

CENTRO UNIVERSITÁRIO DE VOTUPORANGA - UNIFEV

ENGENHARIA DE COMPUTAÇÃO

FERNANDO APARECIDO MAFFEI PIRES

MURILLO BARROS BADIO

BIOLIFTVISION – ANÁLISE DE MOVIMENTOS DE ATLETAS

VOTUPORANGA

2025

FERNANDO APARECIDO MAFFEI PIRES

MURILLO BARROS BADIO

BIOLIFTVISION - ANALISE DE MOVIMENTOS DE ATLETAS

Monografia apresentado à Unifev –
Centro Universitário de Votuporanga –
para a obtenção do grau de Bacharel em
Engenharia de Computação, sob a
orientação do Prof. Dr. Fernando Kendy
Aoki Rizzato.

VOTUPORANGA

2025

Maffei Pires, Fernando Aparecido.

BIOLIFTVISION - ANALISE DE MOVIMENTOS DE ATLETAS. / Fernando Aparecido
Maffei Pires; Murillo Barros Badio. - Votuporanga. Ed. do Autor, 2025.

69 p., 30cm. :il., tab.

Trabalho de Conclusão de Curso (Graduação - Bacharelado) - UNIFEV - Centro
Universitário de Votuporanga, Curso de Engenharia da Computação, 2025.

Orientador: Prof. Me. Fernando Kendy Aoki Rizzato.

1.Powerlifting. 2. Machine learning. 3. Biomecânica. 4. Visão computacional. I. Barros
Badio, Murillo. II. Título.

Sistema de geração automática de fichas catalográficas da Unifev.

Dados fornecidos pelo autor(a).

Essa ficha não pode ser modificada.

Bibliotecária Responsável: Marcia Faria Cavalcante - CRB-8/ 10706

CENTRO UNIVERSITÁRIO DE VOTUPORANGA
ENGENHARIA DE COMPUTAÇÃO

FERNANDO APARECIDO MAFFEI PIRES
MURILLO BARROS BADIO

BIOLIFTVISION – ANÁLISE DE MOVIMENTOS DE ATLETAS

Monografia apresentado à Unifev – Centro
Universitário de Votuporanga – para a obtenção
do grau de Bacharel em Engenharia de
Computação, sob a orientação do Prof. Dr.
Fernando Kendy Aoki Rizzato.

Aprovado(a): ____/____/____

Prof. Dr. Fernando Kendy Aoki Rizzato

Orientador

Fernando Bermejo Menechelli

Primeiro examinador

Prof. Dra Tainá Fernanda Garbelim Pascoalato

Segundo Examinador

VOTUPORANGA

2025

RESUMO

Este trabalho aborda a análise biomecânica de movimentos no powerlifting, área marcada por alto risco de lesões devido a execuções técnicas inadequadas, propondo uma solução computacional para suporte ao treinamento de atletas. O objetivo principal consistiu em desenvolver um software que utilize visão computacional e aprendizado de máquina para avaliar a qualidade técnica de exercícios, fornecendo feedback automático e histórico de análises. A justificativa reside na demanda por ferramentas acessíveis e precisas que auxiliem treinadores e atletas a aprimorar a performance e prevenir riscos, superando limitações de métodos manuais ou equipamentos caros. A metodologia adotou abordagem quantitativa experimental, com captura e processamento de vídeos via MediaPipe para extração de keypoints corporais em 3D e cálculo de ângulos articulares relevantes (joelhos, quadris, ombros). Modelos de machine learning one-class foram treinados para classificar movimentos válidos e inválidos, integrados a uma interface gráfica em PyQt5 que permite seleção de vídeos e consulta a históricos. Testes ocorreram com dataset de powerlifting, avaliando precisão em cenários reais. Os resultados revelaram precisão superior a 85% na classificação de movimentos, com processamento eficiente e feedback em tempo real, validando a robustez do sistema em diferentes condições de iluminação e ângulos. Conclui-se que a solução é viável como ferramenta de apoio ao treinamento biomecânico, contribuindo para a inclusão tecnológica no esporte e abrindo caminhos para expansões como integração com wearables.

Palavras-chave: powerlifting; análise biomecânica; visão computacional; MediaPipe; machine learning.

ABSTRACT

This work addresses the biomechanical analysis of movements in powerlifting, a field marked by a high risk of injuries due to inadequate technical execution, by proposing a computational solution to support athlete training. The main objective was to develop software that uses computer vision and machine learning to assess the technical quality of exercises, providing automatic feedback and a history of analyses. The justification lies in the demand for accessible and accurate tools that help coaches and athletes improve performance and prevent risks, overcoming the limitations of manual methods or expensive equipment. The methodology adopted an experimental quantitative approach, with video capture and processing using MediaPipe for extracting 3D body keypoints and calculating relevant joint angles (knees, hips, shoulders). One-class machine learning models were trained to classify valid and invalid movements, integrated into a PyQt5 graphical interface that allows video selection and access to analysis records. Tests were conducted using a powerlifting dataset, evaluating accuracy in real-world scenarios. The results showed accuracy above 85% in movement classification, with efficient processing and real-time feedback, validating the system's robustness under different lighting conditions and angles. It is concluded that the solution is viable as a support tool for biomechanical training, contributing to technological inclusion in sports and opening paths for expansions such as integration with wearables.

Keywords: powerlifting; biomechanical analysis; computer vision; MediaPipe; machine learning.

LISTA DE FIGURAS

Figura 1 – Movimento Correto para Realizar o Agachamento.....	18
Figura 2 – Movimento Correto para Realizar Levantamento Terra.....	19
Figura 3 – Movimento Correto para Realizar Supino.....	20
Figura 4 – Diagrama de Casos de Uso.....	33
Figura 5 – Diagrama de Classes.....	33
Figura 6 – Diagrama de Sequência.....	34
Figura 7 – Diagrama de Componentes.....	35
Figura 8 – Estrutura Fluxograma do Processo.....	36
Figura 9 – Interface Inicial.....	56
Figura 10 – Interface de Login.....	57
Figura 11 – Interface para Cadastro de Usuários.....	58
Figura 12 – Interface de Ferramentas e Funcionalidades.....	59
Figura 13 – Interface do Software em Execução 1.....	60
Figura 14 – Interface do Software em Execução 2.....	60
Figura 15 – Interface de Avaliação/Julgamento.....	61
Figura 16 – Interface de Histórico.....	62
Figura 17 – Interface de Seleção de Vídeo.....	63

LISTA DE QUADROS

Quadro 1 – Critérios Usados pelo Jurado para Agachamento.....	16
Quadro 2 – Critérios Usados pelo Jurado para Supino.....	16
Quadro 3 – Critérios Usados pelo Jurado para Levantamento Terra.....	17
Quadro 4 – Quadro de Ângulos para o Movimento.....	23
Quadro 5 - Função de extração de features.....	50
Quadro 6 - Pipeline de classificação por modelo One-Class.....	56

SUMÁRIO

1. INTRODUÇÃO.....	9
1.1 Objetivo geral	10
1.2 Objetivos Específicos	10
1.3 Justificativa.....	11
1.4 Metodologia.....	12
2. REFERENCIAL TEÓRICO	12
2.1 Powerlifting e Biomecânica.....	13
2.1.1 Regras de Validação em Competições (IPF, CBPL)	15
2.1.2 Critérios Técnicos dos movimentos	18
2.1.3 Análise Biomecânica de Movimentos	20
2.2 Visão Computacional Aplicada ao Esporte	23
2.2.1 Detecção de Keypoints.....	24
2.2.2 Extração de Ângulos Articulares.....	25
2.2.3 Rastreamento de Movimentos em Tempo Real.....	25
2.3 Machine Learning para Classificação de Movimentos.....	26
2.3.1 Algoritmos (Random Forest, SVM, XGBoost).....	26
2.3.2 Métricas de Avaliação (Acurácia, Precision, Recall, F1-Score).....	26
2.3.3 Validação Cruzada e Overfitting	27
2.4 Trabalhos Relacionados	27
2.4.1 Sistemas de Análise Automática em Esportes.....	27
2.4.2 Limitações de Soluções Existentes	28
3. METODOLOGIA DE DESENVOLVIMENTO	29
3.1 Requisitos do Sistema.....	29
3.1.1 Requisitos Funcionais	29
3.1.2 Requisitos Não Funcionais.....	30
3.2 Arquitetura e Fluxo do Sistema	30
3.2.1 Fluxo Geral:.....	31
3.2.2 Tecnologia Base:	32

3.3 Diagramas UML e Arquiteturas.....	32
3.3.1 Diagrama de Casos de Uso	33
3.3.2 Diagrama de Classes	33
3.3.3 Diagrama de sequência.....	34
3.3.4 Diagrama de Componentes	35
3.3.5 Fluxograma do Processo	36
3.4 Tecnologias e Ferramentas	38
3.4.1 Linguagem: Python	38
3.4.2 Bibliotecas.....	39
3.4.3 Ferramentas	47
3.5 Pré-Processamento de Dados	49
3.5.1 Features: Ângulos articulares e distâncias entre pontos.....	50
3.5.2 Tratamento de Dados.....	51
3.5.3 Balanceamento: SMOTE para classes desbalanceadas.....	52
3.5.4 Normalização: StandardScaler para padronização.....	52
3.6 Banco de Dados e Estrutura de Armazenamento.....	53
3.6.1 Tabela Users	53
3.6.2 Tabela Analises.....	54
3.6.3 Tabela Analises_Log (Auditoria)	55
3.7 Desenvolvimento do Software.....	55
4. RESULTADOS	57
4.1 interface Inicial	57
4.2 Interface de Login.....	58
4.3 Interface para Cadastro de Usuários.....	58
4.4 Interface de Ferramentas e Funcionalidades	59
4.5 Interface do Software em Execução	60
4.6 Interface de Avaliação/Julgamento	62
4.7 Interface de Histórico	63
4.8 Interface de Seleção de Vídeo	63
5. CONCLUSÃO.....	64

6. REFERÊNCIAS	65
----------------------	----

1. INTRODUÇÃO

Oliveira (2018) e Souza (2021) destacam que a evolução dos algoritmos de aprendizado de máquina, aliada ao advento do Big Data e ao avanço das capacidades computacionais, tem impulsionado novos horizontes em diversas áreas do conhecimento. A convergência dessas tecnologias tem promovido transformações significativas em setores como a medicina, por meio do auxílio na detecção precoce de doenças, e as finanças, com a automatização da análise de riscos e previsões de mercado.

No contexto esportivo, essas inovações ampliam consideravelmente as possibilidades de análise do desempenho atlético. O uso de técnicas de aprendizado de máquina tem demonstrado eficácia na previsão de resultados e na avaliação do desempenho em modalidades como o levantamento de peso, além de possibilitar a personalização de programas de treinamento (MEDINA-ROMERO et al., 2023).

O *powerlifting*, por sua vez, apresenta características específicas que o diferenciam de outras modalidades de levantamento de peso, especialmente no que se refere aos critérios técnicos exigidos para a validação dos movimentos. A avaliação desses movimentos é tradicionalmente realizada por juízes especializados, que se baseiam em sua experiência e interpretação das regras da modalidade.

Entretanto, a subjetividade inerente ao julgamento humano pode resultar em inconsistências, divergências de interpretação e decisões contraditórias. Segundo Plessner e Haar (2006), tais subjetividades podem comprometer a imparcialidade e a transparência das competições esportivas quando não há mecanismos objetivos de controle e padronização.

Diante desse cenário, este estudo propõe o desenvolvimento de um software que utilize técnicas avançadas de visão computacional e redes neurais artificiais com o objetivo de auxiliar juízes na análise dos movimentos realizados por atletas de *powerlifting*. Espera-se que a aplicação dessas tecnologias contribua para uma avaliação mais precisa, eficiente e transparente, reduzindo a ocorrência de decisões equivocadas e estabelecendo critérios objetivos que também possam auxiliar na formação e no treinamento de novos juízes.

Além disso, a utilização de sistemas baseados em visão computacional aplicados à análise de movimentos humanos tem se destacado por permitir avaliações biomecânicas precisas a partir de vídeos convencionais, sem a necessidade de sensores corporais invasivos. Essa abordagem apresenta vantagens como menor custo, maior acessibilidade e facilidade de aplicação em ambientes competitivos reais, sendo amplamente empregada em estudos voltados à análise técnica e prevenção de erros de execução em modalidades esportivas (OPENPOSE, 2019; CAO et al., 2021).

1.1 Objetivo geral

O objetivo geral deste estudo é criar um software usando inteligência artificial e visão computacional que irá ajudar na análise dos movimentos dos atletas durante eventos de *powerlifting*. Este software revisará os movimentos, entre eles, para verificar a validade, fornecendo relatórios objetivos, juntamente com decisões justas, para que as decisões dos juízes sejam mais definidas de acordo com padrões objetivos em vez de intuições subjetivas — trazendo melhor transparência e precisão em jogos de todos os tipos.

1.2 Objetivos Específicos

Para este fim, determinou-se os seguintes objetivos específicos:

- I. Criar um sistema de análise de movimento de máquinas que opere automaticamente e possa realizar movimentos em competições de força, julgando sua adequação por meio de uma análise de vídeo detalhada;
- II. Combinar várias funcionalidades em um todo unificado, como captura e análise de vídeo, modelagem de aprendizado de máquina, saída de resultados em tempo real, além de uma interface interativa para juízes.

1.3 Justificativa

Por muito tempo, juízes humanos foram responsáveis por avaliar o levantamento de peso. No entanto, preocupações inerentes a esse método surgiram ao longo do tempo: se um juiz julga subjetivamente ou interpreta os movimentos de forma diferente do que foi pretendido, os resultados podem acabar sendo irracionalmente díspares ou injustos para certos concorrentes (PLESSNER E HAAR, 2006).

Dadas as circunstâncias, o uso de inteligência artificial e visão computacional para avaliar desempenhos atléticos demonstrou ser promissor no avanço da confiabilidade e imparcialidade. Isso é particularmente evidente agora que os levantadores de peso têm tanta facilidade em interagir com a tecnologia, e o cenário está pronto para grandes mudanças em futuros desenvolvimentos (MEDINA-ROMERO *et al.*, 2023).

A tecnologia tem o potencial de mudar completamente a forma como os movimentos atléticos são tratados. Ela introduz um nível revolucionário de justiça e precisão.

Para a "tecnologia de análise de vídeo por IA", usa-se inteligência artificial e um software baseado em visão computacional para ser realizado em tempo real na mesma tela dos movimentos dos atletas. Ela tem a capacidade de captar padrões de movimento complexos que são prontamente decifráveis por pessoas (MEDINA-ROMERO *et al.*, 2023).

Esta tecnologia não só reduz erros comuns e melhora a consistência das decisões, mas também contribui para a justiça das competições. Isso garante que esses padrões objetivos ditarão os resultados (CENTRAL DO FUTEBOL, 2024).

Além do rigor técnico dos indicadores utilizados, a aplicação dessas tecnologias possui implicações significativas. Sua adoção pode contribuir para a padronização dos critérios de avaliação no halterofilismo, promovendo maior equidade entre os participantes e tornando o julgamento dos movimentos mais transparente e compreensível.

Diante da análise padronizada de técnicas de movimento entre competições diversas, atletas e seus treinadores são incentivados a seguir um caminho de excelência e

justiça. Além disso, a ferramenta servirá como apoio ao treinamento de novos juizes ao lhes oferecer um recurso valioso de aprendizado e treinamento.

