rastrear com precisão a posição e os gestos. Esses pontos estão distribuídos da seguinte forma, conforme apresentado na Figura 2:

- **Ponto 0:** centro do pulso
- **Dedão (polegar):** pontos 1 a 4
- **Dedo indicador:** pontos 5 a 8
- **Dedo médio:** pontos 9 a 12
- **Dedo anelar:** pontos 13 a 16
- **Dedo mínimo:** pontos 17 a 20

Cada dedo possui 4 pontos, representando as juntas e a ponta do dedo. Com isso, é possível rastrear a posição, a curvatura e os gestos da mão em tempo real.

Para garantir uma melhor resposta da aplicação, a detecção é limitada a, no máximo, uma mão por vez.

A screenshot of a code editor with a dark background and three colored window control buttons (red, yellow, green) in the top-left corner. The code is written in Python and uses the MediaPipe library for hand detection. The code consists of two lines: the first line imports the 'hands' module from 'mp.solutions', and the second line creates a 'Hands' object named 'detector_maos' with the parameter 'max_num_hands=1'.

```
modulo_maos = mp.solutions.hands
detector_maos = modulo_maos.Hands(max_num_hands=1)
```

Figura 9: A figura representa código que importa o módulo media pipe.solutions.hands e cria um objeto detector de mãos com o parâmetro max_num hands=1.Fonte: Os Autores(2025).



Figura 10: Marca os 21 pontos da mão. Os Autores(2025).

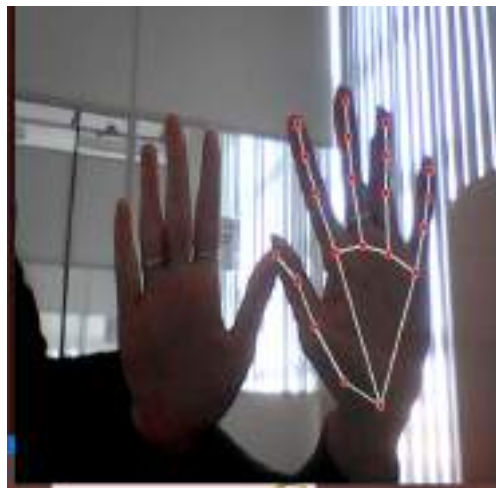


Figura 11: Representação do `max_num_hands=1`. Os Autores(2025).

Para implementar a interação com a tela, o sistema utiliza a posição do ponto 8, correspondente à ponta do dedo indicador, como indicado na figura 4 e no código na figura 5:

```

x1, y1 = int(marcadores[8].x * self.width()), int(marcadores[8].y * self.height())
self.posicao_dedo_indicador = (x1, y1)
```

Figura 12: Apresenta o trecho de código onde é indicado o ponto de posição 8, que é o indicador. Os Autores(2025).

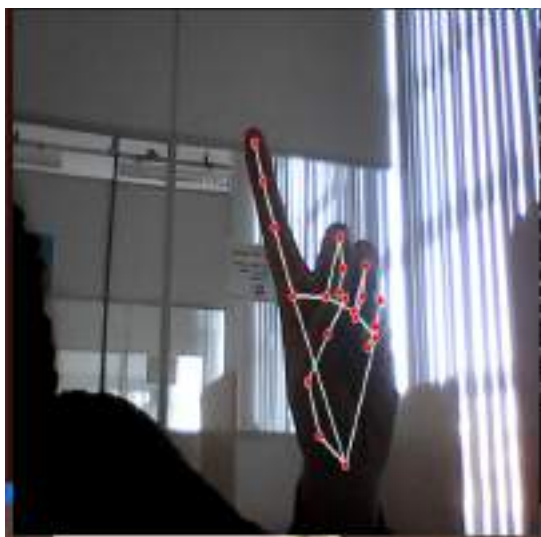


Figura 13: Esse movimento pode controlar a posição do pincel virtual, com a ponta do indicador. Os Autores(2025).

