

INSTITUTO FEDERAL DA PARAÍBA - IFPB
CURSO DE BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

MARÍLIA DE ARAUJO SANTOS OLIVEIRA
FRANCISCO BEZERRA DA SILVA NETO

Nad.IA: Sistema Inteligente de Monitoramento e Correção de Exercícios Físicos em
Tempo Real via Visão Computacional

CAMPINA GRANDE
2025

MARÍLIA DE ARAUJO SANTOS OLIVEIRA
FRANCISCO BEZERRA DA SILVA NETO

Nad.IA: Sistema Inteligente de Monitoramento e Correção de Exercícios Físicos em
Tempo Real via Visão Computacional

Monografia apresentada ao Curso de Bacharelado em Engenharia da Computação do Instituto Federal de Educação, Ciência e Tecnologia da Paraíba, como requisito para obtenção do título de Bacharel em Engenharia da Computação.

Orientador: Elmano Ramalho Cavalcanti

CAMPINA GRANDE

2025

Catálogo na fonte:
Ficha catalográfica elaborada por Gustavo César Nogueira da Costa - CRB 15/479

O 48n Oliveira, Marília de Araujo Santos

Nad.IA: Sistema inteligente de monitoramento e correção de exercícios físicos em tempo real via visão computacional / Marília de Araujo Santos Oliveira; Francisco Bezerra da Silva Neto. - Campina Grande, 2025.

62f. : il.

Trabalho de Conclusão de Curso (Curso Superior de Engenharia de Computação) - Instituto Federal da Paraíba, 2025.

Orientador: Prof. Elmano Ramalho Cavalcanti.

1. Engenharia de computação - Visão computacional
2. Ginástica laboral 3. Exercícios físicos – monitoramento. I. Silva Neto, Francisco Bezerra da II. Cavalcanti, Elmano Ramalho III. Título.

CDU 004.93:796

De Francisco Neto:

Dedico o presente trabalho a minha família, em especial aos meus pais, irmã e avó que me sustentaram e aceitaram a distância para que eu me formasse.

A minha esposa e par neste projeto, que é, muitas vezes, minha força ao acordar e o motivo das minhas melhores risadas.

Por fim, mas acima de tudo, a Jesus Cristo que salvou minha vida.

De Marilia Araujo:

Dedico este trabalho à minha mãe, que sempre foi e continua sendo meu apoio na educação e na vida. Que eu possa honrar todo o seu esforço e suor por meio da oportunidade de continuar aprendendo.

Ao meu pai, por todo amor, carinho e pelos ensinamentos na lei do Senhor.

Ao meu esposo, por todo apoio, abraços e consolo na jornada academica.

Às minhas irmãs, por me ensinarem a ter paciência e a saber dividir.

E a Deus, por me conceder essa rede de apoio tão preciosa.

RESUMO

O crescente trabalho remoto tem elevado a incidência de Distúrbios Osteomusculares Relacionados ao Trabalho (DORTs). Embora a ginástica laboral seja uma solução comum, sua eficácia no formato online é severamente comprometida pela falta de atenção individualizada do instrutor e pelo uso de câmeras desligadas pelos praticantes. Para superar essa limitação, este trabalho propôs e desenvolveu um protótipo de plataforma que utiliza visão computacional para monitorar e corrigir os exercícios em tempo real, sem a necessidade de armazenamento de imagens. A plataforma visa incentivar a prática correta e regular de exercícios físicos remotos. Os resultados experimentais foram majoritariamente positivos, indicando que 100% dos usuários se sentiram mais motivados com o uso da ferramenta. O sistema de feedbacks e alertas de segurança foi considerado útil por 100% e 75% dos participantes, respectivamente, para a correta execução dos movimentos. Embora 75% tenham percebido pequeno lag e 50% tenham notado dificuldades pontuais na captação dos movimentos, os achados confirmam o potencial da visão computacional como um recurso viável para aprimorar a segurança e a aderência à ginástica laboral no ambiente home office.

Palavras-chave: DORT; Visão Computacional; Ginástica Laboral; Exercícios; Home Office.

ABSTRACT

The increasing adoption of remote work has led to a rise in the incidence of Work-Related Musculoskeletal Disorders (WMSDs). Although occupational gymnastics is a common solution, its effectiveness in the online format is severely compromised by the instructor's lack of individualized attention and the use of deactivated cameras by practitioners. To overcome this limitation, this work proposed and developed a platform prototype that uses computer vision to monitor and correct exercises in real-time, without the need for image storage. The platform aims to encourage the correct and regular practice of remote physical exercises. The experimental results were overwhelmingly positive, indicating that 100% of users felt more motivated by the use of the tool. The feedback and safety alert system was considered useful by 100% and 75% of participants, respectively, for the correct execution of movements. Although 75% reported perceiving a small lag and 50% noted occasional difficulties in motion capture, the findings confirm the potential of computer vision as a viable resource to enhance safety and adherence to occupational gymnastics in the home office environment.

Keywords: WMSD; Computer Vision; Occupational Gymnastics; Remote work.

SUMÁRIO

1	INTRODUÇÃO	7
1.1	CONTEXTUALIZAÇÃO	7
1.2	PROBLEMATIZAÇÃO	8
1.3	OBJETIVO	9
1.3.1	Objetivo Geral	9
1.3.2	Objetivos Específicos	9
1.4	JUSTIFICATIVA	10
1.5	ESTRUTURA DO DOCUMENTO	10
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	FUNDAMENTOS DA VISÃO COMPUTACIONAL	11
2.1.1	Detecção de Objetos e Estimativa de Poses	11
2.2	ANATOMIA E BIOMECÂNICA PARA CORREÇÃO DE EXERCÍCIOS	13
2.2.1	Conceitos fundamentais de biomecânica	13
2.2.2	A importância do feedback	14
2.3	TRABALHOS RELACIONADOS	15
3	DESENVOLVIMENTO	16
3.1	METODOLOGIA	16
3.1.1	Procedimentos e Método de Desenvolvimento	16
3.1.2	Requisitos do Projeto	16
3.1.2.1	Requisitos Funcionais	16
3.1.2.2	Requisitos Não Funcionais	17
3.2	CONFIGURAÇÃO DO AMBIENTE E FERRAMENTAS	17
3.3	ARQUITETURA GERAL DO SISTEMA	18
3.3.1	Camada de Persistência e Backend-as-a-Service (BaaS)	19
3.4	FRONTEND	20
3.4.1	Tecnologias e ferramentas Empregadas	20
3.4.2	Padrão Arquitetural MVP (Model-View-Presenter)	22
3.4.2.1	Conceito e Justificativa da Escolha	22
3.4.3	Camada Model (Modelo)	23
3.4.4	Camada View (Visão)	24
3.4.5	Camada Presenter (Apresentador)	25
3.4.6	Fluxo de Comunicação entre Camadas	25
3.5	BACKEND	25
3.5.1	Tecnologias e ferramentas empregadas	25

3.5.2	A escolha do modelo: YOLOvX	26
3.5.3	Abordagem para Detecção de Poses em Vídeos de Exercícios	27
3.5.4	Comparativo de modelos YOLO, MediaPipe, OpenCV e Torchvision	27
3.5.5	Ambiente de Teste e Métricas	28
3.5.6	Justificativa da escolha do modelo YoLo	28
3.5.7	Padrão de arquitetura em camadas	28
3.5.7.1	Camada de Apresentação (api/routers)	29
3.5.7.2	Camada de Negócio (api/services e core/)	31
3.5.7.3	Camada de Persistência (shared/repositories)	32
3.5.7.4	Camada de Conexão a Dados (shared/connectors)	33
3.5.7.5	Camada compartilhada (shared/ e utils/)	33
3.5.8	Análise de Movimento e Feedback em Tempo Real	33
3.5.8.1	Cálculo de Ângulos Articulares	33
3.5.8.2	Máquina de Estados para Detecção de Movimento	34
3.5.8.3	Ajuste Dinâmico de Dificuldade	34
3.5.8.4	Validação de Postura e Alinhamento Corporal	35
3.5.8.5	Sistema de Feedback Textual em Tempo Real	35
4	NAD.IA	37
4.1	VISÃO GERAL DA APLICAÇÃO	37
4.2	PRINCIPAIS TELAS	38
4.2.1	Tela de Login e Cadastro	38
4.2.2	Recursos Compartilhados	38
4.2.2.1	Menu Lateral (Sidebar)	38
4.2.2.2	Barra de Navegação (Header)	39
4.2.3	Tela Principal (Dashboard)	40
4.2.4	Tela de Configuração do Usuário	41
4.2.5	Execução	42
4.2.6	Tela de Visualização de Resultados	43
4.3	RESULTADOS OBTIDOS	45
4.3.1	Usabilidade e Funcionalidades Centrais	45
4.3.1.1	Facilidade de Ações Chave	46
4.3.2	Avaliação da Visão Computacional e Desempenho	46
4.3.3	Impacto e Perspectivas de Melhoria	47
4.3.3.1	Análise das Funcionalidades e Percepção de Valor	47
4.3.3.2	Discussão de Limitações e Recomendações	48
5	CONSIDERAÇÕES FINAIS	49
5.1	PRINCIPAIS APRENDIZADOS	49

5.2	LIMITAÇÕES TÉCNICAS	49
5.3	TRABALHOS FUTUROS	50
	REFERÊNCIAS	52
	APÊNDICE A – FORMULÁRIO DE TESTE DE USABILIDADE	55

1 INTRODUÇÃO

A crescente incidência de distúrbios osteomusculares configura-se como um dos principais desafios de saúde pública do século XXI, afetando milhões de pessoas globalmente e gerando custos significativos para sistemas de saúde e organizações. Além das repercussões clínicas, como dor crônica, redução da capacidade funcional e afastamentos laborais, esses distúrbios têm impacto direto na produtividade, no bem-estar psicossocial e na qualidade de vida dos trabalhadores. Esta problemática está diretamente associada às transformações nos paradigmas de trabalho contemporâneo, onde a predominância de atividades sedentárias, o uso intensivo de dispositivos digitais e a redução da atividade física se tornaram características marcantes da rotina moderna (WU; AL., 2023).

Tradicionalmente, a ginástica laboral supervisionada e outras intervenções fisioterapêuticas têm demonstrado eficácia comprovada na prevenção desses distúrbios, contribuindo para a melhoria da postura, o alívio da tensão muscular e a promoção de hábitos mais saudáveis no ambiente de trabalho (SILVA; CUNHA, 2017). Entretanto, sua implementação enfrenta limitações significativas relacionadas a custos operacionais, necessidade de profissionais especializados, logística complexa e, frequentemente, baixos índices de adesão por parte dos participantes. Essas barreiras são particularmente evidentes em contextos como o trabalho remoto, pequenas empresas ou situações onde o acesso a profissionais de saúde é limitado.

A democratização do acesso à prevenção de distúrbios osteomusculares demanda, portanto, soluções que transcendam as limitações tradicionais. Neste cenário, os avanços em Inteligência Artificial e Visão Computacional emergem como ferramentas promissoras para desenvolver sistemas autônomos de monitoramento e correção de exercícios. Tais tecnologias possibilitam a personalização das práticas de prevenção, o acompanhamento em tempo real e a redução da dependência de recursos humanos altamente especializados, ampliando o alcance de intervenções ergonômicas e preventivas para diferentes perfis de trabalhadores.

1.1 CONTEXTUALIZAÇÃO

Os Distúrbios Osteomusculares Relacionados ao Trabalho (DORT) e as Lesões por Esforços Repetitivos (LER) representam uma das principais causas de afastamento laboral e incapacidade funcional em diversos países (CARVALHO; LESSA, 2025). Segundo a Organização Mundial da Saúde (OMS), os distúrbios musculoesqueléticos representam uma das principais causas de incapacidade em todo o mundo, impactando a qualidade de vida e a produtividade dos trabalhadores (World Health Organization, 2022).

A ginástica laboral emergiu como uma das principais estratégias preventivas, demons-

trando eficácia na redução da incidência de DORT/LER, na melhoria da flexibilidade, no fortalecimento muscular e na promoção do bem-estar geral dos trabalhadores (SILVA; CUNHA, 2017). Contudo, a implementação efetiva dessa prática enfrenta desafios consideráveis.

A necessidade de profissionais especializados, custos de implementação, dificuldades logísticas para coordenar horários e espaços adequados, além da resistência cultural em algumas organizações, constituem barreiras recorrentes. Paralelamente, a expansão do trabalho remoto, especialmente acelerada após 2020, trouxe novos desafios. Trabalhadores em home office frequentemente enfrentam ambientes não ergonômicos, maior isolamento social e dificuldades para manter rotinas estruturadas de exercícios (WELLS et al., 2023). Observações empíricas em ambientes corporativos também revelam baixa participação em atividades síncronas online, como sessões virtuais de ginástica laboral, evidenciando a necessidade de soluções mais flexíveis e personalizadas.