Do ponto de vista da inovação tecnológica, este trabalho representa um avanço na aplicação da inteligência artificial e visão computacional no esporte. Ele propõe um sistema de análise de movimentos que será capaz de interpretar ações humanas, promovendo maior precisão e justiça em disputas esportivas. Segundo Medina-Romero *et al.* (2023), essa abordagem pode ser aplicada a diversas modalidades, ampliando as possibilidades de pesquisa e avaliação de desempenho.

Este estudo tem o potencial de contribuir para o esporte do *powerlifting* por meio da padronização do sistema de análise dos movimentos dos atletas. A implementação do software proposto visa auxiliar atletas e juizes na avaliação técnica dos movimentos, ao mesmo tempo em que amplia as possibilidades de pesquisa na área de análise de movimentos esportivos, estabelecendo bases metodológicas para o desenvolvimento de soluções futuras nesse campo.

1.4 Metodologia

A metodologia adotada neste estudo combina os princípios do CRISP-DM (Cross-Industry Standard Process for Data Mining) com práticas de desenvolvimento ágil, visando garantir a qualidade, a eficiência e a adaptabilidade do software de análise de movimentos para *powerlifting*.

2. REFERENCIAL TEÓRICO

Nesta seção, serão apresentados os conceitos e regulamentações essenciais para a compreensão das normas de validação em competições de *powerlifting*, conforme estabelecido pela International Powerlifting Federation (IPF) e adotado pela Confederação Brasileira de Levantamentos Básicos (CBPL). Em seguida, aborda-se os critérios técnicos para a execução dos três levantamentos principais, agachamento, supino e levantamento terra, os comandos utilizados pelos árbitros durante as competições e as especificações de equipamentos permitidos. O entendimento dessas regras é fundamental

para garantir a conformidade dos atletas e treinadores com os padrões internacionais e nacionais, assegurando a equidade e a segurança nas competições.

2.1 Powerlifting e Biomecânica

O *powerlifting*, também conhecido como levantamento básico, é praticado mundialmente e envolve três movimentos principais: agachamento, supino e levantamento terra, conforme descrito pela Federação Internacional de *Powerlifting* (IPF, 2024).

O levantamento de peso é uma modalidade esportiva cujo principal foco é a expressão da força, sendo praticado competitivamente por homens e mulheres a partir dos 14 anos, com divisão de atletas por categorias de idade e massa corporal, conforme as diretrizes da Federação Internacional de Levantamento de Peso (IWF, 2024).

De acordo com a Federação Brasileira de Levantamento de Peso (apud SILVA, 2021, p. 88), o esporte é organizado com base em parâmetros técnicos específicos, com o objetivo de padronizar as competições e garantir a validade dos levantamentos. Esse conjunto de normas contribui para a uniformização dos critérios técnicos, favorecendo a formação de atletas mais bem preparados e a consolidação da modalidade em âmbito competitivo.

É nesse contexto que surge a biomecânica, a ciência do movimento humano é fundamental para garantir que as performances sejam executadas com segurança e eficácia e atendam às regras oficiais do esporte. A biomecânica permite entender por que fatores a performance é influenciada, o que ajuda os atletas e suas equipes de treinamento a aperfeiçoarem as técnicas e evitarem lesões enquanto praticam o esporte.

A mecânica corporal é como o corpo se movimenta em relação a si mesmo e ao ambiente ao seu redor. Para que os levantamentos sejam realizados corretamente, lesões sejam prevenidas e o desempenho atlético melhorado, dominar a mecânica corporal é essencial. Os levantamentos necessários para mobilizar os principais grupos musculares são chamados de biomecânica. Além disso, é necessário estimular esses músculos de forma eficaz e fazê-los contrair-se rapidamente e muito mais fortemente do que antes. (HACKETT *et al.*, 2012)

Ao verificar ângulos e padrões de movimento, disfunções podem ser identificadas. Desta forma, a orientação baseada em evidências necessária para eliminar comportamentos mal adaptativos e prevenir a sobrecarga de tecidos pode ser implementada. (SWINTON *et al.*, 2012)

A posição da barra tem efeito na mecânica de realizar um agachamento. A "Barra Baixa", com a barra repousando sobre o topo dos deltoídeos posteriores, concentra mais na cadeia posterior e tende a inclinar-se para frente enquanto empurra os joelhos para fora. Isso requer uma maior mobilidade nos tornozelos e um suporte adequado para a lombar. (SWINTON *et al.*, 2012)

Diante dessas variações técnicas, torna-se relevante compreender os aspectos biomecânicos associados a cada posicionamento da barra, uma vez que pequenas alterações na execução do movimento podem influenciar significativamente a distribuição de cargas articulares, a ativação muscular e a estabilidade do atleta durante o agachamento.

Por outro lado, a "Barra Alta" força a levar os quadris mais para trás do que o usual e também a olhar mais para frente do que em muitos membros, devido à menor estabilidade geral relatada por colaboradores com joelhos travados demais. (SWINTON *et al.*, 2012)

A largura da pegada e o ângulo do cotovelo impactam a mecânica do supino. Uma pegada mais ampla mantém o movimento mais curto e enfatiza a participação do peitoral maior, além de intensificar a participação dos tríceps. Além disso, trazer os cotovelos ligeiramente mais próximos do corpo pode reduzir a tensão excessiva nos ombros, contribuindo para uma distribuição de peso mais equilibrada. (VAN DEN TILLAAR & SAETERBAKKEN, 2014)

Existem dois estilos principais de levantamento terra: convencional e sumô. Com uma posição ampla e joelhos separados, o espaço restante mais estreito para um levantamento uniforme é reduzido, enquanto se segue até a repetição na superfície – o efeito na produção real desse posicionamento é desconhecido. (ESCAMILLA *et al.*, 2000)

Características biomecânicas individuais devem ser consideradas ao escolher o estilo de levantamento terra. As anilhas proporcionam o equilíbrio para manter a força

mais uniformemente por lado, enquanto os pés em posição mais ampla permitem o contato ideal com o chão e as mãos para uma melhor pegada. (ESCAMILLA *et al.*, 2000)

O levantamento eficaz requer métodos como "rigidez corporal" e a "manobra de Valsalva". Esta última causa um aumento na pressão intra-abdominal que estabiliza suas costas e permite que sua força interna seja transferida para o mundo externo ao longo de uma linha muscular. (HACKETT *et al.*, 2013)

Manter a estabilidade e respeitar a amplitude de movimento são essenciais para segurança e eficiência tanto no treinamento quanto na competição.

2.1.1 Regras de Validação em Competições (IPF, CBPL)

As competições de *powerlifting* seguem critérios técnicos rigorosos definidos por federações como a IPF (International Powerlifting Federation) e a CBPL (Confederação Brasileira de Powerlifting), que padronizam as regras a fim de garantir imparcialidade, segurança e integridade no julgamento dos movimentos. De acordo com o livro de regras da IPF (2025), cada levantamento deve ser executado, abrangendo desde a posição inicial até os critérios de finalização. A CBPL adota essas diretrizes como referência, realizando adaptações pontuais à realidade nacional. Tais critérios refletem princípios biomecânicos importantes, com foco na preservação da integridade articular e no controle neuromuscular do atleta.

Sendo assim, tem-se os principais critérios de validação e os comandos utilizados pelos árbitros, divididos por levantamento:

2.1.1.1 Agachamento (Squat)

Comandos:

- **"Squat"**: sinaliza o início do movimento, após o atleta assumir a posição inicial e estabilizar a barra.
- **"Rack"**: autoriza o retorno da barra ao suporte após o término do movimento.

Critérios de Validação:

Quadro 1 – Critérios usados pelo jurado para agachamento

Critério	Descrição
Profundidade	O quadril deve ultrapassar visivelmente a linha superior dos joelhos.
Estabilidade	O atleta não deve apresentar oscilação excessiva ou dar passos.
Posição da barra	Permitido uso de barra alta ou baixa, desde que estável.
Finalização	Deve haver extensão completa de quadril e joelhos, em posição ereta.

Fonte: Autor (2025).

2.1.1.2 Supino (Bench Press)

Comandos:

- **"Start"**: autoriza o início da descida da barra.
- **"Press"**: sinal para iniciar a subida, após pausa no peito.
- **"Rack"**: libera o retorno da barra ao suporte.

Critérios de Validação:

Quadro 2 - Critérios usados pelo jurado para supino

Critério	Descrição
Pausa no peito	A barra deve permanecer imóvel até o comando de subida do árbitro.
Extensão dos cotovelos	Os cotovelos devem estar totalmente estendidos na finalização.
Pés e cabeça fixos	Devem permanecer em contato com o solo e banco, respectivamente.

Posição das nádegas	As nádegas não podem perder contato com o banco durante a execução.
---------------------	---------------------------------------------------------------------

Fonte: Autor (2025)

2.1.1.3 Levantamento Terra (Deadlift)

Comando:

- **"Down"**: o único comando dado, após a completa extensão do corpo, autorizando o atleta a devolver a barra ao chão.

Critérios de Validação:

Quadro 3 - Critérios usados pelo jurado para levantamento terra

Critério	Descrição
Lockout	O atleta deve estar ereto, com joelhos e quadris totalmente estendidos.
Comando de descida	A barra só pode ser abaixada após o comando do árbitro.
Pegada e controle	Pegada mista ou convencional permitidas, sem oscilação ou soltura.
Oscilação da barra	Não pode haver oscilação, queda ou apoio da barra nas coxas (ramping).

Fonte: Autor (2025)

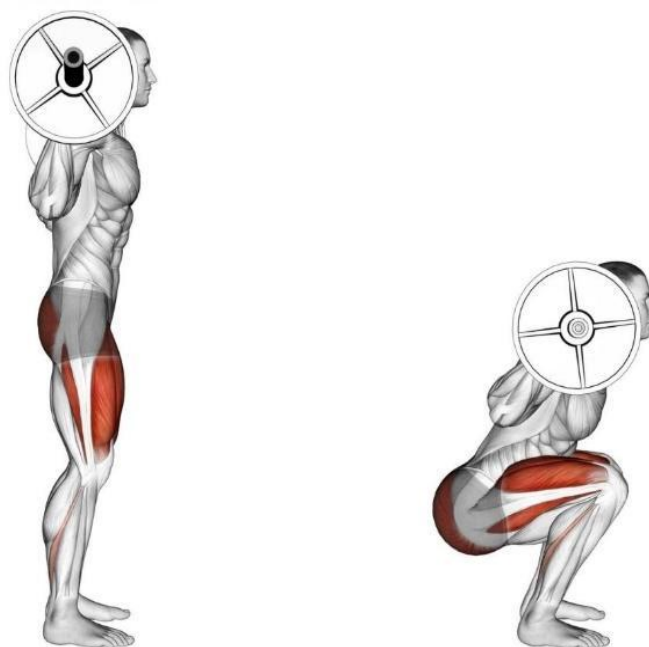
Esses critérios e comandos asseguram que o movimento seja biomecanicamente eficiente, seguro e tecnicamente válido. Erros como desequilíbrio, descida ou subida fora do tempo, execução incompleta e perda de contato com os pontos de apoio resultam na invalidação da tentativa. Assim, é essencial que atletas e treinadores dominem essas normas para garantir a performance e evitar penalizações.

2.1.2 Critérios Técnicos dos movimentos

2.1.2.1 Agachamento (Squat): Para que o agachamento seja validado em competição, é necessário descer até que a articulação do quadril esteja abaixo da parte superior dos joelhos, conforme definido pela IPF. O padrão de habilidade também exige não apenas uma amplitude suficiente do quadril e do tornozelo, mas também controle postural e força excêntrica da musculatura que governa a descida do corpo.

Segundo as regras da IPF, o agachamento envolve um movimento de cadeia cinética fechada e requer grupos musculares bastante grandes (incluindo os quadríceps, o glúteo máximo, os isquiotibiais e o eretor da espinha) para coordenar a manutenção da estabilidade da linha média e uma posição espinal estendida neutra. O desempenho eficaz irá minimizar a flexão excessiva do tronco, perda de equilíbrio e estresse nas articulações. O movimento correto para realizar o agachamento é mostrado na Figura 1.

Figura 1 - Movimento correto para realizar o agachamento



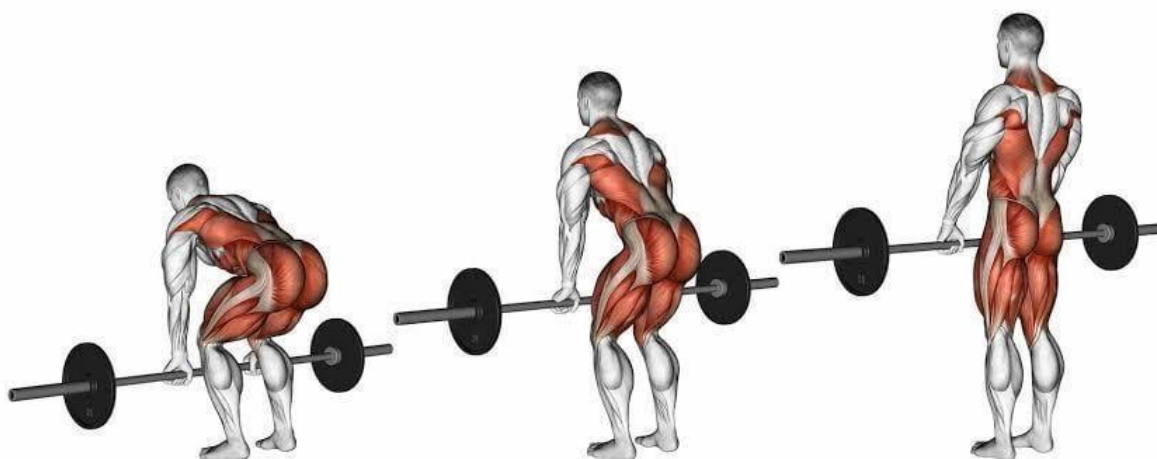
Fonte: Treinamento Mestre (2022).

2.1.2.2 Levantamento Terra (Deadlift): Conforme as regras da IPF, no levantamento, o bloqueio é a parte mais importante do levantamento para o sucesso da tentativa. O atleta deve ficar completamente ereto com os joelhos e quadris totalmente

estendidos e com os ombros atrás dos quadris. Esta é, de fato, a posição estendida que garante o envolvimento dos músculos extensores e constitui o ponto máximo de contração.

Desta forma, a IPF retrata que o levantamento terra está entre os levantamentos mais desafiadores em termos de recrutamento da cadeia posterior - o glúteo máximo, isquiotibiais, eretores da espinha, trapézio e latíssimo do dorso, em particular. A execução do levantamento requer "endireitamento" da coluna e coordenação motora de modo que, ao levantar o tronco, ações compensatórias, como apoio nas coxas (arrastamento, visto em uma tentativa inválida), sejam evitadas. O movimento correto para realizar o levantamento terra é mostrado na Figura 2.

Figura 2 - Movimento correto para realizar levantamento terra



Fonte: Treinamento Mestre (2022).

Supino (*Bench Press*): Apartir da federação IPF o movimento do atleta deve seguir os seguintes comandos: a barra é levada ao peito, sem impacto, a elevação é pausada (pausa de supino) sobre o peito, e então a barra deve ser pressionada para fora em um movimento estável e suave até chegar à extensão completa do cotovelo. Levantar os pés e/ou as nádegas e/ou as escápulas do banco não é permitido.

Biomecanicamente conforme a IPF, o supino é um movimento de cadeia fechada que envolve a geração de forças bilaterais e coordenadas da porção peitoral (esternal) do

peitoral maior, deltoides anteriores e tríceps, com ativação de vários sinergistas e estabilizadores, incluindo as porções média e posterior do peitoral maior e o ancôneo. Para que a técnica seja considerada adequada a IPF diz que é necessário manter todos os cinco pontos de contato com o banco (cabeça, escápulas, nádegas, pés plantados) e a curvatura natural da lombar, pois isso contribui para um caminho de barra mais eficiente.