4.2.1. Desenho na tela

O desenho na tela é feito capturando a posição da ponta do dedo indicador (ponto 8) e usando essas coordenadas como a posição do "pincel" no quadro. Sempre que o dedo indicador estiver "levantado" (ou seja, pronto para desenhar), o programa traça linhas conectando os pontos anteriores com os atuais, simulando um traço contínuo.

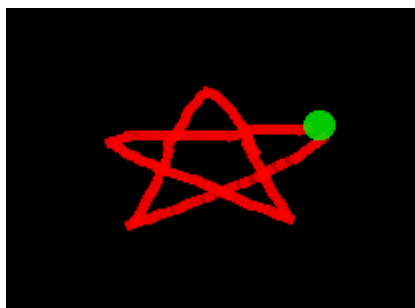


Figura 14: Utilizando o indicador para realizar a linha vermelha. Os Autores(2025).

O desenho na tela é realizado conectando a posição anterior da ponta do dedo indicador à posição atual, sempre que o dedo estiver levantado. Para isso, o código utiliza um "canvas" uma imagem preta do mesmo tamanho do vídeo capturado onde essas linhas são desenhadas. A cada novo frame da câmera, se o dedo estiver levantado, o programa traça uma linha entre a última posição conhecida do dedo e a posição atual, criando o efeito de um traço contínuo, como se fosse um pincel se movendo na tela.

4.2.2. Lousa Transparente

No sistema de desenho por gestos desenvolvido com MediaPipe e OpenCV, a chamada "lousa transparente" é uma estrutura fundamental que permite ao usuário desenhar no ar, como se estivesse interagindo com uma superfície invisível. Essa lousa é implementada

por meio de uma matriz de imagem, chamada canvas, com as mesmas dimensões do vídeo capturado pela webcam. No entanto, diferentemente da imagem da câmera, o canvas não é atualizado a cada quadro, o que permite que os traços desenhados permaneçam visíveis até que uma ação explícita de limpeza seja realizada.

A interação com a lousa ocorre da seguinte forma: a imagem da câmera é capturada continuamente em tempo real, e paralelamente é mantido um canvas preto onde os desenhos são registrados. Quando o sistema detecta que o usuário levantou apenas o dedo indicador — por meio da análise de poses com MediaPipe — ele entra no modo de desenho. Nesse modo, os movimentos do dedo são convertidos em coordenadas, e linhas são desenhadas sobre o canvas conectando os pontos sucessivos.



Figura 15: Tela da aplicação.Os Autores(2025).

4.2.3. Formas Geométricas

Além da funcionalidade de desenho livre com o dedo indicador, o sistema de lousa transparente também permite a criação de formas geométricas, como retângulos, círculos e linhas, por meio do mesmo gesto de desenho na tela, com a biblioteca MediaPipe. Essa funcionalidade amplia as possibilidades de interação, tornando o sistema mais expressivo e útil para atividades como ensino, anotações e ilustrações técnicas.

O processo de desenho é iniciado na função `atualizar_quadro()`, responsável por capturar continuamente os quadros da webcam, inverter a imagem horizontalmente (espelhamento) e convertê-la para o formato RGB, compatível com o modelo de detecção do MediaPipe. Utiliza-se o módulo Hands, que identifica 21 pontos de referência (landmarks) na mão do

usuário. Dentre eles, o ponto 8 correspondente à ponta do dedo indicador é fundamental para controlar o traçado.

Com base na posição vertical do dedo indicador (landmark 8) em relação à sua base (landmark 6), o sistema identifica se o dedo está estendido. Quando apenas o indicador está levantado, o modo de desenho é ativado. Se mais dedos estiverem estendidos, o sistema pode alternar para outros modos, como o modo borracha, que é ativado a partir de dois dedos juntos e assim o indicador que tem ativado no desdo fica em barnco e quando sai do modo retorna para a cor que o usuário selecionou.