A execução inadequada de exercícios não apenas anula os benefícios preventivos, mas pode resultar em novas lesões. A ausência de supervisão profissional torna crítica a necessidade de ferramentas que possam garantir a qualidade do movimento e fornecer orientações precisas em tempo real. Neste contexto, as tecnologias de estimação de pose humana baseadas em aprendizado de máquina têm demonstrado capacidade crescente de análise do movimento humano através de câmeras convencionais. Essas tecnologias oferecem uma oportunidade única para democratizar o acesso a orientações especializadas, permitindo análise biomecânica em tempo real sem a necessidade de equipamentos sofisticados ou supervisão presencial.

É importante ressaltar, entretanto, que tais sistemas não têm como propósito substituir o profissional de saúde. Pelo contrário, devem ser entendidos como ferramentas de apoio e complemento, ampliando o alcance das práticas preventivas e proporcionando maior autonomia aos trabalhadores em contextos onde a presença física de especialistas seja limitada ou inviável. A convergência desses fatores evidencia a necessidade de desenvolver soluções tecnológicas que possam fortalecer e expandir as estratégias tradicionais, tornando a prevenção de distúrbios osteomusculares mais acessível, eficaz e sustentável para um público amplo e diversificado.

1.2 PROBLEMATIZAÇÃO

A transição para ambientes de trabalho descentralizados trouxe benefícios de flexibilidade, mas também impôs barreiras significativas à implementação de programas de ginástica laboral supervisionados. Nesse cenário, o colaborador passa a ser responsável por sua própria prática, o que frequentemente resulta em desmotivação e baixa adesão a rotinas de exercícios. A ausência de orientação profissional em tempo real potencializa o risco de execução inadequada dos movimentos e reduz a efetividade das práticas

preventivas.

Diante disso, emergem questões centrais que orientam este estudo:

- Como auxiliar um colaborador em home office com exercícios de ginástica laboral e transformar sua experiência?
- De que maneira é possível automatizar a contagem de repetições e monitorar a frequência dos exercícios para acompanhar a evolução e a adesão do usuário ao programa?
- Como criar uma solução de baixo custo e alta acessibilidade, que não demande equipamentos especializados além de uma câmera para visualização dos movimentos?

Portanto, o problema central investigado neste projeto consiste na inexistência de ferramentas acessíveis capazes de oferecer supervisão, contagem automática e acompanhamento da prática de ginástica laboral em contextos de trabalho remoto, aliando tecnologia e prevenção em saúde ocupacional.

1.3 OBJETIVO

1.3.1 Objetivo Geral

Desenvolver uma plataforma funcional para monitorar métricas de exercícios de ginástica laboral, fornecendo ao usuário um acompanhamento de sua prática em tempo real.

1.3.2 Objetivos Específicos

Para alcançar o objetivo geral, resultados intermediários devem ser atingidos

- a) Realizar uma revisão bibliográfica sobre os temas de ginástica laboral, visão computacional, algoritmos de detecção de pose e desenvolvimento de APIs.
- b) Implementar o modelo de detecção de pose humana para extrair os pontos-chave (articulações) do corpo do usuário a partir de um *frame*¹ de vídeo.
- c) Desenvolver algoritmos para analisar a geometria e a cinemática dos pontos-chave, permitindo a identificação de padrões de movimento correspondentes a exercícios específicos.
- d) Construir uma API responsável por processar os dados de vídeo, realizar a contagem de repetições e registrar as métricas de desempenho.
- e) Criar uma interface de usuário intuitiva para que a pessoa possa selecionar uma lista de exercícios, visualizar sua execução na câmera e receber o feedback da contagem e do progresso.

¹ No contexto de processamento de vídeo, um *frame* corresponde a uma única imagem estática que compõe a sequência de um vídeo.

1.4 JUSTIFICATIVA

A relevância deste projeto se sustenta em três pilares principais: saúde e bem-estar, inovação tecnológica e demanda de mercado.

Do ponto de vista da saúde, a solução proposta tem o potencial de mitigar um problema de saúde pública crescente, prevenindo lesões ocupacionais e promovendo um estilo de vida mais ativo para milhares de trabalhadores (SILVA; CUNHA, 2017). Ao oferecer feedback corretivo, o sistema atua diretamente na prevenção de lesões causadas pela má execução dos movimentos.

No âmbito tecnológico e acadêmico, este trabalho contribui para o campo da Visão Computacional aplicada à saúde digital (e-health). A utilização do algoritmo para detecção de pose humana em tempo real, integrada a uma API para análise de padrões de movimento, representa uma aplicação prática ainda pouco explorada para resolver um problema concreto do cotidiano.

Finalmente, a demanda de mercado por soluções de bem-estar corporativo digital cresceu exponencialmente: estima-se que o mercado global de wellness corporativo deverá alcançar US\$ 120,23 bilhões até 2032, com CAGR² de 7,1% a partir de 2025 (NETWORK, 2025). Empresas buscam formas escaláveis e eficientes de cuidar da saúde de suas equipes remotas. Uma plataforma como a proposta oferece um diferencial competitivo, promovendo a saúde do time e potencialmente reduzindo custos com afastamentos e tratamentos médicos.

1.5 ESTRUTURA DO DOCUMENTO

Em síntese, este capítulo apresentou a contextualização do tema, o problema de pesquisa, bem como a justificativa, o escopo e os objetivos que orientam o desenvolvimento deste estudo. Essas etapas forneceram a base conceitual necessária para compreender a relevância da investigação e delimitar o caminho metodológico adotado.

Os capítulos seguintes apresentarão a Fundamentação Teórica, que abordará os principais conceitos, estudos e referenciais que sustentam e justificam a proposta deste trabalho. Em seguida, o Desenvolvimento explicitará a arquitetura, a comunicação entre as camadas da aplicação e a justificativa das tecnologias empregadas. Os resultados obtidos serão apresentados no capítulo "Nad.IA", que detalha tanto a plataforma quanto os resultados dos testes com usuários. Por fim, as Considerações Finais trarão a conclusão do trabalho e as sugestões para trabalhos futuros.

² CAGR (Compound Annual Growth Rate) significa Taxa de Crescimento Anual Composta, usada para medir a taxa de crescimento médio de um investimento ou mercado em um período de tempo.

2 FUNDAMENTAÇÃO TEÓRICA

O presente trabalho, ao desenvolver uma ferramenta web e mobile para auxiliar os usuários na realização de exercícios físicos, encontra seu principal alicerce teórico na pesquisa de Adolf(2024) (ADOLF et al., 2025). O estudo de Adolf e colaboradores, embora trate de uma ferramenta semelhante, possui um foco e objetivo distintos, o que o torna um ponto de partida crucial para a presente análise.

A fim de fornecer a base conceitual para o desenvolvimento da ferramenta, este capítulo abordará os fundamentos da visão computacional e suas principais tarefas. Também serão detalhados os modelos testados, a justificativa para a escolha do modelo final e suas aplicações na área da saúde. Por fim, o capítulo contextualiza a pesquisa com trabalhos correlatos e discute a biomecânica dos exercícios propostos, fornecendo a sustentação científica necessária para a aplicação da ferramenta.

2.1 FUNDAMENTOS DA VISÃO COMPUTACIONAL

A Visão Computacional é uma área interdisciplinar da Inteligência Artificial que se dedica a capacitar computadores a “ver”, processar e interpretar o mundo visual. Ela foca no desenvolvimento de modelos e algoritmos que permitem a extração de informações significativas a partir de dados visuais, como vídeos e imagens digitais. Seus objetivos incluem não apenas o reconhecimento de objetos, mas também a compreensão de cenas complexas, a reconstrução 3D de ambientes e a análise de movimento.

A importância da Visão Computacional reside em sua vasta gama de aplicações, que vão desde a segurança (reconhecimento facial e monitoramento), saúde (análise de exames de imagem), indústria (controle de qualidade e automação) até a automação veicular. Para o contexto deste trabalho, o foco principal é a Estimativa de Pose (*Pose Estimation*), uma tarefa fundamental que consiste em identificar e rastrear os pontos-chave (keypoints) do corpo humano. Esta técnica é crucial para o monitoramento e análise de exercícios em tempo real, permitindo a avaliação da forma, alinhamento e movimento corporal.

2.1.1 Detecção de Objetos e Estimativa de Poses

A Visão Computacional, devido à sua amplitude, é subdividida em tarefas que abordam diferentes objetivos de análise de imagem. Embora existam diversas delas, as mais relevantes para o contexto deste trabalho são a Detecção de Objetos e, mais especificamente, a Estimativa de Poses. Para contextualizar a importância da Estimativa de Poses, é fundamental apresentar, de forma geral, outras funcionalidades, como:

1. **Classificação de Imagens (*Image Classification*)**: Esta é a tarefa mais básica.

O objetivo é atribuir um único rótulo a uma imagem inteira, identificando o que ela contém. Por exemplo, um modelo de classificação pode determinar se uma imagem é de um “cachorro” ou de um “gato”.



Figura 1 – Exemplo de detecção de objetos com YOLO (Fonte: (ULTRALYTICS, 2025a)).

2. **Segmentação de imagem (*Image Segmentation*)**: A segmentação visa classificar cada pixel de uma imagem. Ela cria uma “máscara” que delimita as fronteiras de cada objeto ou região de interesse. Existem dois tipos principais:

- **Segmentação Semântica**: Agrupa pixels de uma mesma categoria (ex: todos os pixels de “árvore” em uma imagem). Conferir imagem 2.
- **Segmentação de Instância**: Identifica e separa instâncias individuais de objetos (ex: diferencia a “árvore 1” da “árvore 2”). Conferir imagem 3



Figura 2 – Exemplo de segmentação semântica (Fonte: (ZHANG et al., 2025)).

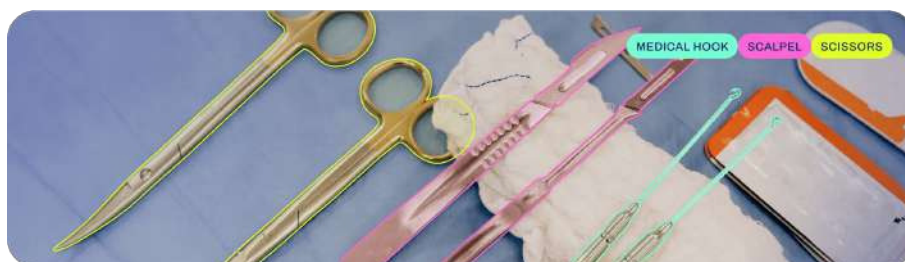


Figura 3 – Exemplo de segmentação de instância com YOLO (Fonte: (ULTRALYTICS, 2025a)).

3. **Detecção de Objetos (*Object Detection*)**: Esta tarefa vai além da classificação, pois não só identifica quais objetos estão presentes em uma imagem, mas também localiza cada um deles, desenhando uma “caixa delimitadora” (*bounding box*) ao redor. É uma etapa fundamental, frequentemente usada como base para tarefas mais complexas. Por exemplo, um sistema de detecção de objetos pode identificar a localização de cada pessoa em uma foto. No Nad.IA é de extrema utilidade, é como o algoritmo o usuário durante a atividade.



Figura 4 – Exemplo de detecção de objetos com YOLO (Fonte: (ULTRALYTICS, 2025a)).

4. **Estimativa de Poses (*Pose Estimation*)**: O foco deste trabalho, a estimativa de poses, é uma tarefa especializada que identifica a localização de pontos-chave (*keypoints*) do corpo humano ou de outros objetos em uma imagem ou vídeo. Esses pontos-chave — como cotovelos, joelhos e ombros — são usados para criar um esqueleto digital (*skeleton*) que representa a postura e o movimento. A estimativa de poses é crucial para aplicações que exigem a análise detalhada do movimento corporal, como a sua ferramenta de monitoramento de exercícios.

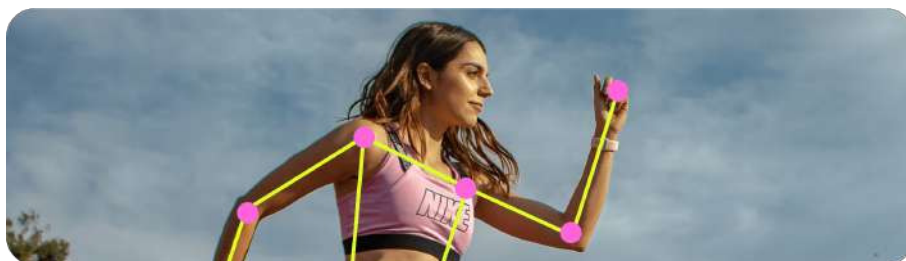


Figura 5 – Exemplo de estimativa de pose com YOLO (Fonte: (ULTRALYTICS, 2025a)).