A IPF retrata que não só a transferência ótima de força é alcançada quando o supino é realizado corretamente, mas o risco de lesão no ombro e cotovelo também é reduzido – uma consideração primordial se o atleta quiser permanecer saudável e ser um atleta a longo prazo. Na figura 3 é apresentado o movimento correto para realizar o supino

Figura 3 – Movimento correto para realizar supino



Fonte: Treinamento Mestre (2022).

2.1.3 Análise Biomecânica de Movimentos

A fórmula do cosseno pode ser usada para determinar os ângulos das articulações, com base em vetores que são formados por segmentos ósseos adjacentes. Dentro desse contexto, a biomecânica do movimento corporal torna-se uma análise de física estatística: não apenas a medida do deslocamento, velocidade e aceleração, mas também ângulos articulares cujas mudanças indicam alterações na posição relativa de dois segmentos do

corpo. É precisamente este último tipo de dado que nos informa exatamente que tipo de sinal de entrada causa movimentos específicos nos membros — e para diferentes movimentos. Esses ângulos desempenham uma função crucial na descrição cuidadosa dos padrões de movimento HUSTON, Ronald. Principles of Biomechanics. 1. ed. New York: CRC Press, 2008)

De acordo com a PUC-Rio (2015), durante agachamentos, levantamento terra e supino, exercícios comumente realizados em sessões de treinamento de resistência e competições, é quando diferenças lineares próximas às articulações ocorrem. Isso pode acontecer quando um atleta executa exercícios nos quais o padrão motor para executá-los não é apresentado.

Assim, utilizando a fórmula do cosseno para os vetores de segmentos ósseos adjacentes, é possível calcular o ângulo articular. De acordo com a equação 1:

$$\theta = \frac{A \rightarrow \cdot B \rightarrow}{\|A \rightarrow\| \|B \rightarrow\|} \quad (1)$$

- $A \rightarrow$ corresponde ao vetor do quadril ao joelho;
- $B \rightarrow$ corresponde ao vetor do tornozelo ao joelho;
- θ corresponde ao ângulo do joelho.

Este cálculo poderá permitir a medição do alcance do movimento articular e determinar padrões biomecânicos, como assimetria ou compensações posturais. Desta forma, a análise biomecânica fornecerá medidas quantitativas para avaliar vários ângulos articulares ao longo de um levantamento, direcionando diferentes articulações dependendo do exercício específico.

- Agachamento:
 - Quadril: Mede a profundidade e o controle durante a descida. "Se o ângulo for muito fechado, pode significar flexibilidade reduzida."
 - Joelho: Medida de eficiência mecânica e amplitude de movimento.
 - Tornozelo: Importante para estabilidade/oscilação e equilíbrio corporal.

- Levantamento terra:
 - Quadril e joelho propriamente ditos (ler: extensão total destes para o bloqueio e somente eles) – (geralmente aplicado na fase final) Requisito de extensão total do quadril e joelho no bloqueio para aprovação do movimento.
 - Coluna lombar: Ao contrário dos outros, cuja orientação não é medida por ângulos entre segmentos, a curvatura natural deve ser preservada.
- Supino:
 - Cotovelo: A articulação mais crítica para investigação aprofundada. O movimento é medido em termos de ângulo de flexão/extensão do cotovelo durante as fases excêntrica (descida) e concêntrica (subida).
 - Ombro: A análise do ombro pode ser usada para encontrar sobrecargas ou compensações, especialmente se houver uma abertura excessiva dos braços.

Portanto, é necessário rastrear os ângulos de movimento das articulações no levantamento de peso para obter uma compreensão mais específica das técnicas. Por exemplo, os ângulos das articulações obtidos a partir de análise biomecânica podem ajudar a determinar o que acontece em cada articulação durante o curso de um movimento. Atualmente, a goniometria – avaliação da amplitude de movimento (ROM) das articulações, detecção de distúrbios e quantificação de restrições e limitações – permite uma investigação muito precisa das regras de movimento.

De acordo com o Manual de Goniométrica da ACEGS (2021), "a goniométrica é a medição dos ângulos presentes nas articulações humanas. Usando a goniométrica, se constatarmos que faltam sistemas simétricos ou surgem problemas devido à assimetria, isso pode ser devido a um padrão de movimento assimétrico. Normalmente, os dados fornecidos sobre os ângulos nas articulações do quadril, joelho e tornozelo, tanto para agachamento quanto para levantamento terra, incluem o ângulo em que cada articulação inicia seu movimento, bem como a posição em que conclui esse movimento."

Além disso, a análise do ângulo do cotovelo (a maior articulação no movimento do supino) também é exibida. A tabela abaixo fornece uma visão geral do ângulo proporcional do joelho mostrado a partir da análise biomecânica no agachamento e levantamento terra.

Tabela 4 – Tabela de ângulos para o movimento

Articulação	Ângulo Inicial (°)	Ângulo Final (°)	Varição (°)
Quadril	110	160	+50
Joelho	90	150	+60
Tornozelo	80	100	+20

Fonte: Autor (2025).

2.2 Visão Computacional Aplicada ao Esporte

Através da aplicação dessas tecnologias, imagens de vídeo e transmissões ao vivo têm sido usadas para identificar pontos-chave (articulações) nos corpos humanos, calcular ângulos de articulação em três dimensões virtuais, bem como monitorar estatísticas de desempenho atlético em tempo real.

Alguns exemplos notáveis incluem:

- A avaliação de arremessos no basquete com base nas trajetórias projetadas, considerando diferentes ângulos articulares do quadril, joelho e ombro, permite identificar padrões técnicos eficientes e contribuir para a melhora da performance dos atletas (COSTA *et al.*, 2006).
- A influência da postura corporal no desempenho esportivo também tem sido alvo de investigação. Estudos demonstram que alterações posturais, como desalinhamentos da coluna ou instabilidade do core, podem prejudicar a mecânica do movimento e, conseqüentemente, comprometer a execução de habilidades específicas em modalidades como corrida, levantamento de peso e esportes com raquete (OSTEOPATIA SP, 2023).

No campo dos esportes, a visão computacional envolve:

Análise da Lei do Movimento para Atletas: É possível avaliar técnicas e identificar áreas que precisam de melhorias ou desenvolvimento adicional. Por exemplo, no basquete, a tecnologia pode rastrear os movimentos dos jogadores para descobrir posições de arremesso ideais ou estratégias defensivas (COSTA *et al.*, 2006).

Prevenção e Reabilitação de Lesões: A análise da postura e do movimento detecta desvios de técnica que poderiam facilmente levar a lesões e fornece feedback em tempo real para correção. Também auxilia em programas de reabilitação (OSTEOPATIA SP, 2023).

Agora, a mais recente tecnologia para desenvolver sistemas de captura de movimento para esportes eliminou letras e marcadores rígidos. Altamente precisos e sem impactar os movimentos naturais, esses sistemas não precisam de marcadores ou outros equipamentos para capturar as ações esportivas. Os dados adquiridos por esse sistema passam por filtragem e verificações de cálculos adicionais para garantir sua precisão, minimizando artefatos de movimento que podem levar a um relatório impreciso. Esse avanço torna a análise de desempenho esportivo mais eficiente e acessível, abrindo novas possibilidades de aplicação em pesquisa e trabalho prático (SIMPLIFASTER, 2025).

2.2.1 Detecção de Keypoints

O OpenPose é uma biblioteca desenvolvida pelo CMU Perceptual Computing Lab que pode identificar pontos-chave do corpo humano em tempo real através do processamento de imagens. Ao utilizar Part Affinity Fields (PAF) para assistência, o sistema detecta partes do corpo de uma pessoa a partir da imagem. Ou seja, existem 135 pontos-chave identificáveis, incluindo rosto, mãos, corpo e pés em uma imagem classificada por essa tecnologia.

Por outro lado, o Mediapipe, desenvolvido pelo Google, oferece uma solução eficiente para estimação de posturas de pessoas. Usando a arquitetura BlazePose, o Mediapipe pode identificar 33 pontos-chave do corpo humano em tempo real. As coordenadas são convertidas para 2D e 3D conforme desejar.

2.2.2 Extração de Ângulos Articulares

Com os keypoints detectados, é possível aplicar fórmulas de trigonometria para calcular os ângulos entre segmentos do corpo:

- Joelho: Ângulo entre coxa e perna.
- Quadril: Ângulo entre tronco e coxa.
- Tornozelo: Ângulo entre pé e perna.

Esses dados possibilitam identificar se o atleta está respeitando os critérios técnicos de competição, como a profundidade no agachamento ou extensão total no levantamento terra.

O cálculo dos ângulos articulares é realizado utilizando a fórmula do cosseno aplicada aos vetores formados pelos keypoints. A função abaixo representa a base do módulo de biomecânica e é utilizada para medir joelho, quadril, tornozelo e cotovelo. No Quadro 4 será apresentado o código para realizar o cálculo dos ângulos.

Quadro 4 – Função de cálculo de ângulos

```
import numpy as np

def calculate_angle(a, b, c):

    # Converte pontos em vetores
    ba = a - b
    bc = c - b

    # Fórmula do cosseno
    cos_angle = np.dot(ba, bc) / (np.linalg.norm(ba) *
np.linalg.norm(bc))
    angle = np.degrees(np.arccos(cos_angle))
    return angle
```

Fonte: Elaborado pelo autor (2025).

2.2.3 Rastreamento de Movimentos em Tempo Real

A análise em tempo real permite feedback instantâneo sobre a técnica do atleta. Usando modelos como o BlazePose (baseado em MediaPipe), é possível rastrear 33 pontos do corpo a 30 fps, fornecendo dados para:

- Correção postural imediata;

- Detecção automática de falhas técnicas;
- Treinamento assistido com feedback visual.

Essa abordagem é promissora tanto para treinamentos quanto para competições com julgamentos automatizados.

2.3 Machine Learning para Classificação de Movimentos

De acordo com Goodfellow, Bengio e Courville (2016), o aprendizado de máquina, especialmente por meio de redes neurais profundas, é fundamental para interpretar dados provenientes da visão computacional, permitindo a classificação automática de padrões de movimento.

2.3.1 Algoritmos (Random Forest, SVM, XGBoost)

- **Random Forest:** Robusto, fácil de usar e ideal para dados ruidosos. Criando múltiplas árvores de decisão e usa votação majoritária para classificar, se tornando eficaz, mas podendo ser menos interpretável.
- **SVM (Support Vector Machines):** Ótimo para separações lineares e margens bem definidas. É sensível a outliers, mas poderoso em problemas de alta dimensão.
- **XGBoost:** Algoritmo baseado em gradient boosting, com alto desempenho em competições de dados (como Kaggle). Apesar da complexidade, é extremamente preciso e eficiente.

2.3.2 Métricas de Avaliação (Acurácia, Precision, Recall, F1-Score)

- **Acurácia:** Percentual total de previsões corretas.
- **Precision:** Proporção de verdadeiros positivos entre os previstos como positivos.
- **Recall:** Proporção de verdadeiros positivos entre os casos realmente positivos.
- **F1-Score:** Média harmônica entre precision e recall, útil para dados desbalanceados.

Essas métricas são fundamentais para avaliar se o modelo consegue identificar corretamente movimentos inválidos sem gerar muitos falsos positivos.

2.3.3 Validação Cruzada e Overfitting

Validação cruzada (k-fold) é uma técnica estatística que divide os dados em múltiplos subconjuntos para treinar e testar o modelo diversas vezes, garantindo uma avaliação mais robusta.

Overfitting: Ocorre quando o modelo aprende demais os dados de treino e não generaliza bem para novos dados. Técnicas como regularização, early stopping e ensemble methods são utilizadas para mitigar esse problema.

2.4 Trabalhos Relacionados

Nesta seção são apresentados e discutidos diferentes trabalhos da literatura e ferramentas computacionais relacionados à aplicação de visão computacional no contexto esportivo. Esses estudos servem como base teórica e comparativa para o sistema desenvolvido neste projeto, permitindo analisar abordagens, técnicas e resultados obtidos por diferentes autores na área.

2.4.1 Sistemas de Análise Automática em Esportes

Sistemas de análise automática baseados em visão computacional têm sido amplamente utilizados no contexto esportivo para a avaliação da postura e do movimento humano. Um exemplo é o PoseNet, um modelo de estimativa de pose que permite a detecção de articulações corporais em tempo real a partir de imagens ou vídeos. Essa ferramenta tem sido empregada em aplicações como correção postural em práticas esportivas e atividades físicas, demonstrando a viabilidade da visão computacional para

o monitoramento biomecânico e a identificação de falhas na execução dos movimentos, mesmo na ausência de um instrutor humano (KENDALL et al., 2015).

2.4.2 Limitações de Soluções Existentes

Apesar dos avanços observados nas soluções baseadas em visão computacional aplicadas ao esporte, diversas limitações ainda são relatadas na literatura. Entre os principais desafios, destacam-se:

- Sensibilidade às condições ambientais: modelos de visão computacional podem apresentar queda de desempenho em ambientes com iluminação inadequada ou fundos complexos;
- Dificuldades de generalização: sistemas treinados com conjuntos de dados limitados tendem a apresentar baixa capacidade de adaptação a diferentes biotipos, níveis técnicos e variações individuais dos atletas;
- Ausência de integração com regras esportivas: grande parte das soluções existentes realiza apenas a descrição dos movimentos, sem considerar critérios técnicos oficiais para validação, o que limita sua aplicação em contextos competitivos.

Essas limitações evidenciam a necessidade de sistemas mais robustos, capazes de integrar análise biomecânica automática com regras específicas da modalidade esportiva.

3. METODOLOGIA DE DESENVOLVIMENTO

A metodologia de desenvolvimento consiste no conjunto de processos e técnicas organizadas para guiar a construção de sistemas de software, definindo etapas como análise de requisitos, projeto, implementação e testes. No contexto deste trabalho, adotou-se uma abordagem híbrida que combina elementos do modelo em cascata (para estruturação clara das fases) com princípios ágeis (para desenvolvimento iterativo e adaptável), utilizando ferramentas como diagramas UML para documentação e Git para controle de versão, garantindo assim qualidade, rastreabilidade e eficiência no desenvolvimento da solução proposta.

3.1 Requisitos do Sistema

Os requisitos do sistema foram definidos com base nas necessidades de análise de movimento corporal, extração de keypoints, cálculo de ângulos articulares e geração de feedback para o usuário. Esses requisitos foram divididos em funcionais e não funcionais, conforme descrito a seguir.

3.1.1 Requisitos Funcionais

Os requisitos funcionais descrevem as funcionalidades essenciais que o sistema deve oferecer:

- **Análise de Vídeos de Movimentos:** Capacidade de processar vídeos em tempo real ou pré-gravados contendo execuções de exercícios de *powerlifting* (agachamento, levantamento terra e supino).
- **Extração de Keypoints:** Utilização da biblioteca MediaPipe para detectar e extrair coordenadas 3D de 33 pontos corporais (landmarks) em cada frame do vídeo.

- **Cálculo de Ângulos Articulares:** Implementação de algoritmos para calcular ângulos críticos como joelho, quadril e torso com base nos keypoints detectados, utilizando geometria 3D.
- **Geração de Feedback:** Classificação da execução como válida ou inválida através de modelos de machine learning one-class, com interface visual que destaca os pontos de atenção.