A forma como o desenho é traçado depende da opção selecionada pelo usuário:

- Desenho livre: o sistema suaviza o movimento da "caneta" digital por meio da média ponderada entre a posição atual e a anterior do dedo. O traçado é realizado no canvas utilizando a ferramenta QPainter, com a função drawLine(), que conecta os pontos suavizados.
- Formas geométricas: ao selecionar uma forma como círculo, retângulo,quadrado ou triângulo, o sistema armazena o ponto inicial (quando o gesto começa) e o ponto final (quando o gesto termina). Ao final do gesto, a função _desenhar_forma_final() é chamada para renderizar a figura no canvas, também utilizando o QPainter. Por exemplo, no caso do círculo, o raio é calculado com base na distância euclidiana entre os dois pontos, e a figura é desenhada com drawEllipse().

A visualização final do desenho é controlada pelo método paintEvent(), que é acionado sempre que o widget precisa ser redesenhado. Esse método exibe o vídeo da webcam (caso esteja ativado), o conteúdo atual do canvas com os desenhos realizados, uma prévia tracejada da forma sendo desenhada (caso o gesto ainda esteja em andamento).

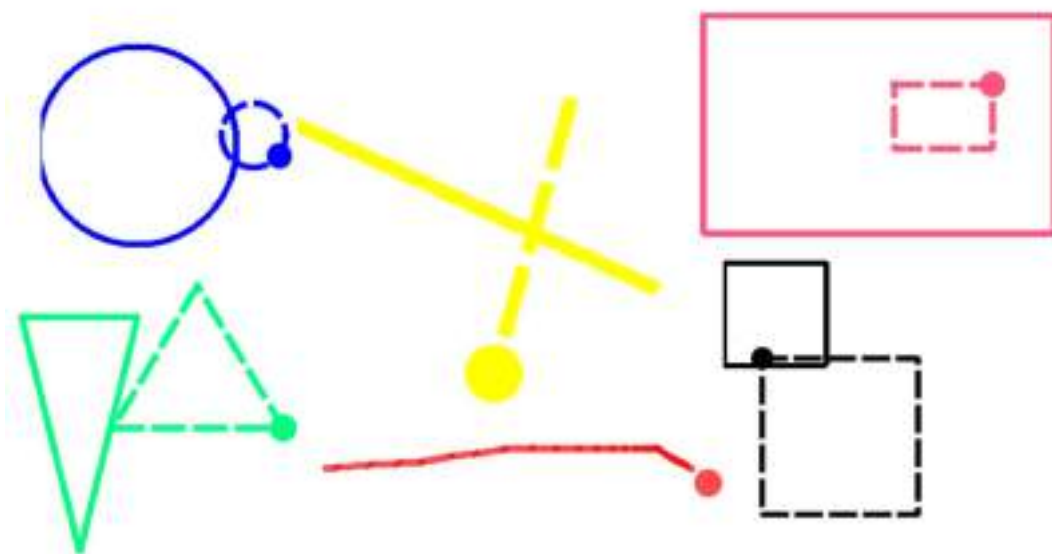


Figura 16: Demonstração das figuras geométricas e desenho livre no projeto. Os Autores(2025).

4.2.4. Menu

A interface da lousa transparente foi projetada para ser minimalista, funcional e intuitiva. No canto superior da janela, há um menu, que permite ao usuário alternar rapidamente entre modos de interação, escolher ferramentas de desenho e executar ações como limpeza da tela ou encerramento do sistema.

Esses atalhos são representados por ícones desenhados diretamente sobre o frame de vídeo, utilizando as funções gráficas do OpenCV. Cada botão é definido como uma região da tela, associada a uma ação específica. Para ativar os botões usa o próprio mouse do computador, assim clicando e ativando a função.

Os botões são:

- **Botão de selecionar cor:** permite alternar entre cores predefinidas para o traço.
- **Botão de forma:** permite alternar entre modo de desenho livre, linha, retângulo, triângulo e quadrado.
- **Botão de limpar:** limpa toda a lousa, removendo todos os desenhos persistentes.
- **Botão de sair:** encerra a aplicação de forma segura.
- **Botão de salvar:** permite ao usuário salvar a imagem que gerou em png em um fundo branco.
- **Botão alterar webcam:** permite ao usuário trocar de webcam durante o uso da aplicação.
- **Botão sair:** encerra a aplicação.