2.2 ANATOMIA E BIOMECÂNICA PARA CORREÇÃO DE EXERCÍCIOS

2.2.1 Conceitos fundamentais de biomecânica

A compreensão dos movimentos do corpo humano é essencial para o desenvolvimento de uma ferramenta eficaz de monitoramento de exercícios. Para tanto, este trabalho se apoia em três conceitos fundamentais da biomecânica: articulação, ângulo de movimento e alinhamento corporal (LACOMBE, 2020; GUISELINI, 2025; ORTOPEDIA, 2023). São

esses os parâmetros que a ferramenta Nad.IA utilizará para avaliar a execução dos exercícios e fornecer feedback ao usuário.

- **Articulação:** No contexto deste estudo, a articulação é o ponto de conexão entre dois segmentos do corpo, como o joelho ou o cotovelo. O algoritmo de visão computacional detecta esses pontos de referência, permitindo que o sistema acompanhe a posição de cada membro durante o exercício. A precisão na detecção das articulações é crucial, pois elas são a base para o cálculo dos ângulos de movimento.
- **Ângulo de Movimento:** O ângulo de movimento é a medida, em graus, da amplitude de flexão ou extensão de uma articulação.
- **Alinhamento Corporal:** O alinhamento corporal refere-se à correta posição de todo o corpo em relação a uma referência, como a coluna vertebral ou o chão.

Na figura a baixo é possível conferir

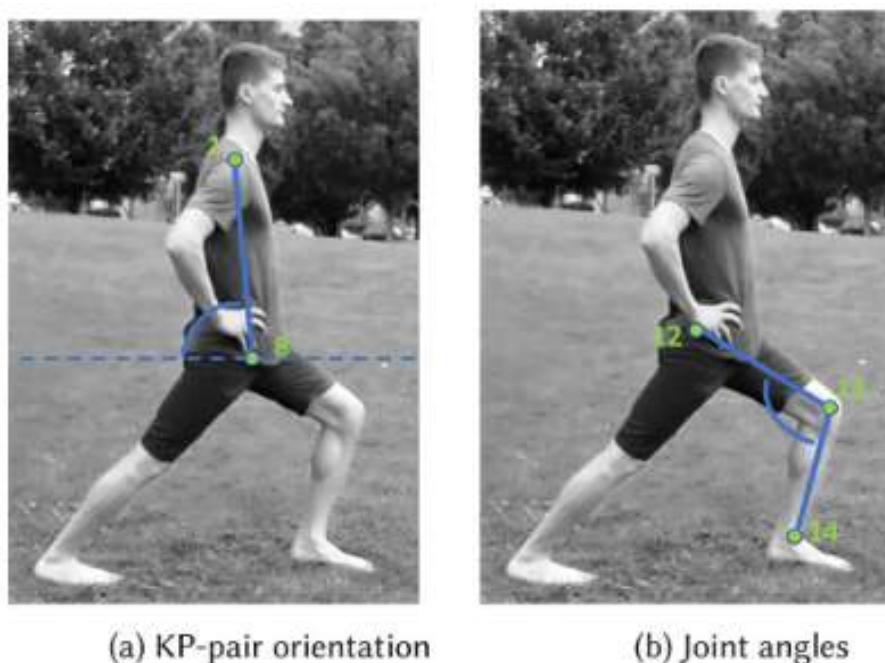


Figura 6 – Imagem que mostra o alinhamento, ângulo e articulação (Fonte: (ADOLF et al., 2025)).

2.2.2 A importância do feedback

Com base nos achados de Adolf et al. (ADOLF et al., 2025), o feedback revelou-se um conforto adicional para os usuários do sistema, e não um elemento indispensável ou criticamente necessário para a sua utilização. O estudo demonstrou que, estatisticamente, o feedback em vídeo não apresentou uma melhoria significativa na execução dos movimentos. Embora os dados não mostrem uma diferença substancial em comparação com a

ausência de feedback, o estudo destaca que os usuários se sentiram mais confortáveis ao realizarem os exercícios com algum tipo de orientação.

2.3 TRABALHOS RELACIONADOS

A visão computacional tem sido amplamente estudada e aplicada na área da saúde por um longo período. Um exemplo é o artigo de Sousa(2008) (SOUSA, 2008), que já em 2008 utilizou a tecnologia para a análise da marcha humana ¹.

Apesar de a aplicação na saúde ser um tema consolidado, o foco desta pesquisa é mais específico: a aplicação da visão computacional no âmbito da ginástica laboral e no monitoramento de exercícios físicos. Para embasar este trabalho, dois estudos correlatos foram de especial relevância para a nossa metodologia.

O “Offistretch”, conforme apresentado por Adolf et. al.(2024)(ADOLF et al., 2025), é uma ferramenta que emprega a visão computacional para auxiliar em alongamentos domiciliares. O sistema utiliza um modelo pré-treinado para comparar as poses do usuário com as poses de referência em tempo real, baseando-se em uma análise geométrica e espacial, além de fornecer feedback imediato.

De forma análoga, o Nad.IA foi concebido com o propósito de aplicar essa tecnologia no contexto da ginástica laboral. A principal distinção metodológica reside no fato de o Nad.IA prescindir do fine-tuning do modelo de detecção de pose. Em vez disso, a ferramenta realiza uma comparação geométrica estática entre os pontos de referência do modelo e a pose do usuário, uma abordagem que se mostra eficiente para o monitoramento de exercícios que não demandam uma análise espacial complexa. Essa escolha metodológica otimiza o processamento e torna a solução mais acessível, sem comprometer a precisão necessária para o seu propósito.

Como demonstrado por Silva(2017)(SILVA; CUNHA, 2017) a prática da ginástica laboral não só contribui para a melhora da saúde e do bem-estar dos colaboradores, mas também traz benefícios tangíveis para as organizações, como o aumento da assiduidade e a redução de acidentes de trabalho.

A visão apresentada por esses autores, que integra aspectos de saúde e mercado, é de extrema relevância para a presente pesquisa. Ela fornece a justificativa central para o desenvolvimento de uma ferramenta que promove e facilita a ginástica laboral, validando a importância e o impacto do trabalho que está sendo proposto.

¹ A marcha humana é o estudo do caminhar e do correr de uma pessoa, geralmente em contextos clínicos e esportivos.

3 DESENVOLVIMENTO

O desenvolvimento abrange desde a configuração do ambiente até a implementação das funcionalidades centrais de análise de exercícios. Conforme estabelecido nos objetivos específicos, o foco foi a utilização de um modelo pré-treinado para detecção de pose, o desenvolvimento de algoritmos para análise de movimento e a criação de uma API para orquestrar a comunicação entre o cliente e o servidor.

A fim de, melhor apresentar a arquitetura e construção desse trabalho, o capítulo está estruturado da seguinte forma: primeiramente, será apresentado a arquitetura geral do sistema; seguido pelo desenvolvimento do *frontend*, a interface com o usuário; posteriormente, descreve-se o detalhamento e implementação do *backend*, responsável pelo processamento de vídeo e pela lógica de negócio; e, por fim, são explicados os algoritmos específicos criados para a análise dos exercícios.

3.1 METODOLOGIA

Esta pesquisa caracteriza-se como de natureza aplicada, com abordagem qualitativa e quantitativa, focada no desenvolvimento e avaliação de um produto. O delineamento do trabalho compreende o levantamento de requisitos, o desenvolvimento técnico utilizando Visão Computacional e a avaliação de usabilidade com usuários finais.

3.1.1 Procedimentos e Método de Desenvolvimento

Inicialmente, adotou-se a metodologia ágil *Scrum*. Entretanto, devido às limitações de recursos humanos (time reduzido) e restrição de tempo, a estratégia foi adaptada para um modelo iterativo simplificado. Esta escolha permitiu focar na entrega incremental de funcionalidades essenciais (*backlog* prioritário), garantindo a viabilidade do protótipo dentro do cronograma acadêmico.

O fluxo de desenvolvimento seguiu as etapas de levantamento de requisitos (descritos nas subseções 3.1.2.1 e 3.1.2.2), desenvolvimento do sistema de avaliação dos exercícios e da interface gráfica.

3.1.2 Requisitos do Projeto

Abaixo, detalham-se os requisitos que nortearam o desenvolvimento do sistema Nad.IA.

3.1.2.1 Requisitos Funcionais

- **Captura de imagem/vídeo em tempo real:** O sistema deve ser capaz de captar imagens do usuário em tempo real durante a execução dos exercícios.

- **Detecção de pose corporal:** O sistema deve utilizar técnicas de *pose estimation* para identificar as articulações e a postura do corpo do usuário.
- **Avaliação da postura:** O sistema deve avaliar a execução do alongamento como correta ou incorreta com base em parâmetros pré-definidos.
- **Feedback automático:** O sistema deve fornecer *feedback* auditivo via TTS (*text-to-speech*) e visual para orientar o usuário.
- **Armazenamento de resultados:** O sistema deve armazenar os resultados das sessões para permitir o acompanhamento do desempenho.
- **Interface de seleção:** O usuário deve conseguir escolher a lista de exercícios que deseja executar.
- **Acompanhamento em tempo real:** O sistema deve exibir, durante a execução, se o movimento está correto.

3.1.2.2 Requisitos Não Funcionais

- **Tempo de resposta do sistema:** O *feedback* deve ser fornecido em no máximo 1 segundo após a execução do movimento.
- **Portabilidade e custo acessível:** O sistema deve ser compatível com *hardware* de baixo custo, como um *notebook* comum.
- **Usabilidade para leigos:** A interface deve ser intuitiva para pessoas sem conhecimento técnico prévio.
- **Compatibilidade com ambientes domésticos:** O sistema deve funcionar em ambientes com iluminação e espaço típicos de uma residência.

3.2 CONFIGURAÇÃO DO AMBIENTE E FERRAMENTAS

A primeira etapa para o desenvolvimento do Nad.IA consistiu na preparação do ambiente de trabalho e na seleção das bibliotecas e tecnologias essenciais. A escolha recaiu sobre ferramentas que oferecem robustez, ampla documentação e suporte da comunidade, alinhadas aos requisitos do projeto.

O emprego de LLMs no seguinte trabalho ocorreu para revisão textual e solução de dúvidas e problemas durante o desenvolvimento do sistema.

Tabela 1 – Ferramentas e tecnologias utilizadas no desenvolvimento

Categoria	Ferramenta/Tecnologia	Versão
IDE	Visual Studio Code	1.85
Linguagem	Python	3.10
Visão Computacional	YOLOv8 (Ultralytics)	8.0.196
Framework Backend	FastAPI	0.104
Framework Frontend	Vue.js	3.3
Controle de Versão	Git/GitHub	-
LLMs	Gemini e Claude	2.5-flash e Sonnet 4.1

3.3 ARQUITETURA GERAL DO SISTEMA

A arquitetura do sistema Nad.IA foi concebida no modelo cliente-servidor, com o objetivo de proporcionar uma análise de movimentos em tempo real e com alta responsividade na interface do usuário. A separação de responsabilidades entre o frontend (cliente) e o backend (servidor) é fundamental para a performance da aplicação, conforme ilustrado na Figura 7.

O processo é iniciado pelo usuário ao interagir com a interface da aplicação, desenvolvida em Vue.js. Nesse momento, o fluxo se divide em duas vias paralelas: a visualização da câmera e o processamento dos dados. Primeiramente, o frontend solicita acesso à câmera do dispositivo e inicia a captura do stream de vídeo, que é imediatamente renderizado na tela para oferecer um retorno visual direto, como um espelho. Para garantir que a interface permaneça fluida e não sofra com travamentos, a tarefa de preparar e enviar os quadros para análise é delegada a um Web Worker. Essa tecnologia, que executa scripts em segundo plano, é crucial para a experiência do usuário.

O Web Worker estabelece uma conexão persistente e bidirecional com o servidor por meio de um WebSocket, enviando os quadros de vídeo em tempo real para o backend. Embora o protocolo WebRTC seja amplamente utilizado para transmissão de vídeo em tempo real por basear-se em UDP e oferecer baixa latência, a arquitetura deste trabalho optou pelo uso de WebSockets (sobre o protocolo TCP). Essa escolha justifica-se por dois pilares fundamentais:

Confiabilidade e Integridade dos Dados: Diferente do UDP, que tolera perdas de pacotes em favor da velocidade, o TCP garante a entrega ordenada e completa de todos os frames. Para a análise de movimento proposta, a fidelidade dos dados é um requisito crítico, onde a perda de pacotes poderia comprometer a precisão dos algoritmos de detecção.