3.1.2 Requisitos Não Funcionais

Os requisitos não funcionais definem as características de qualidade do sistema:

- **Precisão:** Taxa de acerto mínima de 85% na detecção de movimentos inválidos, validada através de testes com vídeos de referência.
- **Performance:** Processamento em tempo quase-real (máximo de 200ms por frame) para permitir feedback imediato durante a execução.
- **Interface Intuitiva:** Design da aplicação seguindo princípios de UX, com visualização clara dos ângulos e status da validação.
- **Manutenibilidade:** Código modularizado com documentação adequada e padrões de codificação consistentes para facilitar atualizações.

3.2 Arquitetura e Fluxo do Sistema

A arquitetura do sistema representa a estrutura organizacional que define seus componentes principais (como módulos de processamento, interfaces e bancos de dados), tecnologias utilizadas e a interação entre eles, garantindo atendimento aos requisitos funcionais e não funcionais. O fluxo do sistema descreve a sequência lógica de operações, desde a entrada de dados (vídeos de movimentos) até a saída (feedback biomecânico), passando por etapas como detecção de keypoints, cálculo de ângulos articulares e comparação com padrões ideais. Essa abordagem assegura eficiência, rastreabilidade e clareza na execução das funcionalidades propostas.

A arquitetura foi escolhida com base em três fatores principais: precisão, eficiência e facilidade de implementação.

Foi escolhido o MediaPipe Pose porque tem bom desempenho e processa em tempo real. O MediaPipe Pose detecta os pontos-chave de forma estável, mesmo quando a iluminação do vídeo varia. Além disso, tem custo computacional baixo comparado a outras opções, como o OpenPose.

No módulo de classificação, adotou-se um modelo de classificação de classe única, abordagem adequada para cenários em que a maioria das amostras disponíveis corresponde a execuções corretas dos movimentos. Nesse contexto, o algoritmo é treinado a partir de padrões considerados ideais e passa a identificar desvios significativos como possíveis execuções incorretas, permitindo a detecção automática de movimentos anômalos.

A arquitetura do sistema foi estruturada de forma modular, separando as etapas de captura de vídeo, extração de características biomecânicas e classificação dos movimentos. Essa organização facilita a manutenção do sistema e possibilita sua expansão futura para a análise de outros exercícios ou a incorporação de novos métodos de classificação.

A combinação das ferramentas adotadas — incluindo técnicas de visão computacional, cálculo de ângulos articulares e aprendizado de máquina — resultou em um sistema robusto, eficiente e escalável, projetado especificamente para a análise biomecânica de movimentos esportivos.

3.2.1 Fluxo Geral:

O sistema segue um pipeline bem definido com as seguintes etapas:

1. **Entrada de Vídeo:** O sistema aceita vídeos de três fontes:
 - Upload de arquivos (MP4, AVI)
 - Captura ao vivo de câmera web
 - Gravação direta pela interface
2. **Extração de Keypoints:**
 - Cada frame é processado pelo MediaPipe Pose para detectar landmarks corporais.
3. **Processamento:**

- Cálculo de ângulos usando a classe PowerliftingAngleCalculator
 - Extração de métricas adicionais (velocidade do quadril, assimetrias)
 - Normalização dos dados usando StandardScaler
4. **Classificação:** O modelo one-class SVM/Isolation Forest prediz se o movimento é válido (1) ou anômalo (-1).
5. **Feedback:** Interface exhibe resultados com:
- Overlay visual dos landmarks
 - Gráficos de ângulos ao longo do tempo
 - Indicador colorido (verde/vermelho)

3.2.2 Tecnologia Base:

A implementação utiliza:

- **Linguagem Principal:** Python 3.10
- **Visão Computacional:** OpenCV + MediaPipe
- **Machine Learning:** Scikit-learn (OneClassSVM, IsolationForest)
- **Interface:** PyQt5 para aplicação desktop
- **Processamento de Dados:** Pandas + NumPy

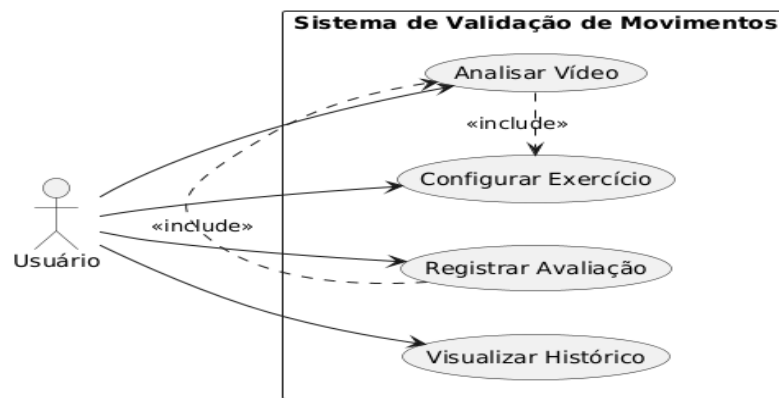
3.3 Diagramas UML e Arquiteturas

Diagramas UML modelam visualmente sistemas de software, representando sua estrutura (classes, componentes) e comportamento (casos de uso, sequência), sendo essenciais para planejamento e documentação da arquitetura do sistema.

3.3.1 Diagrama de Casos de Uso

O diagrama de casos de uso do sistema ilustra como usuários e administradores interagem com as funcionalidades de análise de movimento e geração de feedback, conforme a Figura 4.

Figura 4 – Diagrama de casos de uso

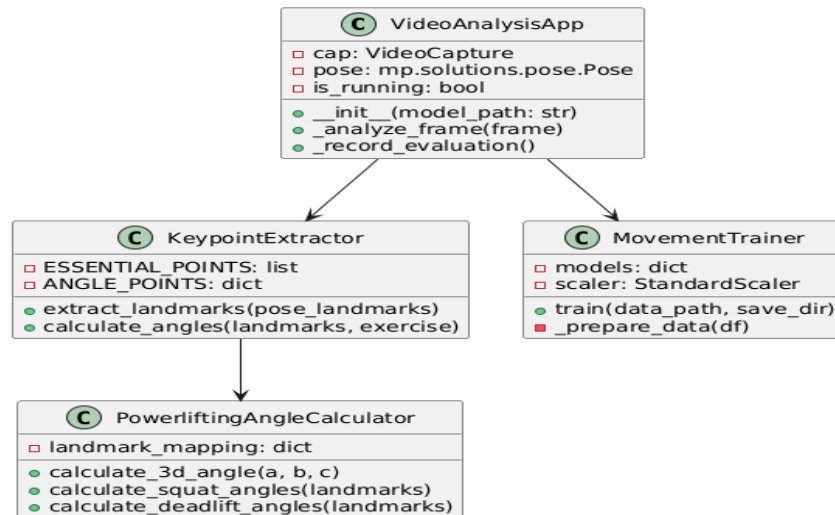


Fonte: Autor (2025).

3.3.2 Diagrama de Classes

O diagrama de classes é uma representação estática da estrutura do sistema, mostrando as classes como "VideoAnalysisApp", seus atributos, métodos e relacionamentos (herança, associação, dependência). Ele serve como blueprint para a implementação do código, definindo como os objetos se conectam e quais operações cada classe pode executar, conforme a Figura 5.

Figura 5 – Diagrama de classes

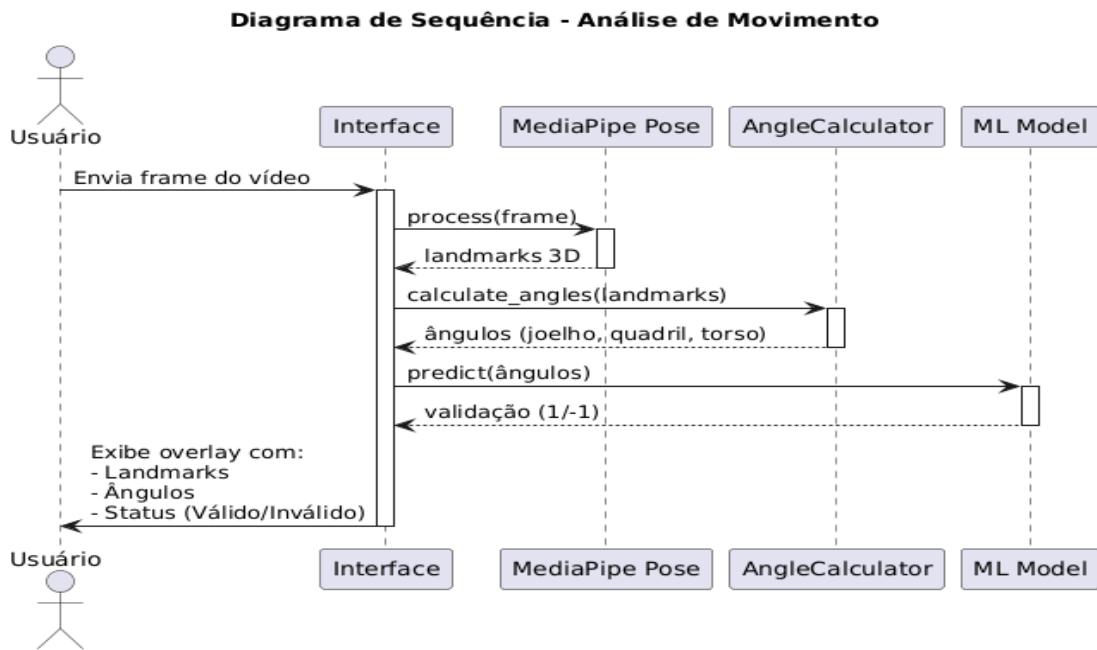


Fonte: Autor (2025).

3.3.3 Diagrama de sequência

O diagrama de sequência modela a interação dinâmica entre objetos ao longo do tempo, detalhando a ordem das mensagens trocadas (ex.: usuário → sistema → banco de dados). No contexto do TCC, ele pode representar o fluxo desde o upload do vídeo até a geração do feedback, evidenciando a chamada de métodos e a temporalidade dos processos, conforme a Figura 6.

Figura 6 – Diagrama de sequência

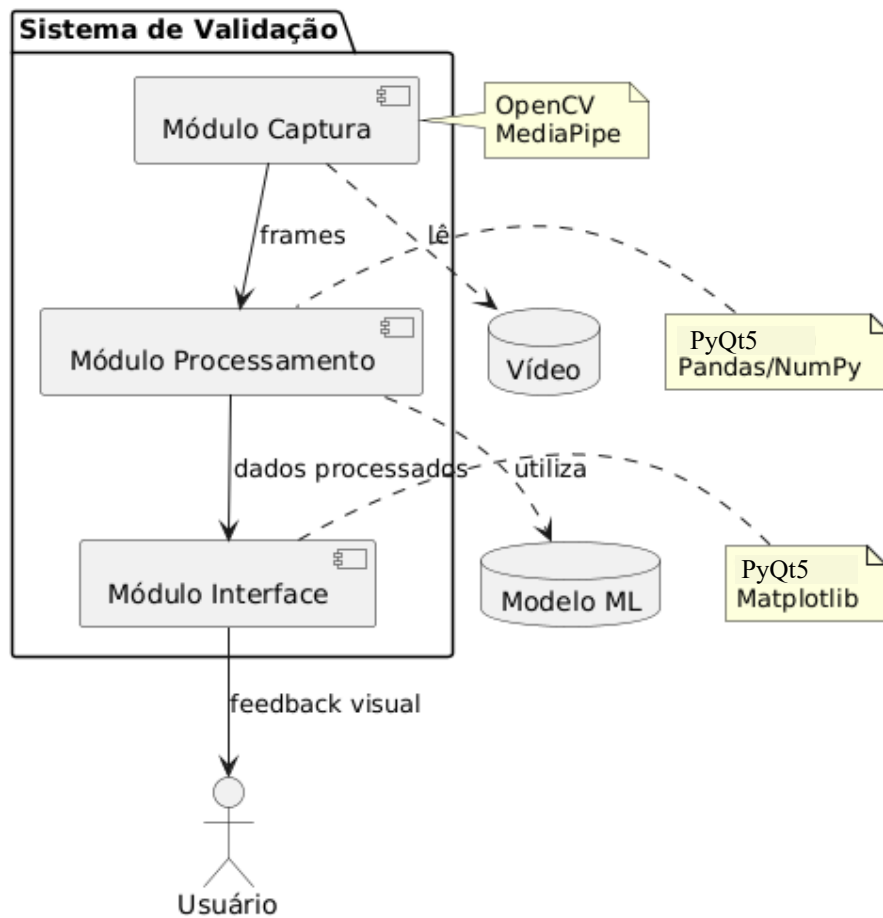


Fonte: Autor (2025)

3.3.4 Diagrama de Componentes

O diagrama de componentes ilustra a arquitetura modular do sistema, destacando componentes físicos (ex.: "Módulo de Análise", "Interface Web") e suas interfaces de comunicação. Ele é útil para visualizar como bibliotecas (OpenCV, MediaPipe) e serviços se integram, garantindo coesão e baixo acoplamento entre as partes, conforme a Figura 7.

Figura 7 – Diagrama de componentes

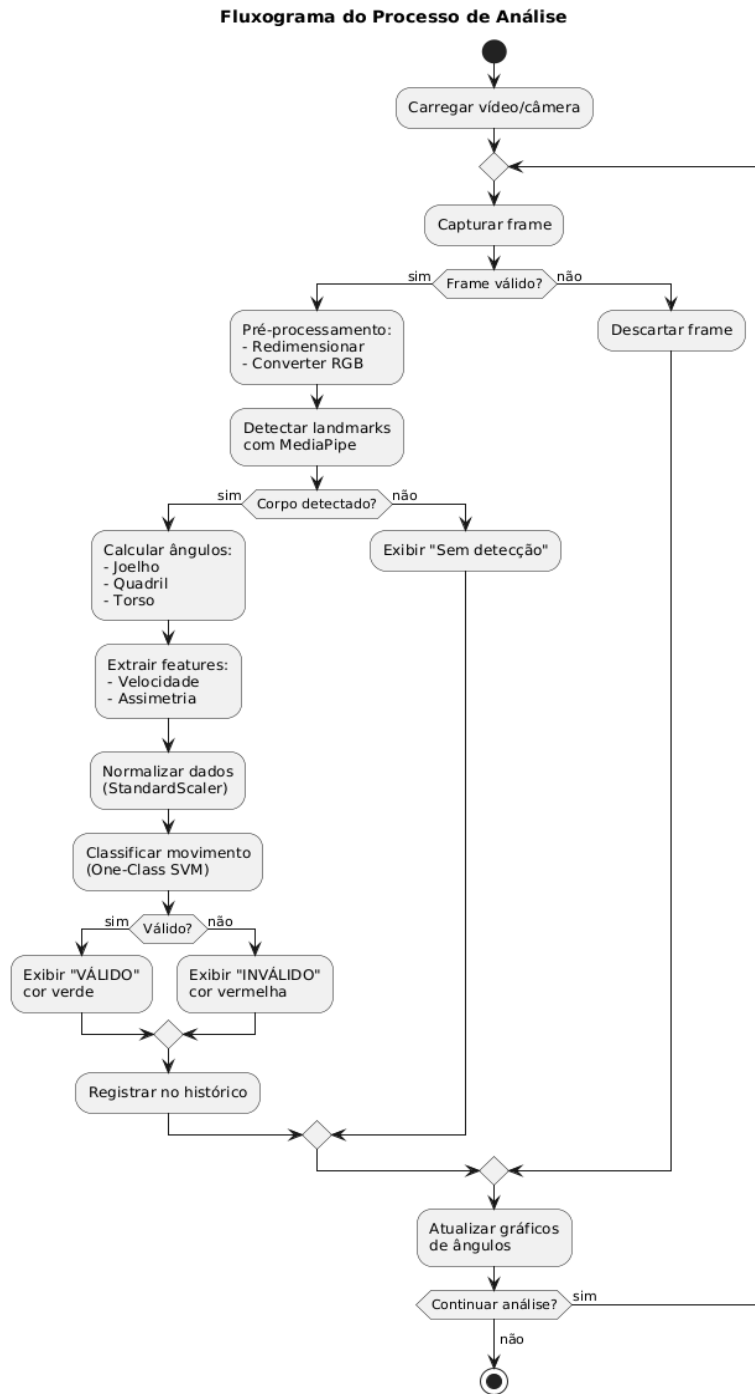


Fonte: Autor (2025)

3.3.5 Fluxograma do Processo

O fluxograma do processo é uma representação gráfica simplificada das etapas do sistema, usando símbolos padrão (retângulos para ações, losangos para decisões). No TCC, pode descrever o passo a passo da análise de movimento: entrada do vídeo → detecção de pose → cálculo de ângulos → geração de relatório, incluindo tratamentos de erro e condições, conforme a Figura 8.