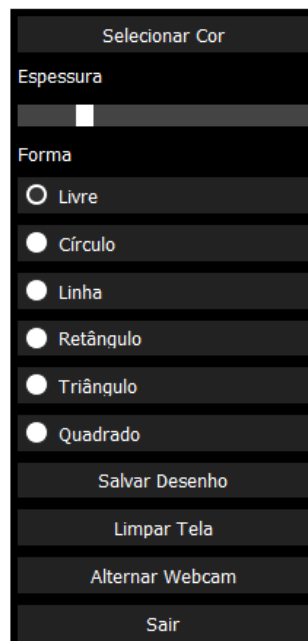


Figura 17: Menu de atalhos. Os Autores(2025).

Ao clicar no botão “Selecionar Cor”, uma janela interativa é exibida na tela, permitindo que a pessoa usuária visualize e escolha uma cor personalizada para utilizar na lousa digital. Essa janela apresenta uma paleta com diferentes tonalidades e, caso necessário, também permite ajustar a transparência da cor selecionada. Após escolher a tonalidade desejada, basta confirmar a seleção clicando em “OK”. A partir desse momento, os traços realizados na lousa passarão a ser desenhados com a nova cor definida.



Figura 18: Painel de cores.Fonte: Os autores(2025).

5. Resultado e conclusões

A aplicação Muppets foi desenvolvida com o objetivo de proporcionar uma interação natural e lúdica entre o usuário e o sistema computacional, por meio do reconhecimento de

gestos realizados com as mãos. Os testes realizados demonstraram a viabilidade da proposta, tanto do ponto de vista funcional quanto didático, validando o uso da biblioteca MediaPipe e das ferramentas OpenCV para esse tipo de interface.

5.1 Funcionalidade Geral

O sistema foi testado em ambiente local utilizando uma webcam padrão (720p), e demonstrou bom desempenho em tempo real. Os gestos definidos foram corretamente detectados em diferentes condições de iluminação e distância moderada da câmera.

As principais funcionalidades validadas foram:

- **Modo pincel:** controle fluido da posição do “mouse” com o dedo indicador (landmark 8), permitindo desenhar livremente na tela.
- **Menu interativo:** possibilita ao usuário salvar a tela, limpar a tela, alternar a webcam e sair da aplicação.
- **Formas fixas:** seleção de formas como triângulo, quadrado e círculo, selecionado no menu.
- **Modo borracha:** possibilita apagar ações anteriores, acionado com dois dedos juntos.

6. Considerações finais

O desenvolvimento da aplicação Muppets: Motion Puppets permitiu explorar, de forma prática e didática, conceitos fundamentais de visão computacional, aprendizado de máquina e interação humano-computador (IHC). A partir do uso de bibliotecas como MediaPipe, OpenCV, foi possível criar uma interface interativa baseada em gestos da mão, que responde em tempo real a comandos simples e intuitivos.

O sistema desenvolvido se mostrou funcional e com baixo custo computacional. Ele oferece um meio alternativo de controle e expressão, o que o torna especialmente relevante em contextos educacionais. A tradução de gestos em ações digitais demonstrou-se eficaz na proposta de criar uma experiência de uso fluida e engajadora.

Apesar dos bons resultados, foram identificadas algumas limitações, como a sensibilidade à iluminação e à posição da câmera, além da dificuldade de interpretação de gestos ambíguos. Essas questões, no entanto, não comprometem a proposta principal, e podem ser solucionadas com ajustes e evoluções futuras.

Conclui-se que o projeto atendeu aos objetivos propostos e contribuiu significativamente para o aprendizado prático dos temas envolvidos, além de demonstrar o potencial das interfaces baseadas em visão computacional como ferramentas interativas e educativas.