Viabilidade de Implementação: O WebRTC introduz uma complexidade significativamente maior, exigindo mecanismos de sinalização e infraestrutura para contornar NAT (STUN/TURN). Dado o cronograma do projeto, o WebSocket apresentou-se como uma solução mais ágil e eficiente para estabelecer o canal de comunicação.

Embora o protocolo WebRTC seja amplamente utilizado para transmissão de vídeo em tempo real por basear-se em UDP e oferecer baixa latência, a arquitetura deste trabalho optou pelo uso de WebSockets (sobre o protocolo TCP). Essa escolha justifica-se por dois pilares fundamentais:

1. **Confiabilidade e Integridade dos Dados:** Diferente do UDP, que tolera perdas de pacotes em favor da velocidade, o TCP garante a entrega ordenada e completa de todos os frames. Para a análise de movimento proposta, a fidelidade dos dados é um requisito crítico, onde a perda de pacotes poderia comprometer a precisão dos algoritmos de detecção.
2. **Viabilidade de Implementação:** O WebRTC introduz uma complexidade significativamente maior, exigindo mecanismos de sinalização e infraestrutura para contornar NAT (STUN/TURN). Dado o cronograma do projeto, o WebSocket apresentou-se como uma solução mais ágil e eficiente para estabelecer o canal de comunicação.

No servidor, esses quadros são submetidos ao modelo de visão computacional YOLO, que realiza a estimativa de pose. Assim que o modelo processa um quadro e extrai os pontos-chave (keypoints) do esqueleto, ele gera um feedback estruturado. Esses dados são enviados de volta ao frontend pela mesma conexão WebSocket. Finalmente, o frontend recebe o feedback e o utiliza para renderizar a visualização final, sobrepondo o esqueleto digital ao stream de vídeo original. O resultado é uma visualização aumentada onde o usuário pode observar seu movimento junto com a análise gerada pelo sistema, concluindo o ciclo de interação em tempo real.

Esta arquitetura garante uma clara separação de interesses: o *frontend* é responsável pela captura e renderização, enquanto o *backend* se concentra no processamento computacionalmente intensivo, com a comunicação entre eles otimizada para baixa latência através do uso de Web Workers e WebSockets.

3.3.1 Camada de Persistência e Backend-as-a-Service (BaaS)

A camada de persistência do Nad.IA é projetada para ser híbrida, aproveitando o MongoDB e o Redis para a lógica de monitoramento em tempo real, e utilizando o Supabase, uma solução Backend-as-a-Service (BaaS)¹, para o gerenciamento robusto e escalável de usuários e dados analíticos.

O Supabase é empregado para gerenciar a Autenticação, os Perfis de Usuário e armazenar as Métricas e Atividades históricas. As tabelas no Supabase (PostgreSQL) são centrais para a experiência do usuário e para o registro de seu progresso ao longo do tempo. Elas se dividem em quatro entidades principais:

¹ Modelo de computação em nuvem que fornece APIs e SDKs prontos para conectar o frontend de uma aplicação a serviços de backend comuns, como autenticação e banco de dados

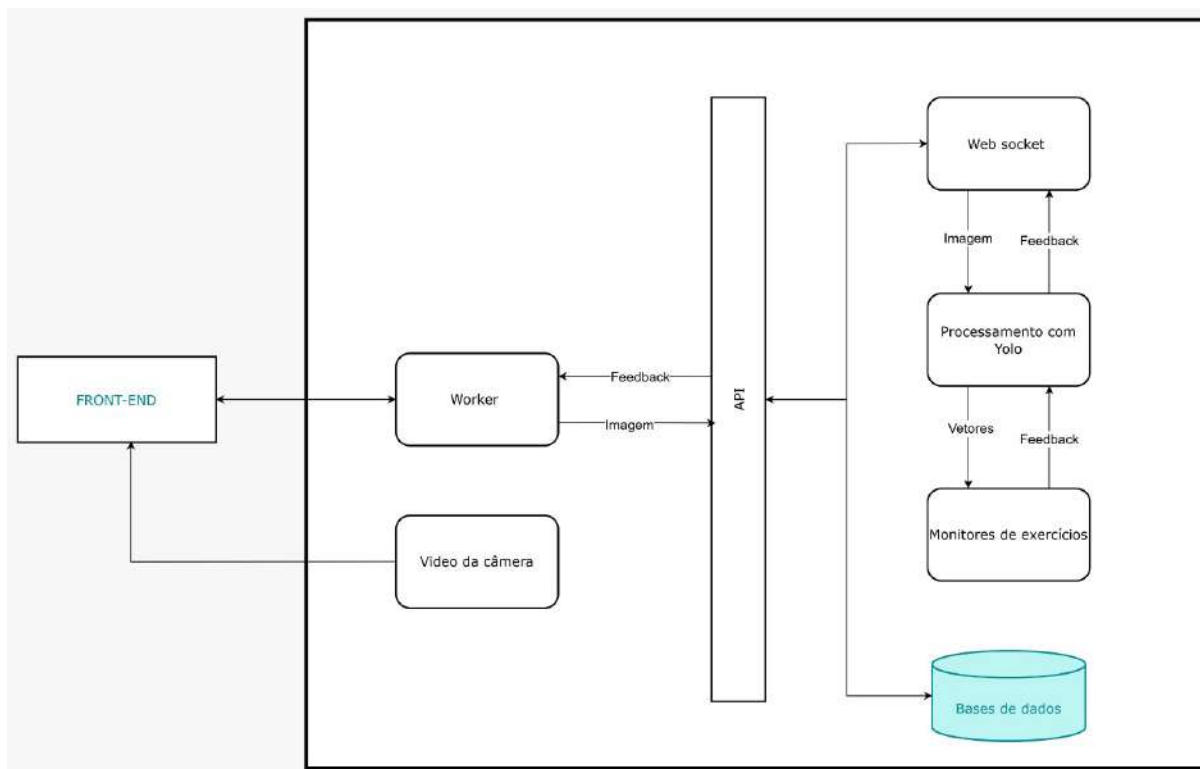


Figura 7 – Fluxo de processamento do sistema. Fonte: Elaboração própria.

3.4 FRONTEND

Nesta secção será apresentado a arquitetura e as tecnologias empregadas no desenvolvimento do frontend da aplicação, que consiste em uma aplicação responsável pela interface de interação do usuário com o sistema de análise de exercícios físicos. A aplicação web foi desenvolvida com foco em responsividade, performance e manutenibilidade, utilizando tecnologias modernas do ecossistema JavaScript/TypeScript. O frontend é responsável por capturar vídeo em tempo real através da câmera do dispositivo, enviar os frames para processamento no backend, receber e exibir os resultados da análise de postura, além de gerenciar todo o fluxo de autenticação e navegação do usuário pela aplicação.

3.4.1 Tecnologias e ferramentas Empregadas

A seleção das tecnologias considerou tanto os requisitos técnicos do projeto quanto a produtividade no desenvolvimento. A escolha pelo ecossistema JavaScript/TypeScript e, em particular, pelo framework Vue.js, foi fortemente influenciada pela afinidade prévia do autor com a tecnologia e sua reconhecida curva de aprendizado acessível. Essa familiaridade, aliada à robustez do framework, foi crucial para construir uma aplicação capaz de lidar com os requisitos de tempo real de forma eficiente. As seguintes ferramentas principais compõem a stack:

- **Vue.js 3 com Composition API:** Escolhido como o framework principal devido

Tabela 2 – Estrutura de Dados do Supabase para Persistência Híbrida

Tabela	Função Principal	Campos Chave	Relação com a Aplicação
users (auth.users)	Gerenciamento de Autenticação e identidade básica.	id (UUID - Chave Primária), email.	Usada pelo serviço de autenticação do Supabase.
profiles	Armazenamento de Dados Pessoais e parâmetros do usuário.	id (UUID - Ref. auth.users), display_name, first_name, last_name.	Preenche a Tela de Configuração do Usuário (4.2.4).
user_metrics	Registro Diário de métricas de saúde e desempenho.	user_id, date (Composta Única).	Alimenta os <i>cards</i> de calorias, minutos de exercício, água e peso no <i>Dashboard</i> (4.2.6).
user_activities	Registro de Atividades Físicas realizadas.	user_id, activity_date.	Armazena histórico de treinos e alimenta o gráfico "Atividades por Categoria".

à sua curva de aprendizado suave, reatividade otimizada e ecossistema maduro. A Composition API, em particular, permite uma organização mais lógica e reutilizável do código, especialmente em componentes complexos como a tela de análise de exercícios.

- **TypeScript**: Adotado para adicionar tipagem estática ao JavaScript. O uso do TypeScript aumenta a robustez do código, melhora a manutenibilidade e permite que erros sejam detectados em tempo de desenvolvimento, o que é fundamental em um projeto de maior escala.
- **Vite**: Utilizado como build tool do projeto. Vite oferece um servidor de desenvolvimento extremamente rápido com Hot Module Replacement ² instantâneo e um processo de build otimizado para produção, melhorando significativamente a experiência de desenvolvimento.
- **Axios**: Empregado como cliente HTTP para realizar a comunicação com a API do backend. Sua API baseada em Promises facilita o tratamento de requisições assíncronas, interceptadores e tratamento de erros de forma centralizada.

² Hot Module Replacement (HMR), ou Substituição de Módulos a Quente, é uma funcionalidade de ferramentas de build (como o Vite) que permite atualizar, adicionar ou remover módulos de código (como componentes ou folhas de estilo) em uma aplicação em execução, sem a necessidade de recarregar a página inteira. Sua principal vantagem é a preservação do estado da aplicação, o que acelera drasticamente o ciclo de desenvolvimento.

- **Pinia:** Foi utilizado para gerenciar o estado global, como informações do usuário autenticado e configurações da aplicação
- **Progressive Web App (PWA):** Implementado para oferecer a capacidade de instalação da aplicação em dispositivos móveis. Isso permite que o usuário adicione o site à tela inicial e o utilize como um aplicativo nativo, melhorando a acessibilidade e a experiência de uso no celular.
- **OAuth2.0:** Para garantir a confiabilidade, o gerenciamento seguro de sessões e a proteção dos dados do usuário, o sistema utiliza o protocolo de autenticação OAuth 2.0.

3.4.2 Padrão Arquitetural MVP (Model-View-Presenter)

3.4.2.1 Conceito e Justificativa da Escolha

Para a estruturação do frontend, foi adotado o padrão de arquitetura Model-View-Presenter (MVP). Esta decisão arquitetural visa estabelecer uma clara separação de responsabilidades entre as diferentes camadas da aplicação, facilitando a manutenção, a testabilidade e a escalabilidade do código ao longo do ciclo de vida do projeto.

O padrão MVP foi escolhido em detrimento de outras arquiteturas, como MVC (Model-View-Controller) ou MVVM (Model-View-ViewModel), por proporcionar uma separação mais nítida de responsabilidades. Nesse modelo, a View é completamente passiva e não contém lógica de aplicação, apenas delegando eventos para o Presenter, o que torna o código mais previsível e de fácil compreensão. Como resultado, a manutenção é simplificada, pois as alterações na lógica de negócio não afetam a camada de apresentação e vice-versa, reduzindo o acoplamento entre os componentes. Além disso, o sistema de reatividade do Vue.js se integra naturalmente com o padrão MVP, permitindo que o Presenter gerencie estados reativos que a View observa e atualiza automaticamente.

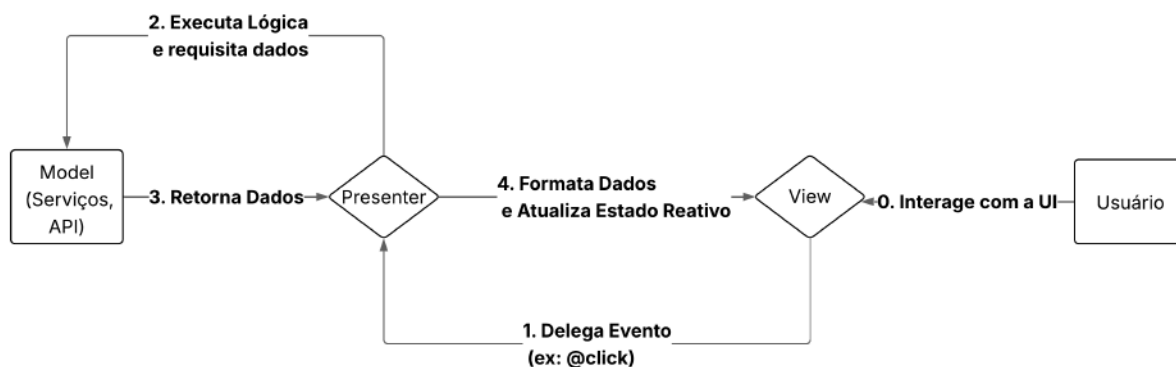


Figura 8 – Fluxo aquitetural frontend. Fonte: Elaboração própria.