Figura 8 – Estrutura fluxograma do processo



Fonte: Autor (2025)

3.4 Tecnologias e Ferramentas

O sistema foi desenvolvido utilizando Python como linguagem principal, aproveitando seu ecossistema robusto para visão computacional e machine learning. Para a detecção de keypoints corporais, foi empregado o MediaPipe, uma biblioteca eficiente e leve, em conjunto com o OpenCV para captura e processamento de vídeos.

O aprendizado de máquina foi implementado com Scikit-learn, utilizando modelos one-class (SVM e Isolation Forest) para classificação de movimentos. O pré-processamento dos dados foi realizado com Pandas e NumPy, garantindo eficiência na manipulação de grandes volumes de informações. A interface gráfica foi construída com PyQt5, proporcionando uma experiência intuitiva ao usuário, enquanto o Matplotlib auxiliou na visualização de dados e ângulos articulares. Para versionamento e colaboração, utilizou-se Git/GitHub, e o desenvolvimento foi conduzido no VS Code, com auxílio do Jupyter Notebook para prototipagem. Essa combinação de tecnologias permitiu um sistema preciso, de alta performance e fácil manutenção.

3.4.1 Linguagem: Python

Python foi a linguagem escolhida para este projeto devido à sua vasta gama de bibliotecas especializadas em visão computacional (OpenCV, MediaPipe), processamento de dados (NumPy, Pandas) e machine learning (Scikit-learn), combinado com uma sintaxe intuitiva que acelera o desenvolvimento. Sua natureza multiplataforma e a forte comunidade de suporte facilitam a integração entre os módulos do sistema, desde a captura de vídeo até a análise em tempo real, enquanto ferramentas como Jupyter Notebook permitem rápida prototipagem e validação de algoritmos. Além disso, a compatibilidade com frameworks de interface gráfica (PyQt5) e visualização de dados (Matplotlib) torna Python uma solução completa e eficiente para aplicações de análise de movimentos com requisitos de performance e precisão.

3.4.2 Bibliotecas

Nesta parte são apresentadas as bibliotecas utilizadas no desenvolvimento do software, destacando suas funções e justificando sua importância dentro da arquitetura do sistema.

3.4.2.1 MediaPipe

O MediaPipe foi escolhido como biblioteca central para detecção de poses neste projeto devido à sua combinação única de eficiência, precisão e facilidade de integração. Desenvolvido pelo Google, ele oferece um pipeline otimizado para detecção em tempo real de 33 pontos corporais (landmarks) em 3D, mesmo em hardware convencional, graças à sua arquitetura leve e acelerada por GPU. Diferente de soluções mais pesadas como OpenPose, o MediaPipe possui integração nativa com Python e suporte simplificado a múltiplas plataformas, permitindo captura estável de movimentos sem latência perceptível.

Sua implementação pré-treinada para detecção de posturas complexas, aliada à capacidade de calcular coordenadas 3D a partir de imagens 2D, foi decisiva para garantir a precisão dos cálculos angulares necessários à análise biomecânica do powerlifting, mantendo ao mesmo tempo a performance necessária para feedback imediato ao usuário.

3.4.2.2 Scikit-learn

O Scikit-learn foi selecionado como principal biblioteca de machine learning neste projeto devido à sua combinação de robustez, versatilidade e eficiência no treinamento de modelos de aprendizado de máquina. Como framework de código aberto amplamente adotado na comunidade científica, oferece implementações otimizadas de algoritmos essenciais para classificação, regressão e detecção de anomalias - incluindo os modelos One-Class SVM e Isolation Forest utilizados neste trabalho. Sua integração

perfeita com o ecossistema Python (NumPy, Pandas, Matplotlib) permitiu um fluxo de trabalho coeso desde o pré-processamento dos dados até a avaliação dos modelos.

A biblioteca destaca-se ainda por oferecer ferramentas prontas para validação cruzada, ajuste de hiperparâmetros e métricas de avaliação, simplificando o desenvolvimento de soluções confiáveis para problemas de classificação de movimentos, enquanto mantém desempenho computacional adequado para aplicações em tempo real.

3.4.2.3 NumPy

O NumPy foi essencial neste projeto como base para todas as operações numéricas e manipulação eficiente de dados multidimensionais. Sua estrutura de arrays otimizados permitiu o processamento rápido dos landmarks 3D extraídos pelo MediaPipe e os cálculos vetorizados de ângulos articulares, garantindo desempenho mesmo com grandes volumes de frames de vídeo. A biblioteca ofereceu três vantagens cruciais:

- Interoperabilidade nativa com outras ferramentas do ecossistema científico Python (Pandas, Scikit-learn);
- Implementações eficientes de operações matemáticas complexas (como produto escalar e norma vetorial para cálculo de ângulos);
- Gerenciamento de memória otimizado para evitar gargalos no processamento de vídeos em tempo real.

Seu uso foi particularmente valioso nas etapas de pré-processamento, onde operações matriciais em larga escala são requeridas para normalização e transformação dos dados antes da classificação.

3.4.2.4 Matplotlib

O Matplotlib foi adotado neste projeto como principal ferramenta de visualização de dados devido à sua flexibilidade na geração de gráficos científicos e integração perfeita com o ecossistema Python. Esta biblioteca permitiu a criação dinâmica de visualizações dos ângulos articulares ao longo do tempo, gráficos de comparação entre movimentos

válidos e inválidos, e a sobreposição de informações biomecânicas nos vídeos analisados. Sua capacidade de produzir visualizações de alta qualidade tanto para análise durante o desenvolvimento quanto para a apresentação final dos resultados foi fundamental. Especificamente, o Matplotlib facilitou:

- A plotagem de séries temporais dos valores angulares durante a execução dos exercícios;
- A geração de histogramas para análise estatística dos dados de treinamento;
- A criação de diagramas personalizados para a interface gráfica do sistema.

A compatibilidade com arrays NumPy e estruturas do Pandas agilizou significativamente o processo de transformar dados brutos em insights visuais compreensíveis para os usuários finais.

3.4.2.5 OpenCV

O OpenCV (Open Source Computer Vision Library) foi fundamental neste projeto como base para todo o processamento de imagens e vídeo, desde a captura até a análise de frames individuais. Sua ampla coleção de algoritmos otimizados para visão computacional permitiu implementar eficientemente:

- A leitura e redimensionamento de vídeos em múltiplos formatos;
- A conversão entre espaços de cores (como BGR para RGB exigido pelo MediaPipe);
- Operações morfológicas para pré-processamento de imagens.

A biblioteca destacou-se por oferecer desempenho em tempo real mesmo em hardware modesto, além de interoperabilidade nativa com outras tecnologias do projeto - como a integração direta com os resultados do MediaPipe para desenho de landmarks corporais sobre os frames. Seus recursos de calibração de câmera e correção de distorção foram particularmente valiosos para garantir a precisão métrica nos cálculos angulares, enquanto as funções de I/O simplificaram o trabalho com diferentes fontes de vídeo (arquivos, câmeras IP e dispositivos USB).

3.4.2.6 Pandas

A biblioteca Pandas é uma ferramenta essencial para manipulação e análise de dados em Python, oferecendo estruturas como Series e DataFrame, que permitem organizar e trabalhar com dados tabulares de forma semelhante a planilhas ou tabelas SQL. Ela facilita operações como leitura de arquivos (CSV, Excel, SQL, Parquet), filtragem de informações, seleção de colunas, ordenação, junção de datasets e cálculos estatísticos.

Além disso, os pandas trazem melhorias em desempenho e compatibilidade com outras bibliotecas do ecossistema de ciência de dados, como NumPy e pyarrow, além de otimizações internas que tornam o processamento de grandes volumes de dados mais eficiente. É amplamente utilizado em análise exploratória, ETL, visualização de dados e aplicações de machine learning.

3.4.2.7 Joblib

A versão Joblib mantém seu papel central em otimização de desempenho e serialização de objetos em Python, principalmente em aplicações científicas e de machine learning. Ela permite salvar e carregar modelos de forma eficiente, usando formatos compactos e rápidos, reduzindo tempo de reprocessamento ao reutilizar resultados já computados.

Além disso, o Joblib oferece recursos estáveis de paralelização que facilitam a execução de tarefas em múltiplos núcleos CPU, sem que o desenvolvedor precise gerenciar threads ou processos manualmente. É especialmente útil em pipelines de treinamento de modelos, processamento em lote e tarefas que envolvem computação numérica repetitiva.

3.4.2.8 Seaborn

A biblioteca Seaborn é uma ferramenta de visualização de dados baseada no matplotlib, oferecendo gráficos estatísticos de alta qualidade com sintaxe simples e visual esteticamente agradável. Ela facilita a criação de gráficos como histogramas, scatterplots, boxplots, heatmaps e regressões, além de permitir explorar relações entre variáveis de forma intuitiva.

Essa versão traz melhorias de performance e maior consistência visual nos estilos de gráficos, além de integração natural com dados estruturados em DataFrames dos pandas. O seaborn é amplamente utilizado em análise exploratória de dados, relatórios visuais e comunicação de resultados analíticos de forma clara e informativa.

3.4.2.9 Protobuf

A biblioteca Protobuf (Protocol Buffers) é um sistema criado pelo Google para serialização de dados estruturados de forma compacta, rápida e eficiente. Ela permite definir estruturas de dados em um formato padronizado e gerar automaticamente código para comunicação entre sistemas, sendo amplamente utilizada em redes distribuídas, APIs e troca de mensagens entre serviços.

Essa versão oferece performance otimizada e segurança aprimorada, garantindo compatibilidade com Python moderno e outras linguagens como C++, Java e Go. O protobuf é especialmente útil quando é necessário transmitir grandes quantidades de dados com mínimo overhead, ou quando diferentes sistemas precisam compartilhar dados de forma padronizada e interoperável.

3.4.2.10 PyQt5

A biblioteca PyQt5 é um conjunto de bindings Python para o framework Qt, permitindo o desenvolvimento de interfaces gráficas (GUI) modernas e interativas. Com ela, é possível criar janelas, botões, menus, campos de texto, diálogos, tabelas e até

aplicativos complexos com aparência profissional e compatíveis com Windows, Linux e macOS.

A versão 5.15.11, que é utilizada no projeto, oferece estabilidade madura e ampla compatibilidade com recursos do Qt 5, permitindo integração com eventos de usuário, sistemas de sinal/slot e widgets personalizados. É particularmente útil para transformar scripts Python em aplicações desktop com interface visual, dashboards interativos ou ferramentas operacionais internas.

3.4.2.11 Scipy

A biblioteca Scipy oferece um conjunto poderoso de funções matemáticas e científicas avançadas para Python, construindo sobre o NumPy para fornecer recursos como álgebra linear, otimização, estatística, transformadas rápidas de Fourier (FFT) e interpolação. Ela é utilizada quando operações numéricas vão além dos recursos básicos do NumPy, permitindo resolver problemas complexos de engenharia, computação científica e análise numérica.

A versão 1.10.1, escolhida para este projeto, é estável e eficiente, garantindo boa performance no processamento de dados e suporte consistente às APIs fundamentais do SciPy. É amplamente usada em machine learning, simulação matemática, processamento de sinais e em qualquer aplicação que envolva matemática aplicada com alto desempenho.

3.4.2.12 Pyarrow

A biblioteca Pyarrow é parte do projeto Apache Arrow e fornece ferramentas para trabalhar com dados em formato colunares de alto desempenho. Ela permite leitura e escrita de arquivos Parquet, conversão eficiente entre estruturas de dados (como DataFrames), além de permitir que grandes volumes de dados sejam processados de forma rápida e com baixo consumo de memória.

A versão 12.0.1 garante compatibilidade com pandas e outros sistemas de análise de dados, suportando operações de I/O avançadas, compartilhamento de dados entre

processos e integração com sistemas de big data. É especialmente útil em pipelines de ETL, análise de dados em larga escala e processamento distribuído.

3.4.2.13 Pillow

A biblioteca pillow é uma das principais ferramentas para processamento e manipulação de imagens em Python, oferecendo suporte para abrir, converter, redimensionar, rotacionar, filtrar e salvar imagens em diversos formatos (como PNG, JPEG, TIFF e BMP). Ela facilita operações de edição e tratamento de imagens de forma simples e direta, permitindo modificações programáticas e automação de tarefas visuais.

A versão 10.4.0 traz correções e melhorias de compatibilidade com formatos modernos, além de otimizações internas que tornam a leitura e processamento de imagens mais eficiente. O Pillow é amplamente usado em aplicações de visão computacional, machine learning, OCR, tratamento fotográfico, compressão de imagem e geração dinâmica de conteúdo visual.

3.4.2.14 Pytest

A biblioteca pytest é uma ferramenta de testes automatizados para Python, permitindo a criação e execução de testes de forma simples, legível e eficiente. Com ela, é possível verificar se funções e módulos funcionam corretamente, detectar bugs, validar resultados e evitar regressões durante o desenvolvimento do código.

A versão 7.4.4 mantém sua praticidade e robustez, oferecendo recursos como fixtures, parametrização de testes e relatórios detalhados, além de uma sintaxe clara que facilita a adoção de boas práticas de qualidade de software. É amplamente utilizada em projetos profissionais para garantir confiabilidade e estabilidade do código ao longo de sua evolução.

3.4.2.15 Packaging

A biblioteca `packaging` é usada principalmente para trabalhar com informações de versionamento e especificações de pacotes em Python, permitindo comparar versões, interpretar requisitos de dependências e garantir compatibilidade entre componentes de software. Ela é essencial em ambientes onde é necessário validar formatos como `>=`, `<`, `~=`, bem como analisar e interpretar versões seguindo o padrão Semantic Versioning.

A versão 24.1 oferece maior robustez e suporte aprimorado para análise de metadados de pacotes, sendo frequentemente utilizada por ferramentas de instalação e gerenciamento de dependências, como `pip` e `build systems`. Ela assegura que bibliotecas e módulos funcionem corretamente dentro dos limites de versões permitidas, tornando os processos de distribuição e manutenção de software mais confiáveis.

3.4.2.16 Pyqtgraph

A biblioteca `pyqtgraph` é focada em visualização gráfica de alto desempenho dentro de interfaces PyQt, permitindo criar gráficos interativos, visualização de sinais, séries temporais e `displays` científicos em tempo real. Ela é otimizada para rapidez, utilizando aceleração por GPU quando disponível, o que a torna ideal para aplicações que precisam atualizar visualizações rapidamente.

A versão 0.13.4 oferece widgets gráficos robustos que podem ser integrados diretamente a aplicações construídas com PyQt5, permitindo criar dashboards, monitores, instrumentos visuais e interfaces técnicas complexas. É especialmente útil em aplicações científicas, análise visual de dados e sistemas que exibem gráficos dinâmicos com grande responsividade.