7. Referências

- ANACONDA**, Inc. *Anaconda (Python distribution)*. Disponível em: [https://en.wikipedia.org/wiki/Anaconda_\(Python_distribution\)](https://en.wikipedia.org/wiki/Anaconda_(Python_distribution)). Acesso em: 21 maio 2025.
- BAZAREWICZ, A. et al.** *On-device, real-time body pose tracking with MediaPipe BlazePose*. Google Research Blog, 2020. Disponível em: <https://research.google/blog/on-device-real-time-body-pose-tracking-with-mediapipe-blazepose/>. Acesso em: 21 jun. 2025.
- CORDEIRO, Karolina Maria de Araújo.** *O impacto da pandemia na educação: a utilização da tecnologia como ferramenta de ensino*. 2020. Trabalho de Conclusão de Curso (Graduação em Pedagogia) – Universidade Federal do Amazonas, Manaus, 2020. Disponível em: https://d1wqtxts1xzle7.cloudfront.net/76871647/O_IMPACTO_DA_PANDEMIA_NA_EDUCACAO_A_UTILIZACAO_DA_TECNOLOGIA_COMO_FERRAMENTA_DE_ENSINO-1ibre.pdf. Acesso em: 21 maio 2025.
- FAVERO, Matheus.** *Personalização de gestos para interação com páginas web*. 2019. Trabalho de Conclusão de Curso (Graduação em Engenharia de Computação) – Universidade Tecnológica Federal do Paraná, Curitiba, 2019. Disponível em: <https://repositorio.utfpr.edu.br/jspui/bitstream/1/34394/1/personalizacaogestosinteracaoweb.pdf>. Acesso em: 21 maio 2025.
- GODOY, G. A. S.** *Interface interativa para reconhecimento de gestos de avaliação em tempo real*. Medium, 2020. Disponível em: <https://medium.com/senior/interface-interativa-para-reconhecimento-de-gestos-de-avalia%C3%A7%C3%A3o-em-tempo-real-d4fab8185dc>. Acesso em: 21 jun. 2025.
- GOOGLE AI.** *Guia de soluções do MediaPipe*. Disponível em: <https://ai.google.dev/edge/mediapipe/solutions/guide?hl=pt-br>. Acesso em: 21 maio 2025.
- KENHUB.** *Anatomia da mão*. Disponível em: <https://www.kenhub.com/pt/library/anatomia/anatomia-da-mao>. Acesso em: 21 jun. 2025.
- LU, F. et al.** *On-device, real-time hand tracking with MediaPipe*. Google Research Blog, 2019. Disponível em: <https://research.google/blog/on-device-real-time-hand-tracking-with-mediapipe/>. Acesso em: 21 jun. 2025.
- MEDIUM.** *Real-time hand tracking and gesture recognition with MediaPipe: Rerun Showcase*. Disponível em: <https://medium.com/data-science/real-time-hand-tracking-and-gesture-recognition-with-mediapipe-rerun-showcase-9ec57cb0c831>. Acesso em: 21 maio 2025.

PYTHON SOFTWARE FOUNDATION. *Python 3.13.0 Release*. Disponível em: <https://www.python.org/downloads/release/python-3130/>. Acesso em: 21 maio 2025.

RIBEIRO, M. A. B. et al. Avaliação da destreza manual em idosos através da Escala de Destreza de Minnesota. *Revista da Sociedade Brasileira de Fonoaudiologia*, São Paulo, v. 22, n. 3, p. 217–222, jul./set. 2017. Disponível em: <https://www.scielo.br/j/rsbf/a/qdrXnTJ7m6YzfwWggM5Y67F/>. Acesso em: 21 jun. 2025.

SOBOTTA, J. *Figura 2.51: Vista Posterior da Mão Direita Humana – Ossos e Articulações*. Disponível em: https://www.researchgate.net/figure/Figura-251-Vista-Posterior-da-Mao-Direita-Humana-Ossos-e-Articulacoes_fig1_316261909. Acesso em: 21 jun. 2025.

WIKIPÉDIA. *NumPy*. Disponível em: <https://pt.wikipedia.org/wiki/NumPy>. Acesso em: 21 maio 2025.

WIKIPÉDIA. *OpenCV*. Disponível em: <https://pt.wikipedia.org/wiki/OpenCV>. Acesso em: 21 maio 2025.