3.4.3 Camada Model (Modelo)

A camada Model representa a estrutura de dados e encapsula a lógica de negócios da aplicação, constituindo um dos pilares fundamentais da arquitetura MVP (Model-View-Presenter). No contexto deste projeto, essa camada desempenha um papel central na manutenção da integridade dos dados e na implementação das regras de negócio, operando de forma completamente independente das camadas de apresentação e interface. Esta separação arquitetural assegura que a View permaneça agnóstica em relação à lógica de aplicação, concentrando-se exclusivamente na renderização e interação com o usuário.

A primeira responsabilidade da camada Model consiste na definição de interfaces TypeScript que representam as entidades do domínio da aplicação, tais como `ExerciseModel.ts`, `UserModel.ts` e `StatsModel.ts`. Estas interfaces funcionam como contratos de dados fortemente tipados, estabelecendo não apenas a estrutura esperada para cada entidade, mas também proporcionando verificação de tipos em tempo de compilação. Esta abordagem facilita a manutenção e refatoração do código.

Além disso, a camada Model encapsula toda a comunicação com a API backend através de serviços HTTP especializados. Este encapsulamento promove o princípio da separação de responsabilidades (Separation of Concerns), isolando a lógica de comunicação de rede em componentes dedicados e permitindo que o Presenter interaja com os dados através de métodos, sem necessidade de conhecimento sobre os detalhes técnicos das requisições HTTP. Os serviços atuam como uma camada de abstração que mascaram aspectos como endpoints, métodos HTTP, headers de autenticação e tratamento de respostas e erros, criando uma interface limpa, intuitiva. Esta arquitetura não apenas simplifica significativamente o código do Presenter, reduzindo sua complexidade de leitura do código, mas também facilita a realização de testes unitários isolados e a eventual substituição ou modificação da API backend sem impacto nas demais camadas do sistema.

No contexto da comunicação com o servidor, a camada Model também assume a responsabilidade crítica de validar dados antes de enviá-los ao backend. Esta validação no lado do cliente desempenha múltiplas funções: além de proporcionar feedback imediato ao usuário, ela permite classificar a urgência e a severidade de cada mensagem retornada pela API, facilitando a construção de experiências de usuário mais informativas e contextualizadas. Através desta validação prévia, o usuário consegue compreender com clareza e precisão quais ajustes devem ser realizados em cada momento específico da interação, reduzindo frustrações e melhorando a usabilidade geral do sistema.

A etapa final e possivelmente mais crítica do fluxo de dados antes de sua apresentação ao usuário ocorre na função de transformação e mapeamento, cuja implementação central no projeto é exemplificada pela classe `ExerciseModel.ts`. Esta camada atua como uma fronteira de tradução essencial, recebendo dados brutos geralmente em formato JSON vindos do `exerciseService.ts` e convertendo-os em uma estrutura de dados rica, tipada

e otimizada para o consumo interno da aplicação. Por exemplo, APIs externas frequentemente possuem contratos de dados que não são ideais para a lógica de apresentação ou que seguem convenções de nomenclatura distintas. No sistema desenvolvido, a API retorna propriedades em `snake_case` (ex. `video_url`), que são mapeadas para `camelCase` (ex. `videoUrl`) no `ExerciseModel.ts`, mantendo às convenções padrão do TypeScript e JavaScript e garantindo consistência em todo o código da aplicação.

A camada Model ajuda na clareza dos dados recebidos. Uma funcionalidade implementada é a resolução de caminhos para recursos estáticos locais. Por exemplo, um identificador simples de vídeo, como "agachamento", é processado pela camada de Modelo para gerar o caminho completo e correto que o componente de vídeo na View pode carregar e renderizar. Sem essa centralização, cada componente individual precisaria conter e manter a lógica para montar essas URLs, gerando duplicação substancial de código e dificultando a manutenção evolutiva do sistema.

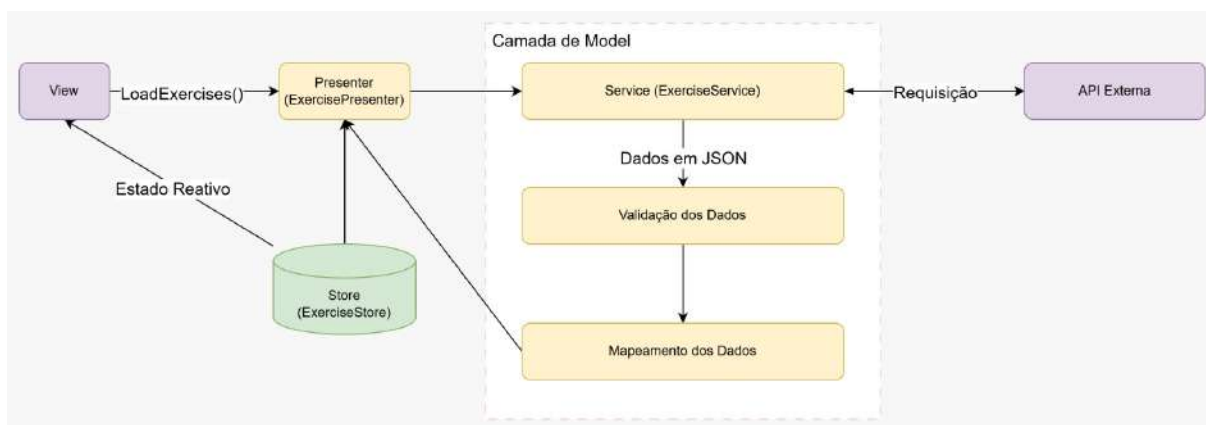


Figura 9 – Fluxo de processamento da model. Fonte: Elaboração própria.

3.4.4 Camada View (Visão)

A View é a camada de apresentação, composta pelos componentes da interface do usuário (UI). No contexto do Vue.js, as Views são implementadas como arquivos `.vue` que combinam template HTML, lógica de apresentação e estilos. No padrão MVP adotado, a View possui as seguintes características:

- **Passividade:** A View não contém lógica de aplicação ou de negócio. Sua única responsabilidade é renderizar dados e capturar eventos do usuário;
- **Delegação de Eventos:** Todas as interações do usuário (cliques, submissões de formulário, etc.) são imediatamente delegadas ao Presenter;
- **Observação de Estado:** A View observa propriedades reativas fornecidas pelo Presenter e se atualiza automaticamente quando esses dados mudam.

3.4.5 Camada Presenter (Apresentador)

O Presenter atua como intermediário entre o Model e a View, sendo responsável por toda a lógica de apresentação. Suas principais responsabilidades incluem:

- Receber eventos da View e processá-los;
- Utilizar o Model para acessar ou modificar dados quando necessário;
- Formatar dados para exibição na View;
- Gerenciar o estado da interface (loading, erros, dados carregados);
- Implementar regras de negócio relacionadas à apresentação.

3.4.6 Fluxo de Comunicação entre Camadas

A comunicação entre as camadas do padrão MVP segue um fluxo bem definido:

- Inicialização: A View instancia o Presenter, passando referências aos elementos DOM necessários (como `<video>` e `<canvas>`);
- Interação do Usuário: Quando o usuário clica no botão "Iniciar Exercício", a View invoca o método `presenter.startExercise()`;
- Processamento no Presenter: O Presenter processa o evento, utilizando o Model (através do `ExerciseService`) para enviar frames de vídeo ao backend;
- Atualização do Estado: Quando a resposta da API é recebida, o Presenter atualiza as propriedades reativas (`stats`, `isStreaming`);
- Renderização Automática: A View, observando essas propriedades reativas, automaticamente se re-renderiza com as novas informações, exibindo estatísticas atualizadas ao usuário.

Este fluxo garante que a lógica de aplicação permaneça no Presenter, enquanto a View se mantém como uma camada puramente de apresentação.

3.5 BACKEND

3.5.1 Tecnologias e ferramentas empregadas

- **Python:** Python foi a linguagem de programação escolhida para a construção do *backend* devido à sua integração nativa com o YOLO e por oferecer excelentes opções de *frameworks web*, como o FastAPI.

- **Pydantic:** Esta biblioteca Python é essencial para a validação e verificação de tipos, garantindo que as funções da aplicação recebam objetos no formato exato esperado. Sua utilização é dupla: na construção da API, define os *schemas* de requisição e resposta; e na camada de *repositories*, estabelece os modelos dos objetos a serem persistidos no MongoDB. Tal validação é crucial para melhorar a manutenibilidade e a legibilidade do código.
- **FastAPI:** FastAPI é um *framework web* de alto desempenho que implementa o padrão ASGI³. A utilização desta tecnologia foi essencial, pois permitiu o uso de *Web Sockets* de modo que as outras rotas (como a de *reset* e a de escolha do exercício) não fossem bloqueadas.
- **Web Socket:** *Web Socket* é um protocolo que permite um handshake de abertura⁴ no HTTP por meio de headers específicos, de forma que a conexão permaneça aberta mesmo após o recebimento da resposta. Como o fluxo de comunicação permanece aberto, o envio de *frames* e o recebimento de *feedback* ocorressem de maneira fluida.
- **Mongo:** Este é um banco do tipo NoSQL⁵ que possibilitou uma maior versatilidade no armazenamento de exercícios.
- **Redis:** O Redis também é um banco NoSQL; no entanto, é especializado em *caching*, ou seja, consegue performar persistências na escala de nanosegundos. Sua utilização no Nad.IA se deu para funcionar como um gerenciador de contexto, tendo em vista que o ID do exercício precisa ser consultado por duas rotas distintas em diferentes *routers*⁶
- **YOLO:** YOLO é o modelo de visão computacional utilizado para a leitura dos *frames* e a extração de *keypoints* da imagem. A escolha desta tecnologia deveu-se à sua alta performance e aos resultados excelentes, mesmo em contextos de pouco recurso computacional.

3.5.2 A escolha do modelo: YOLOvX

O desenvolvimento da ferramenta proposta exige uma solução de visão computacional eficiente para a detecção de pose humana. A escolha do modelo ideal foi baseada no *benchmarking*⁷ (POWERER, 2023), que compara quatro ferramentas. A partir dessa análise, ficou claro que o modelo YOLOvX era o mais apropriado, por duas razões principais.

³ Asynchronous Server Gateway Interface (ASGI) é um conjunto de especificações que permite múltiplas comunicações com a API de forma assíncrona.

⁴ Abertura do canal de comunicação

⁵ Significa que a estrutura do banco não é baseada em *schemas* rígidos e bem definidos, como ocorre em um banco de dados relacional, comumente SQL.

⁶ Estrutura que serve para construir as rotas da API de forma descentralizada, agrupando *endpoints* por domínio funcional, como, neste caso, o exercício e o monitoramento.

A primeira é a integridade dos dados, pois, entre os modelos comparados, o YOLO foi o único que conseguiu manter a rastreabilidade dos pontos-chave (os pipes) mesmo em movimentos rápidos e bruscos. Essa estabilidade é um fator crítico para a nossa aplicação.

O segundo motivo foi o custo-benefício. Apesar de o YOLO ter uma demanda de processamento maior, o seu resultado se mostrou superior, mesmo quando comparado a modelos mais leves como o MediaPipe. A relação entre a necessidade de processamento e a qualidade do resultado entregue justifica a escolha.

3.5.3 Abordagem para Detecção de Poses em Vídeos de Exercícios

A seleção de modelos pré-treinados foi uma etapa determinante no desenvolvimento deste trabalho, dada a limitação de poder computacional para o treinamento ou fine-tuning de uma nova rede. Optou-se por explorar ferramentas populares e bem documentados, como YOLO, OpenCV, MediaPipe e Torchvision, que oferecem ampla base de suporte e são amplamente utilizados na comunidade de Visão Computacional.

Tais modelos são geralmente treinados em grandes conjuntos de dados de propósito geral, como o COCO (Common Objects in Context) (LIN et al., 2015; ULTRALYTICS, 2025b), o que os torna versáteis para a detecção e análise de objetos em diversas situações. Apesar de não serem específicos para a análise de exercícios, a abordagem do presente trabalho não se baseia no treinamento de uma rede neural dedicada, como no trabalho de Mo et al. (2016)(MO et al., 2016), que se concentrou em desenvolver uma rede neural específica para avaliação do movimento humano.