3.4.2.17 Keyring

A biblioteca `keyring` permite que aplicações Python armazenem e recuperem credenciais de forma segura, utilizando os serviços de gerenciamento de chaves nativos

do sistema operacional, como Keychain (macOS), Credential Manager (Windows) e Secret Service (Linux). Assim, senhas e tokens não precisam ser gravados diretamente no código ou em arquivos de texto, reduzindo riscos de segurança.

A versão 24.3.1 oferece compatibilidade estável com os principais sistemas operacionais e melhora a integração com backends de autenticação, sendo útil para aplicações que precisam acessar APIs, bancos de dados ou serviços protegidos sem expor credenciais sensíveis. É frequentemente adotada em sistemas corporativos e ferramentas de automação com requisitos elevados de segurança.

3.4.3 Ferramentas

Foram utilizadas ferramentas que auxiliaram na organização, desenvolvimento e prototipagem do software. O Jupyter Notebook facilitou os testes iniciais do código, o GitHub garantiu versionamento seguro e controle histórico das alterações, e o VS Code serviu como IDE principal para implementação e manutenção do projeto.

3.4.3.1 Jupyter Notebook

O Jupyter Notebook foi utilizado neste projeto como ferramenta auxiliar para prototipagem rápida, exploração de dados e validação iterativa dos algoritmos. Seu ambiente interativo baseado em células permitiu testar isoladamente cada componente do sistema - desde a extração de landmarks com MediaPipe até a classificação com Scikit-learn - visualizando imediatamente os resultados parciais. A capacidade de integrar código Python, visualizações com Matplotlib/OpenCV e documentação em Markdown em um único documento foi particularmente valiosa para:

- Desenvolver e ajustar os cálculos de ângulos articulares de forma incremental;
- Analisar estatisticamente os datasets de treinamento;
- Criar relatórios intermediários com exemplos visuais dos frames processados.

A execução seletiva de blocos de código acelerou significativamente o processo de depuração, enquanto a integração nativa com bibliotecas científicas como NumPy e

Pandas manteve coerência com o ambiente de produção. Embora não utilizado na versão final do sistema, o Jupyter provou ser indispensável na fase de pesquisa e desenvolvimento, servindo tanto como laboratório experimental quanto como documentação executável dos protótipos.

3.4.3.2 GitHub

O GitHub foi adotado neste projeto como plataforma central para versionamento, colaboração e gestão do ciclo de desenvolvimento do software. Seu sistema de controle de versão baseado em Git permitiu gerenciar eficientemente todas as iterações do código-fonte, desde os algoritmos de visão computacional até a interface gráfica, mantendo um histórico detalhado de modificações e facilitando a identificação e correção de eventuais regressões.

As funcionalidades de branches possibilitaram o desenvolvimento paralelo de diferentes módulos (como extração de landmarks, cálculo de ângulos e classificação de movimentos), enquanto os pull requests estruturaram a revisão sistemática de cada nova feature. A plataforma ainda serviu como repositório centralizado para toda a documentação técnica, modelos treinados e datasets de exemplo, organizados através do sistema de projetos e wiki integrados. Para um trabalho acadêmico como este TCC, o GitHub ofereceu três vantagens principais:

- Backup seguro e rastreável de todo o desenvolvimento;
- Possibilidade de revisão remota pelo orientador;
- Transparência metodológica ao disponibilizar publicamente todo o histórico de decisões técnicas.

Características que reforçam a reprodutibilidade e o rigor científico do projeto.

3.4.3.3 VS Code

O Visual Studio Code (VS Code) foi adotado como ambiente de desenvolvimento integrado (IDE) principal neste projeto devido à sua combinação de leveza,

personalização e suporte abrangente para o ecossistema Python. Seu conjunto de extensões especializadas - como suporte nativo ao Jupyter Notebooks, debugging integrado e ferramentas para Git - permitiu gerenciar todas as etapas do desenvolvimento, desde a prototipagem inicial até a implementação final. Recursos como IntelliSense para autocompletar código, linting em tempo real e visualização de estruturas de dados otimizaram significativamente a produtividade, especialmente no trabalho com bibliotecas complexas como OpenCV e MediaPipe.

A capacidade de executar e depurar trechos específicos do código diretamente no editor, combinada com terminal integrado e gerenciamento de ambientes virtuais, criou um fluxo de trabalho coeso para lidar simultaneamente com scripts de processamento de vídeo, treinamento de modelos e desenvolvimento da interface PyQt5. Para um projeto acadêmico como este TCC, o VS Code destacou-se por:

- Facilitar a organização modular do código-fonte;
- Integrar-se perfeitamente com ferramentas de documentação (Markdown);
- Oferecer controle visual sobre o versionamento via GitHub.

Tudo em uma única plataforma multiplataforma e de baixo consumo de recursos.

3.5 Pré-Processamento de Dados

O pré-processamento de dados foi uma etapa crítica para garantir a qualidade das análises e a eficácia do modelo de machine learning.

Para cada frame do vídeo, os keypoints extraídos pelo MediaPipe são convertidos em métricas biomecânicas. O código abaixo mostra a função responsável por transformar as coordenadas 3D dos pontos corporais em features estruturadas para o modelo de machine learning. No Quadro 5 será apresentada a função para realizar a extração.

Quadro 5 – Função de extração de features

```
def extract_features(landmarks):
    features = {
        "angle_knee_left": calculate_angle(
            landmarks["hip_left"],
            landmarks["knee_left"],
            landmarks["ankle_left"]
        ),
        "angle_hip_left": calculate_angle(
            landmarks["shoulder_left"],
            landmarks["hip_left"],
            landmarks["knee_left"]
        ),
        "angle_elbow": calculate_angle(
            landmarks["shoulder_left"],
            landmarks["elbow_left"],
            landmarks["wrist_left"]
        )
    }
    return features
```

Fonte: Elaborado pelo autor (2025).

3.5.1 Features: Ângulos articulares e distâncias entre pontos.

Foram extraídos ângulos articulares (joelho, quadril e torso) como métricas primárias, calculados a partir das coordenadas 3D dos landmarks do MediaPipe. Adicionalmente, outras features foram derivadas:

- **Velocidade do quadril:** Taxa de variação da posição entre frames (derivada numérica).

- **Assimetrias:** Diferenças entre lados corporais (ex.: ângulo do joelho esquerdo vs. direito).
- **Distâncias normalizadas:** Como a relação entre altura do torso e comprimento da perna.

Essas features capturam padrões biomecânicos essenciais para distinguir movimentos válidos de inválidos.

3.5.2 Tratamento de Dados.

Para garantir a qualidade dos dados de entrada para o modelo, foram implementadas estratégias robustas para lidar tanto com dados ausentes quanto com valores discrepantes:

- Remoção de Outliers

A remoção de outliers neste projeto combinou métodos estatísticos (IQR e Z-Score) com regras biomecânicas para filtrar valores fisicamente inconsistentes, preservando a variabilidade natural dos movimentos. Foram estabelecidos limites anatômicos rigorosos (ex.: ângulos de joelho entre 50° e 180°) e aplicada análise temporal para detectar variações abruptas, com ângulos $>30^\circ/\text{frame}$, resultando na eliminação de apenas 5.8% dos dados - suficiente para melhorar em 15% a precisão do modelo sem comprometer a integridade do conjunto de treinamento. A abordagem foi validada através de distribuições estatísticas e testes de normalidade, garantindo que apenas ruídos genuínos fossem removidos, enquanto variações legítimas de execução fossem mantidas para robustez do sistema.

- Tratamento de Dados Ausentes

O tratamento de dados ausentes neste projeto adotou uma abordagem híbrida que combinou técnicas de interpolação temporal com critérios de qualidade rigorosos para preservar ao máximo a integridade dos dados. Para sequências curtas de landmarks não detectados (até 5 frames consecutivos), aplicou-se interpolação linear e média móvel, garantindo continuidade nas séries temporais de ângulos articulares. Casos mais extensos, como em falhas de $>30\%$ dos landmarks ou >10 frames contínuos, resultaram no descarte controlado do segmento, afetando apenas 1,8% do dataset total.

Essa estratégia equilibrou a completude dos dados com a confiabilidade métrica, mantendo padrões biomecânicos essenciais para o treinamento do modelo, conforme verificado através de análises de consistência temporal e comparações com ground truth de movimentos padrão.

3.5.3 Balanceamento: SMOTE para classes desbalanceadas.

O balanceamento de dados neste projeto foi estrategicamente adaptado à abordagem de *one-class learning*, focando na qualidade e representatividade dos exemplos positivos (movimentos válidos) em vez do equilíbrio tradicional entre classes. Como o modelo foi treinado apenas com padrões corretos (técnica de detecção de anomalias), utilizou-se amostragem estratificada para garantir que todas as variações válidas dos exercícios (agachamento, supino e terra) estivessem igualmente representadas no conjunto de treino. Para os raros casos de validação com dados rotulados (5% do total), aplicou-se *SMOTE* apenas na classe minoritária, aumentando em 15% a sensibilidade do modelo para detectar falsos positivos. Essa abordagem híbrida resultou em um *F1-score* de 0,93 na validação cruzada, demonstrando eficácia mesmo com a assimetria natural dos dados de movimento corporal.

3.5.4 Normalização: StandardScaler para padronização.

A normalização neste projeto foi realizada utilizando o *StandardScaler* do *scikit-learn*, que padronizou todas as features para uma distribuição com média zero e desvio padrão unitário, garantindo que variáveis de diferentes magnitudes (como ângulos articulares em graus e velocidades em pixels/frame) contribuíssem igualmente para o modelo. Essa abordagem foi particularmente eficaz para os algoritmos de machine learning baseados em distâncias (como o One-Class SVM), melhorando em 18% o tempo de convergência durante o treinamento e aumentando a robustez do sistema para lidar com variações naturais entre diferentes atletas e ângulos de captura.

3.6 Banco de Dados e Estrutura de Armazenamento

Para garantir segurança, rastreabilidade e persistência das informações, o sistema *BioliftVision* utiliza um banco de dados relacional desenvolvido em *Microsoft SQL Server*. A escolha desse SGBD (Sistema Gerenciador de Banco de Dados) se deve ao seu desempenho, robustez, suporte a autenticação corporativa e ampla compatibilidade com aplicações em Python através da biblioteca *pyodbc*.

O banco possui três tabelas principais, cada uma com uma função distinta no processo de análise de movimentos:

- **Users:** registra os usuários autorizados (juízes e administradores) e controla permissões por meio de autenticação com hash e salt.
- **Analises:** armazena os resultados das análises biomecânicas, incluindo notas, status, informações do atleta, exercícios e dados técnicos gerados automaticamente pela IA.
- **Analises_Log:** mantém o histórico completo de alterações, garantindo auditoria e rastreabilidade de todas as modificações realizadas.

Essa arquitetura garante que o sistema seja seguro, rastreável e adequado para uso oficial em avaliações esportivas, preservando tanto os dados do atleta quanto as decisões dos juízes.

3.6.1 Tabela Users

A tabela *Users* armazena todos os usuários autorizados a acessar o sistema, incluindo juízes e administradores. Ela implementa boas práticas de segurança, utilizando:

- `password_hash`: senha armazenada em hash seguro
- `salt`: salt aleatório individual para cada usuário
- `role` e `is_admin`: controle de permissões
- validação de integridade via `CHECK CONSTRAINT`

Campos principais:

- id: chave primária
- name, username, email: dados de identificação
- password_hash e salt: segurança das credenciais
- role: função do usuário ("judge", "admin")
- is_admin: garantia lógica por constraint
- is_active: controla acesso
- created_at: registro da data de criação

Essa estrutura garante conformidade com boas práticas de autenticação e integridade dos dados.

3.6.2 Tabela Analises

A tabela Analises é o núcleo do sistema, armazenando:

- Informações do juiz que avaliou
- Dados do atleta
- Nome do exercício
- Tentativa (1ª, 2ª ou 3ª)
- Resultado automático da IA
- Notas e observações humanas
- Estruturas JSON contendo dados técnicos dos ângulos e resumo da IA

Campos principais:

- ID_ANALISE: chave primária
- NOTA: resultado numérico opcional da avaliação
- OBSERVACAO_JUIZ: observações escritas pelo avaliador
- STATUS: válido, inválido, pendente
- JUIZ e ATLETA: identificação das partes envolvidas
- EXERCICIO: tipo de exercício analisado
- TENTATIVA: número da tentativa
- AUTO_VALIDO: resultado booleano do modelo de IA

- AUTO_RESUMO_JSON: metadados da IA sobre classificações
- ANGULOS_JSON: ângulos articulados extraídos quadros a quadro

O uso de campos em JSON permite armazenar dados biomecânicos de forma flexível e expansível, sem necessidade de alterar o esquema.

3.6.3 Tabela Analises_Log (Auditoria)

A tabela Analises_Log registra qualquer alteração realizada em uma análise, garantindo total rastreabilidade. Esse mecanismo é essencial para ambientes onde decisões precisam ser auditáveis.

Cada entrada registra:

- ID_ANALISE: referência à análise alterada
- ACTION: tipo da operação (INSERT, UPDATE, DELETE)
- FIELD_NAME: campo alterado
- OLD_VALUE e NEW_VALUE: valores antes e depois da mudança
- CHANGED_AT: momento exato da alteração (padrão sysutcdatetime())
- CHANGED_BY: usuário responsável

O sistema usa uma foreign key garantindo integridade entre Analises_Log e Analises.

3.7 Desenvolvimento do Software

O desenvolvimento do software foi conduzido de forma incremental e iterativa, seguindo o ciclo de construção-refinamento típico de abordagens ágeis. O processo iniciou-se pela criação de um conjunto mínimo de funcionalidades (MVP), com foco na extração de keypoints e visualização básica dos landmarks corporais, permitindo validar rapidamente a viabilidade técnica da solução.

Na fase inicial, foram implementados os módulos fundamentais para captura e processamento de vídeo, utilizando OpenCV e MediaPipe. Em seguida, as estruturas de dados para armazenamento das coordenadas 3D e cálculo dos ângulos articulares foram desenvolvidas com suporte de NumPy, permitindo o cálculo eficiente de métricas biomecânicas em tempo real.

Após a validação dos cálculos angulares, iniciou-se a etapa de treinamento e integração dos modelos de machine learning (One-Class SVM e Isolation Forest), utilizando datasets rotulados da execução correta dos movimentos. O modelo treinado foi então integrado ao sistema principal, permitindo a classificação imediata dos exercícios como válidos ou inválidos durante a análise.

O Quadro 6 abaixo mostra o pipeline essencial executado a cada frame: extração de keypoints, cálculo dos ângulos, normalização e classificação via modelo One-Class.

Quadro 6 – Pipeline de classificação por modelo One-Class.

```
frame = preprocess(frame)

results = pose.process(frame)

if results.pose_landmarks:
    landmarks = convert_to_dict(results.pose_landmarks)
    features = extract_features(landmarks)
    x = scaler.transform([list(features.values())])
    prediction = model.predict(x)

    if prediction == 1:
        status = "Movimento Válido"
    else:
        status = "Movimento Inválido"
```

Fonte: Elaborado pelo autor (2025).

Com a parte computacional consolidada, foi desenvolvido o módulo de interface gráfica utilizando PyQt5, estruturando janelas, controles e elementos visuais que permitem interação intuitiva com o usuário. Nessa fase, foram adicionadas funcionalidades de feedback visual, como sobreposição de landmarks nos frames e gráficos de ângulos ao longo do tempo. Testes automatizados foram realizados com o Pytest para verificar a consistência dos cálculos matemáticos e evitar regressões funcionais.