Em vez disso, a metodologia aqui adotada se alinha à proposta de Adolf et al. (2024)(ADOLF et al., 2025) no projeto Offistretch, que utiliza modelos pré-treinados para a detecção de *keypoints*⁸ e, a partir deles, calcula e compara ângulos por meio de funções matemáticas para analisar a execução dos exercícios. Essa abordagem permite uma solução eficaz e menos dependente de um alto poder computacional, focando na precisão da análise geométrica do movimento.

3.5.4 Comparativo de modelos YOLO, MediaPipe, OpenCV e Torchvision

É fundamental, antes de prosseguir, compreender o que é “contexto” neste trabalho e por que é tão importante. Aqui, o contexto refere-se à capacidade de um modelo em manter uma visualização clara dos *keypoints* e *pipes*⁹ ao longo dos *frames*. Este parâmetro é crucial, tendo em vista que a abordagem é baseada no cálculo de ângulos entre vetores, tornando-se imperativo que esses artefatos sejam capturados e calculados com a maior

⁷ O termo *benchmark* refere-se a uma avaliação comparativa, que pode ser tanto entre diferentes tecnologias de mesmo propósito quanto baseada em uma métrica ou padrão de referência estabelecido.

⁸ *Keypoints* (ou pontos-chave) são marcadores no corpo: como cotovelos, ombros ou nariz, que os modelos de VC usam como base para analisar a pose humana. Eles funcionam como as "juntas" de uma marionete, ajudando o modelo a mapear a postura.

⁹ O termo *pipe* é utilizado para designar as arestas, ou barras, coloridas entre os *keypoints*

fidelidade possível. A perda desses dados compromete, diretamente, o desempenho e a confiabilidade do sistema.

Durante esta pesquisa, foram realizados testes empíricos para identificar os melhores modelos de detecção de movimento entre as opções: YOLO, MediaPipe, OpenCV e Torchvision. O principal critério de seleção foi a capacidade de manter o contexto, avaliando como cada biblioteca se comportava diante de diversos tipos de movimentos.

3.5.5 Ambiente de Teste e Métricas

Para assegurar a clareza e robustez da escolha, foi realizado um *benchmark* considerando as métricas mais relevantes para a aplicação: taxa de detecção, continuidade, confiança média, FPS (Frames por Segundo), tempo de inferência e *Joints* (ou *keypoints*). O ambiente de teste consistiu em um notebook com 12GB de RAM e uma CPU AMD Ryzen 5 3500U.

3.5.6 Justificativa da escolha do modelo YoLo

Os resultados detalhados na Tabela 3 demonstram que o YoLo oferece o **melhor custo-benefício** entre os modelos avaliados.

- **YoLo vs. OpenCV:** Embora o OpenCV seja superior em velocidade, sua confiança média de $\sim 30\%$ é inadequada para a aplicação. O YoLo atinge uma confiança significativamente superior ($\sim 80\%$), justificando seu tempo de inferência ligeiramente maior.
- **YoLo vs. MediaPipe:** O YoLo se destaca pela **taxa de detecção** mais elevada, tendo em vista que a do MediaPipe é de ($\sim 50\%$), um parâmetro fundamental para a confiabilidade do sistema proposto.
- **YoLo vs. Torchvision:** A superioridade se manifesta na rapidez de processamento. Apesar de o Torchvision apresentar uma confiança de $\sim 71\%$, seu tempo de inferência é de $\sim 6192,70ms$, enquanto o YoLo executa em apenas $\sim 1,26ms$.

A escolha do YoLo se consolida, portanto, por seu **equilíbrio ideal** entre alta precisão ($\sim 80\%$ de confiança) e velocidade de inferência, o que é crucial para a operação em tempo real do sistema.

3.5.7 Padrão de arquitetura em camadas

Para a construção do backend, foi adotada a arquitetura em camadas (Layered Architecture)²², onde cada camada possui responsabilidades bem definidas e se comunica preferencialmente com a camada imediatamente inferior, promovendo o isolamento de

Tabela 3 – Comparação de Performance entre Modelos de Detecção de Pose

Modelo	Taxa De- tecção (%)	Cont.	Conf. Média (%)	FPS	Tempo (ms)	Joints
MediaPipe	50.0	100.0	94.4	9.70	103.05	33.0
OpenCV	100.0	100.0	30.0	673.98	1.48	18.0
YOLO	100.0	100.0	80.9	7.93	126.18	17.0
Torchvision	100.0	100.0	71.00	0.16	6191.70	17.0

responsabilidades e a coesão do código. O sistema foi estruturado em quatro camadas principais e uma compartilhada:

1. **Camada de Apresentação (api/routers)**: Responsável pela comunicação via HTTP/WebSocket e recepção de frames;
2. **Camada de Negócio (api/services e core/)**: Encapsula a lógica de processamento de exercícios, integração com YOLO para extração de keypoints, cálculo vetorial de ângulos e detecção de fases de movimento através de máquina de estados;
3. **Camada de Acesso a Dados (shared/repositories)**: Abstrai as operações de persistência;
4. **Camada de Dados (shared/connectors)**: Responsável pela comunicação com MongoDB (armazenamento de exercícios) e Redis (cache do exercício ativo).
5. **Camada compartilhada (shared/)**: fornece recursos transversais, como modelos de dados (ODM com Bunnet), schemas de validação (Pydantic), enumerações de domínio e tratamento de exceções, garantindo reutilização e consistência entre as camadas do sistema.

3.5.7.1 Camada de Apresentação (api/routers)

A Camada de Apresentação (api/) é a interface de comunicação entre o frontend e as camadas inferiores do sistema. Esta camada foi implementada utilizando o framework FastAPI, combinando os paradigmas REST¹⁰ para operações CRUD¹¹ e WebSocket para comunicação bidirecional em tempo real, garantindo padronização e baixa latência, necessária para o processamento contínuo de frames de vídeo.

Essa camada é distribuída em routers, afim de melhorar a manuenabilidade, aplicando o princípio de injeção de dependências com as ferramentas do FastAPI para desacoplar a

¹⁰ Representational State Transfer é um estilo arquitetural que define um conjunto de regras para sistemas distribuídos.

¹¹ Acrônimo que significa: Create, Read, Update e Delete; Diz respeito às operações de persistência no banco de dados

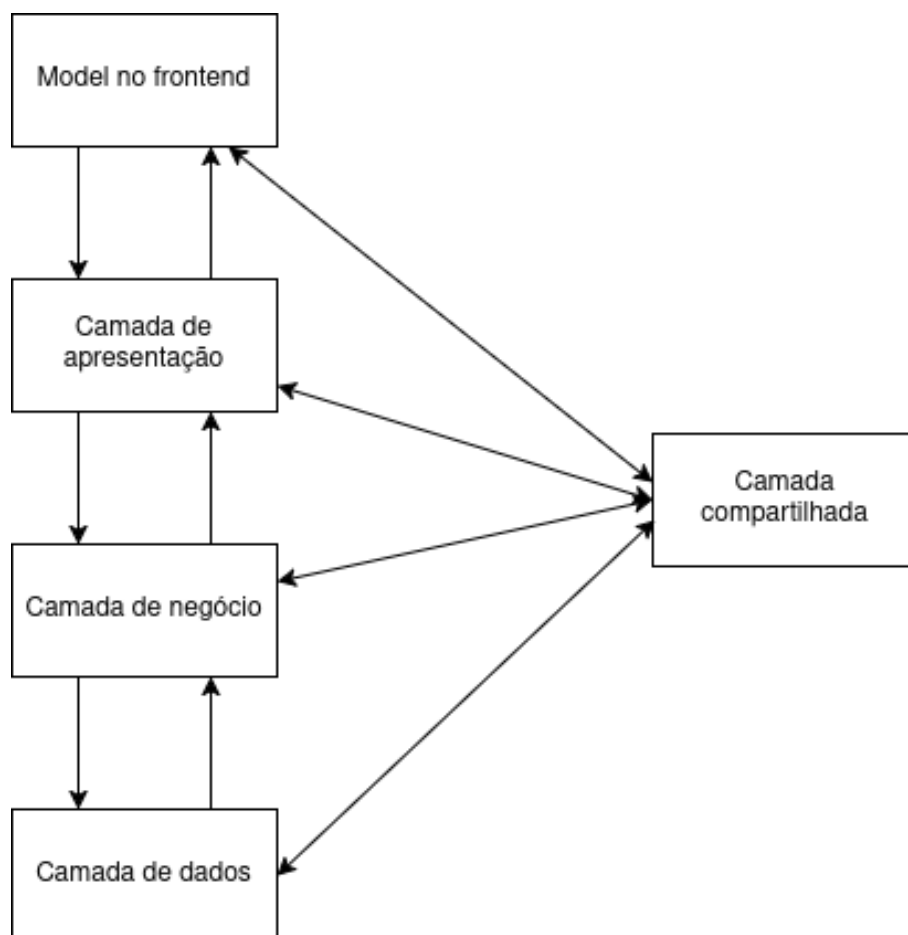


Figura 10 – Arquitetura backend

lógica de apresentação da lógica de negócio. As principais rotas e suas responsabilidades são:

- **GET /exercise/**: Retorna a lista de todos os exercícios cadastrados no sistema, incluindo metadados como nome, categoria, nível de dificuldade e ângulos-alvo para cada articulação monitorada.
- **PATCH /exercise/{exercise_id}**: Define o exercício ativo para a sessão de monitoramento, armazenando o identificador em cache (Redis) para que a conexão WebSocket possa recuperá-lo sem necessidade de estado na camada de apresentação.
- **POST /exercise/**: Cria um novo exercício no banco de dados, permitindo a extensibilidade do sistema com diferentes tipos de movimentos sem alteração de código.
- **WebSocket /monitor**: Estabelece uma conexão persistente para recepção de frames codificados em JPEG (formato binário). Cada frame recebido é decodificado, processado pelo modelo YOLO para extração de keypoints, submetido à lógica de

avaliação de pose e retorna, em tempo real, um objeto JSON contendo estatísticas de execução (contador de repetições e feedback de postura).

- **POST /monitor/reset**: Reinicia o contador de repetições e o estado da máquina de detecção de movimento, permitindo ao usuário recomeçar a sessão de exercícios sem reiniciar a conexão WebSocket.

Adicionalmente, a camada incorpora middleware CORS (Cross-Origin Resource Sharing) para permitir requisições do frontend hospedado em origem distinta, essencial para o desenvolvimento e deploy da aplicação web.

3.5.7.2 Camada de Negócio (**api/services** e **core/**)

Essa camada, fundamental na arquitetura do sistema, atua como um intermediário robusto entre a camada de Apresentação (API) e a Regra de Negócio (Core). Seu papel fundamental reside no encapsulamento e orquestração da lógica de tratamento de requisição e resposta. Ao assumir essa responsabilidade, ela desonera a API, que passa a funcionar estritamente como uma interface de comunicação (recepção e retorno de dados), e o Core, que se concentra exclusivamente na execução da ordem do cliente, como a geração do feedback para os exercícios. A aplicação é estruturada em dois serviços principais: **ExerciseService** e **MonitorService**.

O **ExerciseService** atua como o mediador entre a camada de Apresentação e a camada de Persistência. Sua principal função é fornecer à Apresentação a interface necessária para manipulação dos dados, sem expor diretamente o repositório ou a lógica de persistência. Para isso, ele utiliza o **ExerciseRepository**, delegando a ele os métodos necessários para a interação com o banco de dados. Dessa forma, esta classe se torna responsável por intermediar e pré-processar as requisições oriundas da API, tratando e validando os dados antes que sejam submetidos à inserção (persistência) ou recuperados para visualização, garantindo o encapsulamento da camada de dados.

O **MonitorService** também interage com a camada de persistência, mas com um acesso restrito à leitura (Read-Only) dos dados. Este mecanismo usado para recuperar os parâmetros específicos do exercício a ser monitorado. Além disso, o **MonitorService** atua como um orquestrador, encarregando-se de fornecer à regra de negócio (Core) todas as informações necessárias. Especificamente, ele informa às classes de monitoramento (**Monitor** 3.5.8 ou as implementações de **StaticMonitor**) qual exercício está sendo tratado e quais parâmetros são requeridos para a execução da sua lógica de monitoramento.