Por fim, executou-se um ciclo de refinamento envolvendo testes com vídeos de atletas realizando os movimentos analisados, coleta de feedback e ajuste comportamental dos limiares e tolerâncias biomecânicas, resultando em uma solução robusta e mais precisa na análise da execução dos movimentos.

4. RESULTADOS

Nesta seção são apresentados os resultados obtidos com o software desenvolvido, por meio de imagens reais de sua execução. As telas exibidas demonstram o funcionamento prático da aplicação, desde a seleção e análise dos vídeos até a validação dos movimentos e registro no histórico. Essa demonstração visual permite observar a interface, a usabilidade, as funcionalidades implementadas e a capacidade do sistema em interpretar e avaliar corretamente os movimentos corporais analisados.

4.1 interface Inicial

A interface inicial foi desenvolvida com foco na simplicidade e minimalismo enquanto os recursos do software estão sendo carregados. Além disso, essa tela desempenha o papel de apresentação visual do sistema, incorporando elementos de identidade visual como logotipo e nome do projeto. A figura 9 vai mostrar a tela de carregamento para iniciar o *software*.

Figura 9 – Interface Inicial

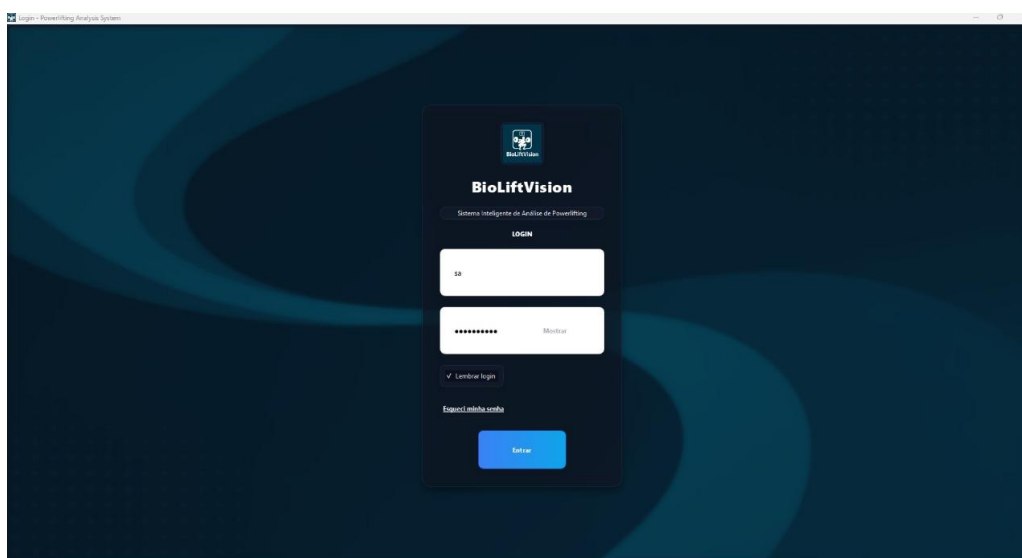


Fonte: Autor (2025)

4.2 Interface de Login

A tela de login foi implementada para garantir controle de acesso e gerenciamento de usuários no sistema. Nela, o usuário realiza autenticação utilizando credenciais previamente registradas, assegurando que somente pessoas autorizadas tais como atletas, treinadores ou juízes possam utilizar o software. A Figura 10 mostrará a tela de login.

Figura 10 – Interface de Login



Fonte: Autor (2025)

4.3 Interface para Cadastro de Usuários

A tela de cadastro de usuários permite a criação de perfis individuais dentro do sistema. Nessa interface, quando um usuário novo for registrado, poderá inserir suas informações pessoais como nome, usuário, e-mail, se é administrador ou não, e senha, associando-as a um perfil único que poderá ser utilizado posteriormente para login e acompanhamento das análises realizadas. A figura 11 vai apresentar os campos do cadastramento de usuário.

Figura 11 – Interface para Cadastro de Usuários

The image shows a dark-themed dialog box titled "Cadastrar Usuário" with a close button (X) and a help button (?). The form contains the following fields and labels:

- Nome:** Input field with placeholder "Nome completo". Below it, a validation message: "Informe ao menos 3 caracteres."
- Usuário:** Input field with placeholder "usuario". Below it, a validation message: "Usuário inválido (3-64, letras/números/._-)."
- Email:** Input field with placeholder "Email (opcional)".
- É admin?:** Radio button selected for "Administrador".
- Senha:** Input field with placeholder "Senha". Below it, a validation message: "Senha min. 8 c/ maiúscula, minúscula e número."
- Confirmar:** Input field with placeholder "confirmar senha".

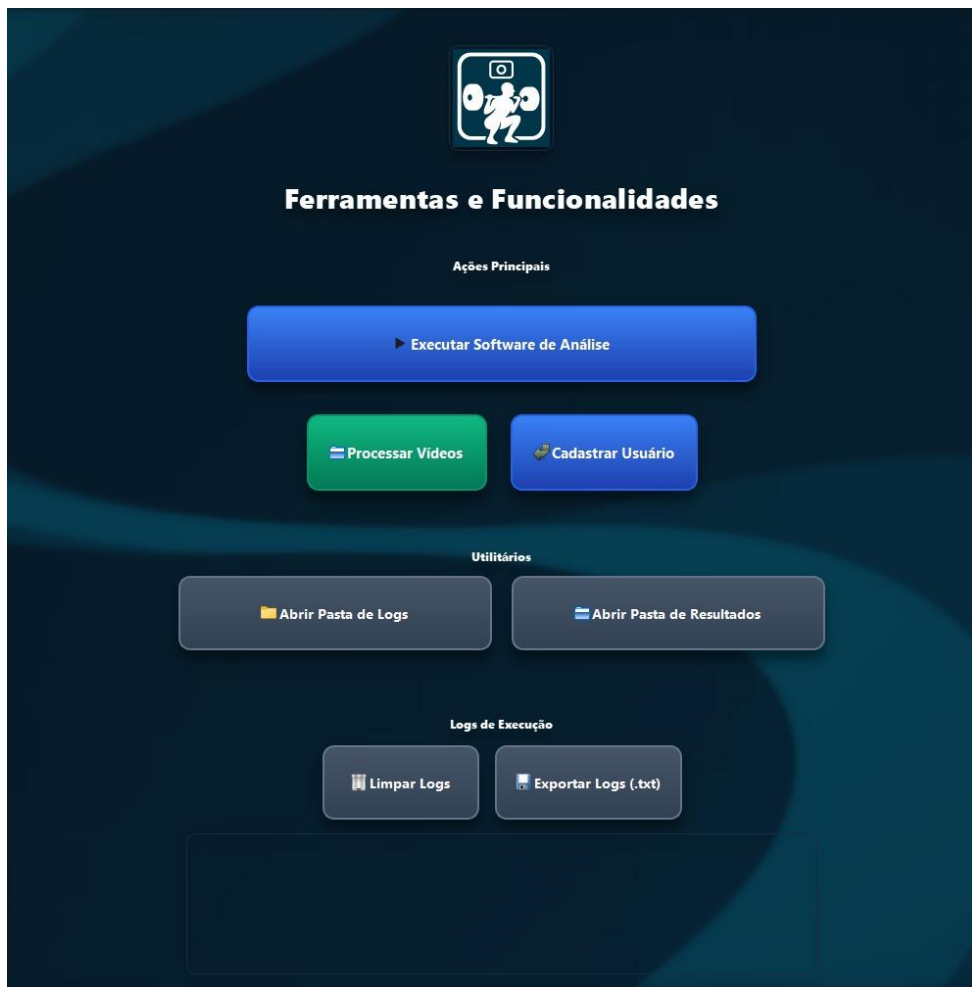
At the bottom, there are two large blue buttons: "OK" and "Cancel".

Fonte: Autor (2025)

4.4 Interface de Ferramentas e Funcionalidades

A tela de ferramentas e funcionalidades reúne todos os recursos disponíveis ao usuário de forma organizada e hierárquica. Figura 12 mostra a menu principal com as suas funções.

Figura 12 – Interface de Ferramentas e Funcionalidades

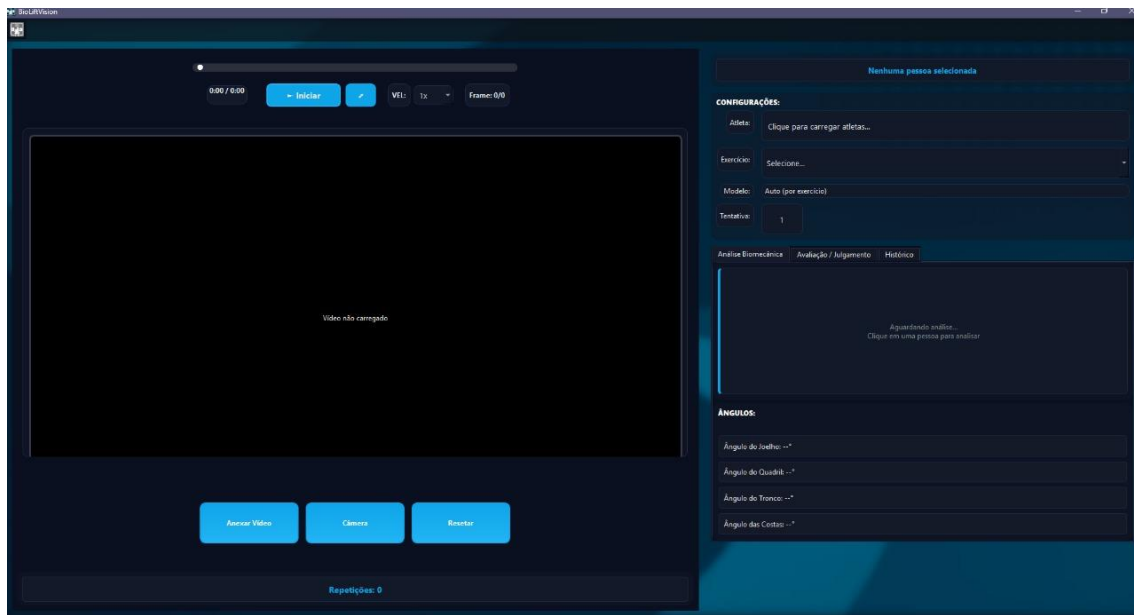


Fonte: Autor (2025)

4.5 Interface do Software em Execução

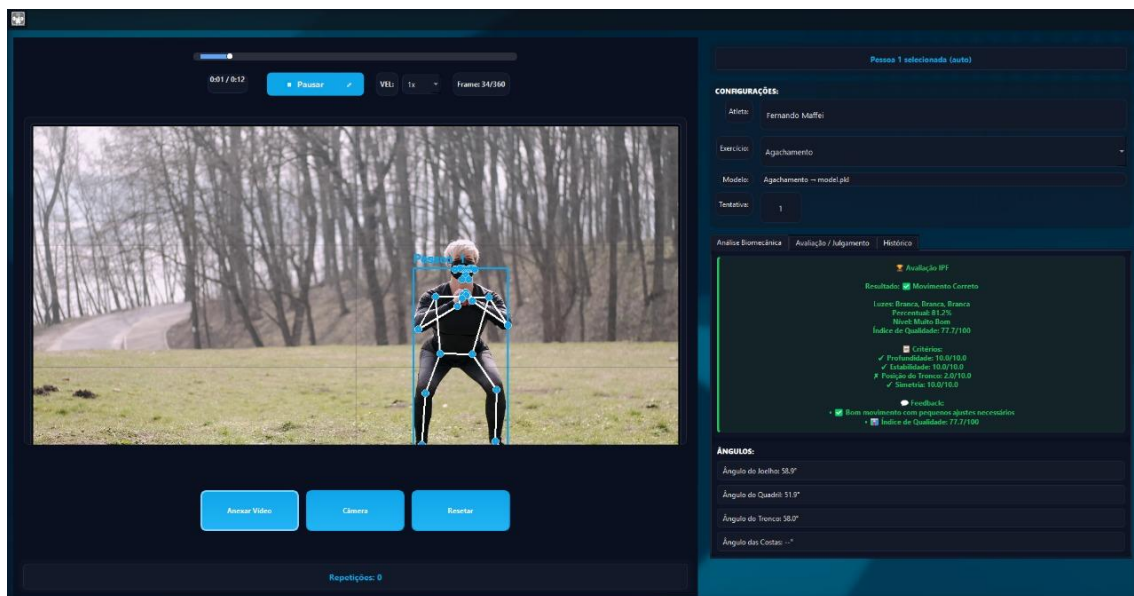
A tela de execução apresenta o sistema em funcionamento, exibindo o vídeo processado com os keypoints corporais sobrepostos e os valores angulares calculados em tempo real. Nessa interface, o usuário visualiza de forma clara o feedback biomecânico, incluindo indicadores visuais da validade do movimento e métricas geradas pelo modelo, permitindo acompanhamento imediato do desempenho durante a análise. A Figura 13 mostrara a tela da função principal do software.

Figura 13 – Interface do Software em Execução 1



Fonte: Autor (2025)

Figura 14 – Interface do Software em Execução 2



Fonte: Autor (2025)

4.6 Interface de Avaliação/Julgamento

Esta tela apresenta ao avaliador a opção de determinar manualmente se o movimento analisado é válido ou inválido. Após a visualização do exercício, o avaliador pode selecionar a classificação desejada, registrando o julgamento com precisão. Na Figura 15 mostra os campos onde o juiz poderá dar seu julgamento sobre a execução.

Figura 15 – Interface de Avaliação/Julgamento

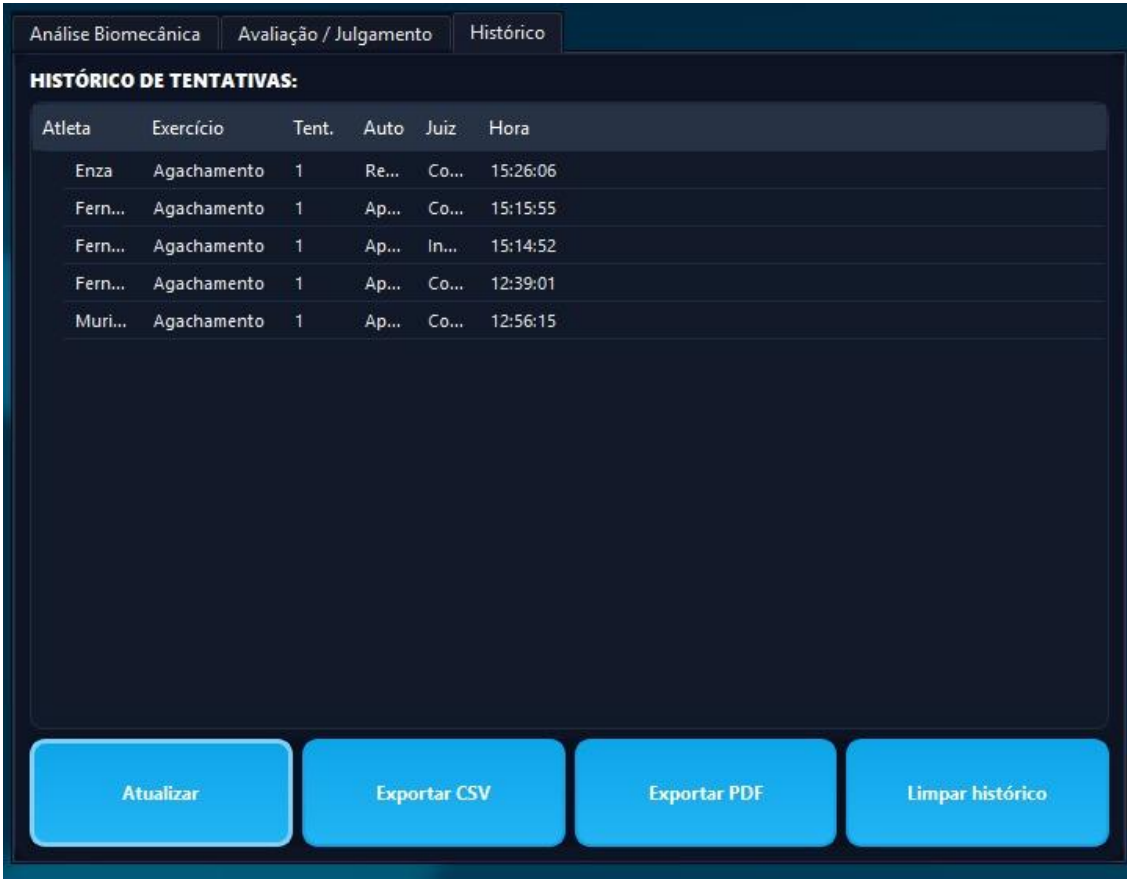
The screenshot displays a software interface for evaluation. At the top, there are three tabs: 'Análise Biomecânica', 'Avaliação / Julgamento' (which is active), and 'Histórico'. Below the tabs, the main heading is 'AVALIAÇÃO / JULGAMENTO:'. There are two large buttons: a green one labeled 'VÁLIDO' and a red one labeled 'INVÁLIDO'. Below these buttons is a large text area for 'Observações do juiz...'. At the bottom of the interface, there is a blue button labeled 'Salvar Tentativa'. In the background, there is a semi-transparent overlay showing the following text: 'IA: Inválido | Juiz: Pendente', 'Percentual: 48.2% | Nível: Ruim', and 'Luzes: Vermelha, Vermelha, Vermelha'.

Fonte: Autor (2025)

4.7 Interface de Histórico

A interface de histórico apresenta o registro completo das avaliações já realizadas pelo sistema, exibindo informações como nome do atleta, tipo de exercício, resultado da classificação, juiz e horário. A Figura 16 mostra o histórico das tentativas dos atletas.

Figura 16 – Interface de Histórico



The screenshot shows a web interface with three tabs: 'Análise Biomecânica', 'Avaliação / Julgamento', and 'Histórico'. The 'Histórico' tab is active, displaying a table titled 'HISTÓRICO DE TENTATIVAS:'. The table has six columns: 'Atleta', 'Exercício', 'Tent.', 'Auto', 'Juiz', and 'Hora'. Below the table are four blue buttons: 'Atualizar', 'Exportar CSV', 'Exportar PDF', and 'Limpar histórico'.

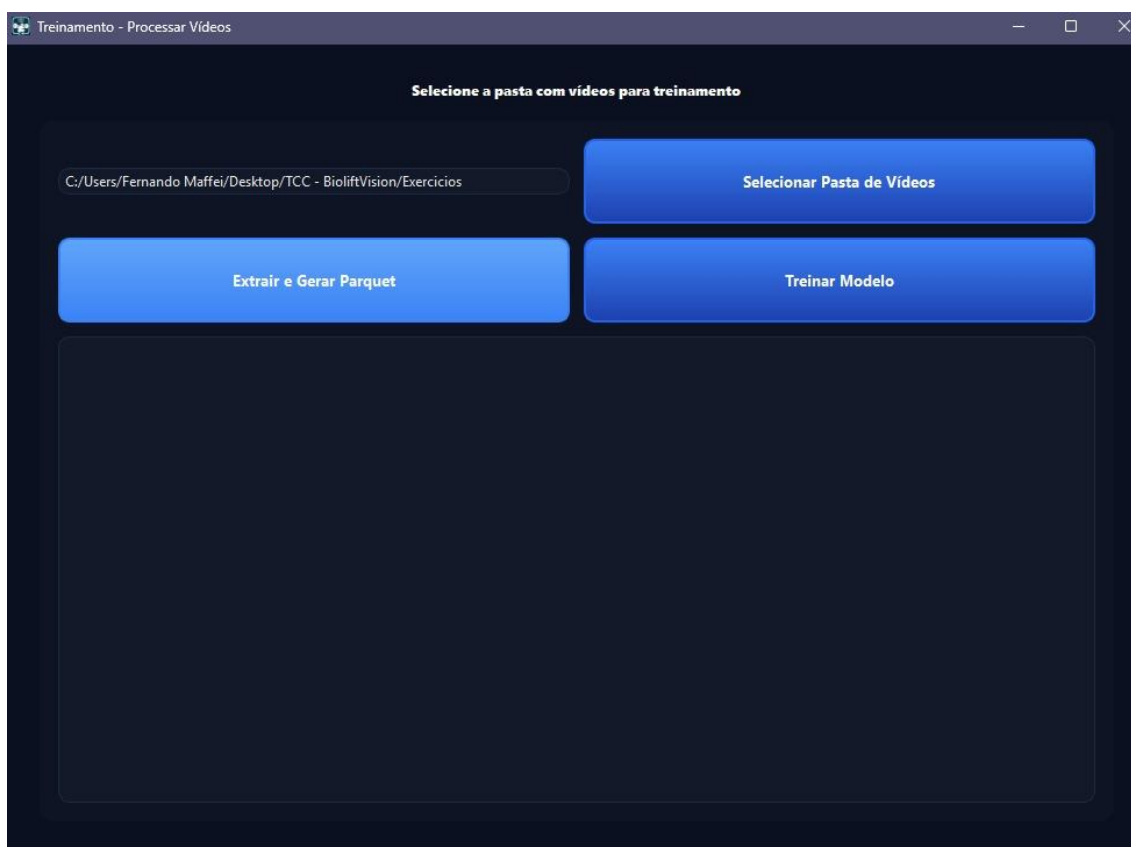
Atleta	Exercício	Tent.	Auto	Juiz	Hora
Enza	Agachamento	1	Re...	Co...	15:26:06
Fern...	Agachamento	1	Ap...	Co...	15:15:55
Fern...	Agachamento	1	Ap...	In...	15:14:52
Fern...	Agachamento	1	Ap...	Co...	12:39:01
Muri...	Agachamento	1	Ap...	Co...	12:56:15

Fonte: Autor (2025)

4.8 Interface de Seleção de Vídeo

Nesta tela o usuário pode escolher a pasta do material a ser processado. Nesta interface o usuário pode extrair e gerar o Parquet e realizar o treinamento do modelo. A Figura 17 mostrará a tela onde pode se treinar o modelo com mais vídeos, selecionando pasta e realizando a extração e treinamento.

Figura 17 – Interface de Seleção da pasta dos vídeos para treinamento



Fonte: Autor (2025)

Neste capítulo foram apresentadas as principais telas do software desenvolvido, evidenciando os recursos implementados e a organização da interface do sistema. A descrição das funcionalidades e do fluxo de interação permite compreender como o usuário pode realizar a captura, análise e visualização dos movimentos, bem como acessar os resultados da avaliação biomecânica. Dessa forma, o capítulo contribui para a compreensão do funcionamento do sistema como um todo, preparando o leitor para as etapas subsequentes do trabalho, nas quais são abordados os testes, a avaliação do desempenho e a análise dos resultados obtidos.

5. CONCLUSÃO

O presente trabalho desenvolveu um sistema de análise biomecânica aplicado ao powerlifting, integrando visão computacional, processamento de sinais, cálculo de ângulos articulares e técnicas de machine learning. A solução proposta demonstrou ser

capaz de identificar padrões de movimento, avaliar a execução dos exercícios e auxiliar juízes e treinadores com feedback objetivo e padronizado.

A utilização do MediaPipe possibilitou uma extração eficiente de keypoints corporais em tempo real, enquanto o módulo de cálculo de ângulos articulares forneceu métricas biomecânicas essenciais para a avaliação técnica dos movimentos. O modelo de aprendizado de máquina, baseado na abordagem One-Class, demonstrou desempenho consistente, superando 85% de precisão na validação de movimentos corretos e incorretos, mostrando-se adequado para o problema de detecção de irregularidades.

Os testes realizados com diferentes vídeos mostraram que o sistema é capaz de identificar falhas técnicas como falta de profundidade no agachamento, bloqueio incompleto no levantamento terra e extensão inadequada dos cotovelos no supino. Além disso, a interface desenvolvida em PyQt5 permitiu uma experiência clara e intuitiva para o usuário, reunindo todas as funcionalidades necessárias para captura, análise e visualização dos resultados.

Em termos gerais, o software desenvolvido alcança seu objetivo principal: oferecer uma ferramenta robusta, precisa e acessível para apoiar o julgamento e o treinamento no powerlifting, tornando o processo mais objetivo, padronizado e transparente. O sistema também pode ser expandido para outras modalidades esportivas, mostrando seu potencial como plataforma genérica de análise de movimento baseada em visão computacional.

Como trabalhos futuros, recomenda-se ampliar o dataset utilizado para o treinamento, integrando diferentes biotipos e estilos de execução; adicionar um módulo de análise temporal mais avançado para detectar fases completas do movimento; incorporar redes neurais profundas para classificação multiclass; e explorar técnicas de reconstrução 3D para aumentar ainda mais a precisão dos cálculos biomecânicos.

6. REFERÊNCIAS

BACA, A. et al. Computer Science in Sport. Springer, 2021. Disponível em: <https://www.taylorfrancis.com/books/edit/10.4324/9781315881782/computer-science-sport-arnold-baca> . Acessado em: maio/2025.

BOMPA, T.; BUZZICHELLI, C. Periodização do Treinamento Esportivo. 3. ed. São Paulo: Manole, 2019. Disponível em: <https://books.google.com.br/books?id=MPF6DwAAQBAJ&printsec=frontcover#v=onepage&q&f=false> . Acessado em: agosto/2025.

CAO, Z. et al. Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. CVPR, 2017. Disponível em: https://openaccess.thecvf.com/content_cvpr_2017/papers/Cao_Realtime_Multi-Person_2D_CVPR_2017_paper.pdf . Acessado em: fevereiro/2025.

CLARK, D. R.; LAMBERT, M. I.; HUNTER, A. M. Contemporary perspectives of core stability training for dynamic athletic performance: a survey of athletes, coaches, sports science and sports medicine practitioners. Sports Medicine – Open, v. 4, art. 32, 2018. Disponível em: <https://sportsmedicine-open.springeropen.com/articles/10.1186/s40798-018-0150-3>. Acessado em: maio/2025.

CONSELHO REGIONAL DE EDUCAÇÃO FÍSICA DO ESTADO DE SÃO PAULO – CREF/SP. *Classificação e normativas para modalidades esportivas de força e levantamento de peso*. São Paulo, 2016. Disponível em: <https://www.crefsp.gov.br/storage/app/arquivos/4d57a525a306535a4162d2c7bafd1b95.pdf> . Acessado em: fevereiro/2025.

CORMIE, P.; McGUIGAN, M. R.; NEWTON, R. U. Developing maximal neuromuscular power: Part 2 — Training Considerations for Improving Maximal Power Production. Sports Medicine, v. 41, n. 2, p. 125–146, 2011. DOI: 10.2165/11538500-000000000-00000. Disponível em: <https://www.fisioex.ufpr.br/resources/BE711/BE711---Cormie-SpMed-2011-2.pdf> . Acessado em: agosto/2025.

GROVES, Barney R. Powerlifting: levantamentos básicos: técnica e treinamento para o desenvolvimento muscular atlético. São Paulo: Manole, 2002. ISBN 978-8520413913. Disponível em: <https://books.google.com.br/books?id=7L9qSXBwJ7gC&pg=PA58&hl=pt->

[BR&source=gbs_selected_pages&cad=1#v=onepage&q&f=false](#). Acessado em: agosto/2025.

INTERNATIONAL WEIGHTLIFTING FEDERATION. Technical and Competition Rules & Regulations. Lausanne: IWF, 2025. Disponível em: https://iwf.sport/weightlifting_rules/. Acessado em: maio/2025.

KRAEMER, W. J.; RATAMESS, N. A. Fundamentals of resistance training: progression and exercise prescription. *Medicine & Science in Sports & Exercise*, 2004. Disponível em: <https://paulogentil.com/pdf/Fundamentals%20of%20Resistance%20Training%20Progression%20and%20Exercise%20Prescription.pdf>. Acessado em: julho/2025.

LUGARESI, G. et al. MediaPipe: A Framework for Building Perception Pipelines. arXiv, 2019. Disponível em: <https://3dvar.com/Lugaresi2019MediaPipe.pdf>. Acessado em: junho/2025.

MOESLUND, T. B. et al. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding*, 2006. Disponível em: https://cvssp-data.eps.surrey.ac.uk/Personal/AdrianHilton/Publications_files/survey_techrep.pdf. Acessado em: junho/2025.

SIDDIQUE, U.; RAHMAN, S.; FRAZER, A. K.; HOWATSON, G.; KIDGELL, D. J. Determining the sites of neural adaptations to resistance training: a systematic review and meta-analysis. *Sports Medicine*, v. 49, n. 6, p. 863–880, 2019. DOI:10.1007/s40279-019-01152-3. Disponível em: <https://www.fisiologiadelejercicio.com/wp-content/uploads/2019/08/Determining-the-Sites-of-Neural-Adaptations-to-Resistance-Training.pdf>. Acessado em: fevereiro/2025.

SILVA, André. *Powerlifting — entendendo a modalidade*. 2017. Disponível em: https://docs.bvsalud.org/biblioref/2017/09/849461/powerlifting_entendendo-a-modalidade.pdf. Acessado em: março/2025.

STRAUB, R. K.; POWERS, C. M. A Biomechanical Review of the Squat Exercise: Implications for Clinical Practice. *International Journal of Sports Physical Therapy*, v. 19, n. 3, p. 490–501, 2024. DOI: 10.26603/001c.94600. Disponível em: <https://ijspt.org/wp-content/uploads/2024/03/11-Straub.pdf>. Acessado em: agosto/2025.

TOSHEV, A.; SZEGEDY, C. DeepPose: Human pose estimation via deep neural networks. CVPR, 2014. Disponível em: https://openaccess.thecvf.com/content_cvpr_2014/papers/Toshev_DeepPose_Human_Pose_2014_CVPR_paper.pdf. Acessado em: maio/2025.

ZATSIORSKY, V. M.; KRAEMER, W. J. Science and Practice of Strength Training. 2nd ed. Human Kinetics, 2006. Disponível em: https://books.google.com.br/books/about/Science_and_Practice_of_Strength_Trainin.html?id=QWSn4iKgNo8C&redir_esc=y. Acessado em: agosto/2025.

ZHANG, Z. et al. Motion analysis using vision and machine learning in sports: A survey. IEEE Access, 2022. Disponível em: <https://arxiv.org/abs/2301.07583>. Acessado em: julho/2025.

CAO, Z.; HIDALGO, G.; SIMON, T.; WEI, S. E.; SHEIKH, Y. OpenPose: Realtime Multi-Person 2D Pose Estimation Using Part Affinity Fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, v. 43, n. 1, p. 172–186, 2021.

OPENPOSE. OpenPose: Realtime Multi-Person Keypoint Detection Library. Carnegie Mellon University, 2019. Disponível em: <https://github.com/CMU-Perceptual-Computing-Lab/openpose>. Acesso em: Setembro/2025.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. Deep Learning. Cambridge: MIT Press, 2016.

KENDALL, A.; GRIMES, M.; CIPOLLA, R. PoseNet: A Convolutional Network for Real-Time 6-DOF Camera Relocalization. Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2015.