A regra de negócio está contida nas classes **Monitor**, **NeckTiltMonitor** e **BicepCurlMonitor**, as últimas as quais implementam o contrato **StaticMonitor**. Sua responsabilidade é processar o frame e gerar o feedback do exercício. Utilizando o modelo YOLO para a extração dos keypoints (pontos-chave) presentes na imagem. Esses keypoints são submetidos a uma cadeia de processamento que inclui filtragem, transformação em vetores e, subsequente,

em ângulos. Por fim, esses ângulos são comparados com os parâmetros de referência estáticos ou, quando aplicável, resgatados pelo `MonitorService` na camada de persistência. O resultado dessa comparação culmina na geração do feedback, que é então apresentado ao usuário.

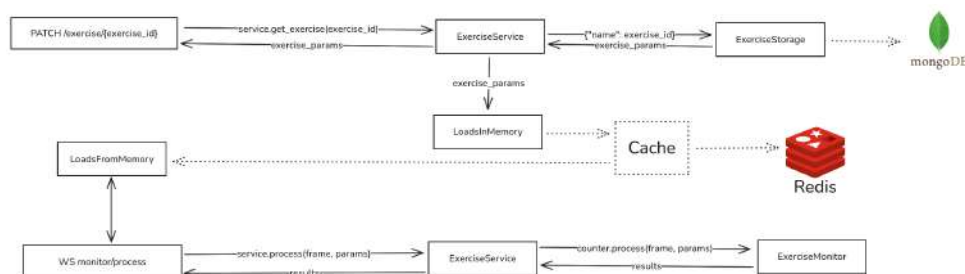


Figura 11 – Avaliação Média de Usabilidade e Funcionalidades Centrais (Escala 1 a 5).

3.5.7.3 Camada de Persistência (shared/repositories)

Esta camada é responsável por abstrair e expor os métodos de persistência, oferecendo uma interface clara para a manipulação dos dados. A aplicação foi concebida para interagir com três fontes de dados distintas, que possuem propósitos e utilizações específicas: MongoDB, Redis e cache estático.

O MongoDB, um banco de dados NoSQL, é empregado com a finalidade de armazenar os parâmetros de monitoramento dos exercícios. A escolha estratégica por essa tecnologia é justificada por sua inerente flexibilidade de esquema (schema-less). Considerando que o armazenamento de exercícios implica em uma estrutura de dados altamente variável em termos de quantidade e especificidade de parâmetros, a versatilidade do modelo NoSQL é fundamental, pois permite uma modelagem de dados mais permissiva e adaptável a essas variações.

O Redis, embora classificado como um banco de dados NoSQL, é empregado por sua especialização em caching (armazenamento em memória). No Nad.IA, ele desempenha o papel crucial de gerenciador de contexto global da aplicação e de estado de sessão. Considerando que a seleção do exercício e o monitoramento são manipulados por rotas (endpoints) distintas, o compartilhamento das informações necessárias seria arquiteturalmente oneroso e ineficiente se dependesse de variáveis globais expostas entre diferentes manipuladores de rota. O uso do Redis soluciona esse desafio de forma limpa e performática, oferecendo um mecanismo eficiente para persistir e recuperar o contexto entre as requisições.

O cache estático é implementado com o uso de um dicionário que armazena as instâncias ativas dos monitores de exercício. A sua importância reside no fato de permitir o armazenamento direto do objeto instanciado em memória, o que é crucial para gerenciar seu estado em diversos endpoints. Diferentemente de soluções de caching como o Redis,

que manipulam primariamente dados serializados (texto), o cache estático oferece o controle necessário para interações diretas com o objeto, como na rota de reset, onde todos os atributos estatísticos do monitor são diretamente reiniciados na instância em tempo real.

3.5.7.4 Camada de Conexão a Dados (shared/connectors)

Esta camada é designada para realizar a conexão direta entre a lógica de persistência e os sistemas de bancos de dados subjacentes. A divisão clara destas responsabilidades é de suma importância, pois torna o código mais testável ao isolar o acesso a dados de outras preocupações. Dessa forma, ela garante que cada camada terá uma responsabilidade única, sendo esta a de lidar com toda a estrutura de conexão e comunicação entre o backend e as fontes de dados.

3.5.7.5 Camada compartilhada (shared/ e utils/)

Por fim, esta camada é designada como o módulo de utilitários e recursos compartilhados, sendo essencial para a fundação da aplicação. Ela é responsável por expor módulos que serão consumidos globalmente pelo sistema, incluindo: o serviço de registro de logs (logger), a implementação de exceções (erros) personalizadas, os modelos de entidade (para a manipulação de dados em banco) e os schemas para as estruturas de requisição e resposta da API.

3.5.8 Análise de Movimento e Feedback em Tempo Real

A análise de movimento é categorizada em dois tipos principais de monitores: estáticos e dinâmicos. Os monitores estáticos (hard coded) contêm seus parâmetros de execução diretamente na definição da classe, como é o caso de `NeckTiltMonitor` e `BicepCurlMonitor`. Esta abordagem permite maior controle e precisão sobre os feedbacks e a execução do movimento monitorado. Estes monitores herdam de uma classe abstrata, `StaticMonitor`, que define o contrato genérico. Por outro lado, o monitor dinâmico, implementado na classe `ExerciseMonitor`, utiliza dados de configuração armazenados em banco de dados, como os pontos-chave e ângulos relevantes para o exercício. O objetivo central é possibilitar uma implementação genérica e reaproveitável para cenários e exercícios com similaridades estruturais.

3.5.8.1 Cálculo de Ângulos Articulares

O sistema Nad.IA realiza o cálculo de ângulos entre segmentos corporais adjacentes — como braço-antebraço (cotovelo) ou coxa-perna (joelho) — para avaliar a amplitude de movimento durante a execução dos exercícios. A partir dos 17 keypoints detectados

pelo modelo YOLOv8-**nano**, o sistema filtra apenas os pontos anatômicos relevantes para o exercício em questão, conforme os parâmetros armazenados no banco de dados MongoDB.

Cada ângulo articular é calculado a partir de três pontos consecutivos utilizando geometria vetorial. Por exemplo, para determinar o ângulo do cotovelo esquerdo, o sistema utiliza as coordenadas 2D dos keypoints correspondentes ao punho esquerdo (p_1), cotovelo esquerdo (p_2) e ombro esquerdo (p_3), aplicando a seguinte formulação:

$$\theta = \arccos \left(\frac{\vec{v}_1 \cdot \vec{v}_2}{|\vec{v}_1| \cdot |\vec{v}_2|} \right)$$

onde $\vec{v}_1 = p_1 - p_2$ e $\vec{v}_2 = p_3 - p_2$ representam os vetores formados entre os segmentos anatômicos.

Para reduzir o impacto de oscilações causadas por instabilidades na detecção frame a frame, o sistema aplica um filtro de média móvel com janela de 5 frames aos ângulos calculados, garantindo maior estabilidade na análise temporal do movimento.

3.5.8.2 Máquina de Estados para Detecção de Movimento

Todas as classes de monitoramento são estruturadas com base em Máquinas de Estados Finitas Determinísticas (MEFD), garantindo maior previsibilidade e constância na detecção de movimentos. As transições entre os estados são acionadas por limiares de tolerância que são aplicados aos ângulos articulares previamente filtrados. A contagem de repetições é efetuada somente quando um ciclo completo é concluído. No monitor dinâmico, por exemplo, o sistema registra uma repetição quando a sequência `IDLE` → `MOVING_UP` → `HOLD` → `MOVING_DOWN` → `IDLE` é finalizada com sucesso, assegurando que o movimento tenha atingido sua amplitude completa.

Estados do ExerciseMonitor	
Estado	Descrição
IDLE (Repouso)	Estado inicial, aguardando o início do movimento.
MOVING_UP (Fase Concêntrica)	Detecção de movimento em direção ao ângulo-alvo (contração muscular).
HOLD (Fase Isométrica)	Manutenção da posição no ângulo-alvo ou próximo dele.
MOVING_DOWN (Fase Excêntrica)	Retorno à posição inicial (extensão muscular).

Tabela 4 – Estados do ExerciseMonitor

3.5.8.3 Ajuste Dinâmico de Dificuldade

O Nad.IA oferece três níveis de dificuldade que ajustam dinamicamente os ângulos-alvo através de fatores de tolerância:

- **Iniciante:** Reduz o ângulo-alvo em 40% (fator 0,6), permitindo maior margem de erro.

Estados do BicepCurlMonitor	
Estado	Descrição
IDLE (Repouso)	Estado inicial, aguardando o início do movimento.
CURLING_UP (Fase Concêntrica)	Deteção de movimento em direção ao topo do movimento.
HOLD_TOP	Manutenção da posição no pico de contração.
LOWERING_DOWN (Fase Excêntrica)	Retorno à posição inicial.

Tabela 5 – Estados do BicepCurlMonitor

Estados do NeckTiltMonitor	
Estado	Descrição
NEUTRAL (Neutro)	Estado inicial, aguardando a inclinação da cabeça para um dos lados (esquerda ou direita).
TILTED_LEFT (Inclinado à Esquerda)	Deteção e manutenção da inclinação da cabeça para o lado esquerdo durante o tempo configurado.
TILTED_RIGHT (Inclinado à Direita)	Deteção e manutenção da inclinação da cabeça para o lado direito durante o tempo configurado.
COMPLETED (Completo)	Ambos os lados (esquerdo e direito) foram completados com sucesso.

Tabela 6 – Estados do NeckTiltMonitor

- **Intermediário:** Reduz em 20% (fator 0,8), exigindo execução mais precisa.
- **Avançado:** Reduz em apenas 10% (fator 0,9), demandando controle motor refinado.

Por exemplo, em um agachamento com ângulo-alvo de 90° para os joelhos, um usuário iniciante precisa atingir aproximadamente 54° para validar a fase HOLD, enquanto um usuário avançado deve alcançar 81°.

3.5.8.4 Validação de Postura e Alinhamento Corporal

Além da análise angular, o sistema verifica continuamente o alinhamento corporal para garantir a segurança e eficácia do exercício. A deteção de múltiplas pessoas no quadro resulta em alerta imediato, impedindo o processamento e instruindo o usuário a garantir que está sozinho no enquadramento da câmera. Quando keypoints críticos não são detectados (por oclusão ou enquadramento inadequado), o sistema identifica especificamente quais articulações estão ausentes e solicita ajuste na posição corporal.

3.5.8.5 Sistema de Feedback Textual em Tempo Real

Diferentemente da abordagem de (ADOLF et al., 2025), que utiliza vídeos pré-gravados com instrutores para guiar os usuários, o Nad.IA emprega um sistema de feedback textual dinâmico e contextualizado. As mensagens são geradas em tempo real pela classe `ExerciseMonitor` de acordo com a fase atual do movimento e a qualidade da execução:

- **Início de movimento:** "Vamos lá, continue!"
- **Execução perfeita** ($\text{erro} \leq 5^\circ$): "Muito bem, execução perfeita!"
- **Sobrepasso do ângulo-alvo:** "Passou do ponto, mas tá valendo!"
- **Conclusão de repetição:** "Repetição {N} concluída! Parabéns!"

Esse mecanismo de feedback textual síncrono é transmitido via WebSocket ao frontend a cada frame processado, permitindo que o usuário ajuste sua técnica instantaneamente durante a execução do exercício, sem necessidade de revisão posterior ou interrupção da atividade.

4 NAD.IA

Neste capítulo, será apresentado o resultado prático do desenvolvimento: a aplicação web de análise de exercícios. Serão detalhadas as principais telas, os fluxos de interação do usuário e como as funcionalidades implementadas atendem aos objetivos propostos por este trabalho.

4.1 VISÃO GERAL DA APLICAÇÃO

A plataforma foi projetada para atender ao requisito de baixo custo e alta acessibilidade, necessitando apenas que o usuário possua um dispositivo com câmera. A interface do usuário foi criada para ser simples, guiando o colaborador desdeo login até a execução dos exercícios e a visualização de seu progresso.

O diagrama de casos de uso Figura 12 descreve o passo a passo da aplicação e os relacionamentos entre o ator (Usuário) e as funcionalidades centrais. O fluxo principal do usuário envolve três ações: Login / Cadastro, Escolher lista de exercícios e Visualizar estatísticas.

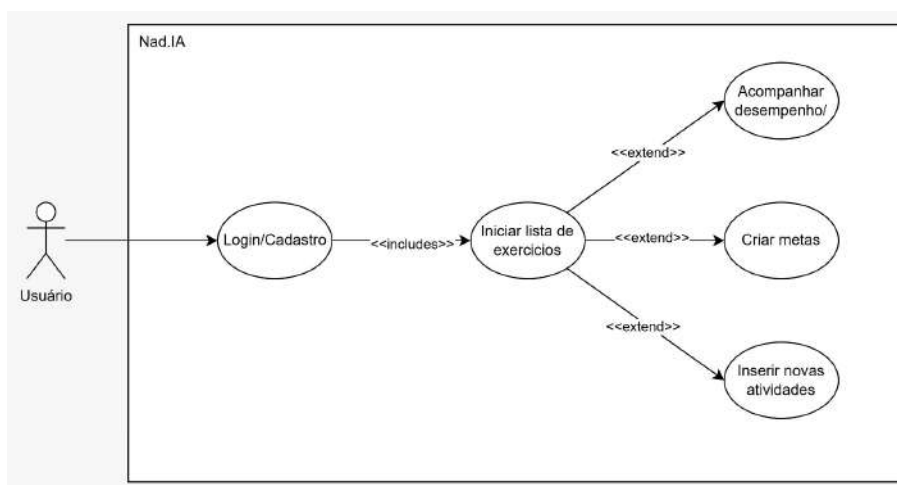


Figura 12 – Caso de usos. Fonte: Elaboração própria.

Conforme detalhado no diagrama, o sistema impõe pré-requisitos lógicos para o acesso às funcionalidades. Tanto o caso de uso “Escolher lista de exercício” quanto o “Visualizar estatísticas” possuem uma relação de dependência com “Login / Cadastro”, indicando que o usuário precisa estar autenticado para acessar ambas as áreas.

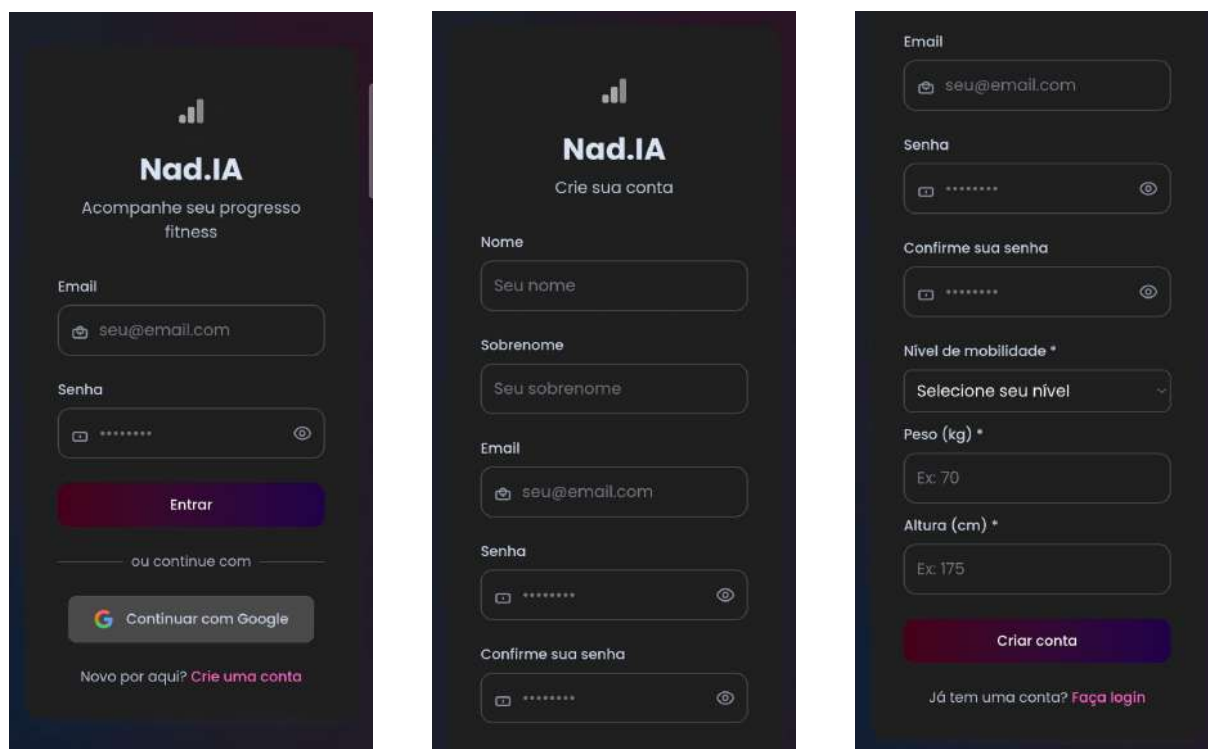
Além disso, para que o usuário possa visualizar estatísticas, é necessário que ele já tenha realizado algum exercício. Isso é representado por uma relação de dependência entre visualizar estatísticas e o fluxo de execução de exercícios (iniciado por escolher lista de exercício). Esta regra garante que o usuário só acesse suas estatísticas após ter dados de desempenho registrados.

Finalmente, o diagrama mostra que o ato de escolher a lista de exercício aciona obrigatoriamente a funcionalidade seguir sequência pré-definida (relação de inclusão), que é o núcleo do sistema de monitoramento em tempo real.

4.2 PRINCIPAIS TELAS

4.2.1 Tela de Login e Cadastro

A Tela de Login é a primeira interface visualizada pelo usuário. Ela centraliza as duas modalidades de acesso disponíveis: o fluxo de login/cadastro padrão (e-mail e senha) e a opção de login via provedor social Google.



(a) Tela de login.

(b) Tela de cadastro.

(c) Tela de cadastro cont.

Figura 13 – Telas de autenticação. Fonte: Elaboração própria.

4.2.2 Recursos Compartilhados

Antes de detalhar as telas específicas, é fundamental descrever os componentes que garantem a navegação e usabilidade em toda a aplicação: o Menu Lateral (Sidebar) e a Barra de Navegação (Header ou Navbar).

4.2.2.1 Menu Lateral (Sidebar)

O Menu Lateral (Figura 15) é um elemento fundamental na navegação do sistema e está presente em todas as telas, otimizando o fluxo do usuário.

- **Identidade e Fechamento:** No topo, exibe a logomarca ("Nad.IA") e um ícone de "X" para ocultar o menu, garantindo mais espaço para o conteúdo principal, especialmente em dispositivos móveis.
- **Acesso Rápido:** Contém as principais rotas da aplicação: Início, Meu Progresso, Configurações e Informações. O item selecionado ("Início") é visualmente destacado para indicar a página atual.
- **Identificação do Usuário:** Na parte inferior, exibe a foto/inicial e o nome do usuário logado ("Francisco Neto"), reforçando a identidade e a sessão ativa.
- **Logout:** Um ícone de sair (seta) ao lado do nome do usuário permite o encerramento imediato da sessão, desvinculando o token de acesso.

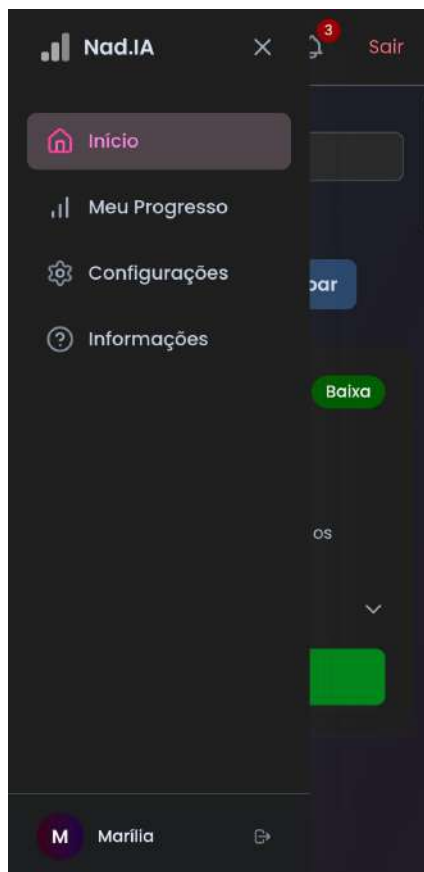


Figura 14 – Menu lateral. Fonte: Elaboração própria.

4.2.2.2 Barra de Navegação (Header)

A barra de navegação superior (Header) é um componente fixo que oferece acesso rápido a funcionalidades essenciais. À esquerda, o ícone de hambúrguer permite a exibição ou ocultação do menu lateral, otimizando a área de trabalho. No lado direito, estão

dispostos: o ícone de notificações (sininho) e a área de perfil do usuário (com nome e foto), que serve como gatilho para o log out da aplicação.



Figura 15 – Barra de navegação. Fonte: Elaboração própria.

4.2.3 Tela Principal (Dashboard)

A Tela Principal (Dashboard), ilustrada na Figura 16, é o ponto central de interação e gestão das listas de exercício. A interface adota um layout otimizado, que prioriza o acesso rápido ao conteúdo através de uma estrutura que se divide claramente em duas zonas funcionais principais:

1. **Filtragem do conteúdo:** Localizada na porção superior, os filtros de conteúdo englobam: barra de pesquisa e filtragem por categoria e nível de exercício. Sua função é permitir que o usuário refine a visualização por meio da Busca Textual (Buscar exercícios...) ou pelos Filtros por Metadados (Todas as categorias e Todos os níveis).
2. **Painel Central de Conteúdo:** Ocupando a maior parte da tela, esta área apresenta as listas de exercícios em um padrão de cards. Cada card exibe as informações essenciais do exercício (Nível, Duração, Foco) e as ações contextuais (Começar Treino, Favoritar, Compartilhar), facilitando a decisão e o engajamento imediato do usuário com o conteúdo.

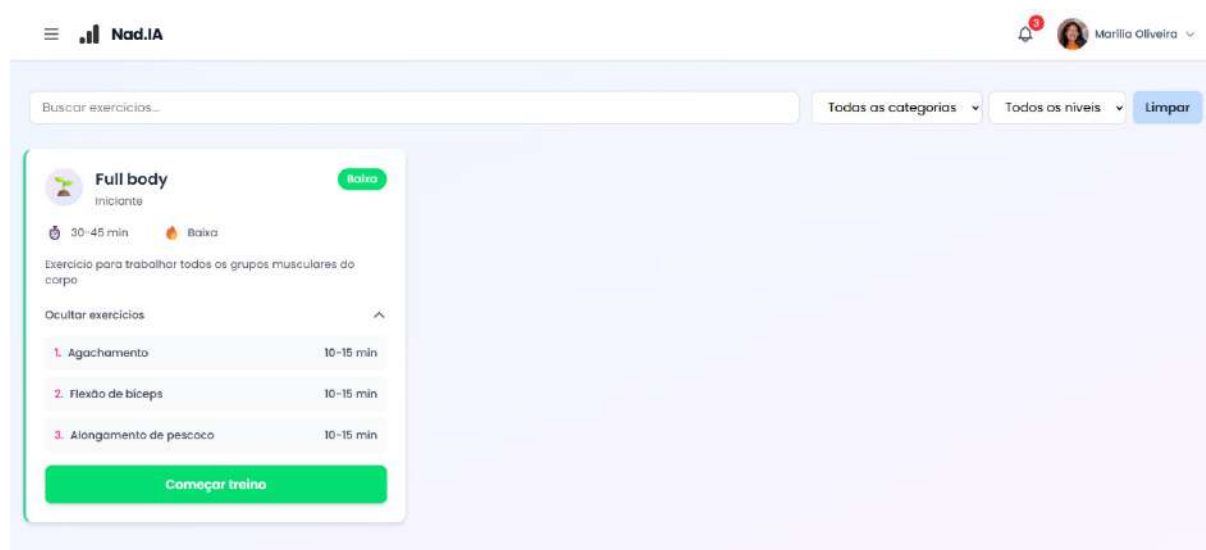


Figura 16 – Tela principal. Fonte: Elaboração própria.

4.2.4 Tela de Configuração do Usuário

A Tela de Configurações de Perfil concentra a gestão das informações do usuário, permitindo a consulta e atualização de dados cadastrais, informações físicas e preferências de compartilhamento e notificação.

Para facilitar a visualização e a descrição técnica, os elementos da tela foram segmentados nas Figuras 17a, 17b e 17c. Deve-se ressaltar que as funcionalidades de notificação e compartilhamento não foram implementadas, estando presentes na interface apenas para fins de demonstração.

Informações Pessoais

Marília de Araujo Santos Oliveira
Membro desde Janeiro 2024
[Alterar foto](#)

Nome Completo:

Email:

Telefone:

Data de Nascimento:

Informações Físicas

Altura (cm):

Peso (kg):

Nível de Atividade:

(a) Informações cadastrais

Informações Físicas

Altura (cm):

Peso (kg):

Nível de Atividade:

(b) Informações físicas

Preferências

Notificações de Treino
Receber lembretes para seus treinos ☒

Compartilhar Progresso
Permitir que outros vejam seu progresso ☐

(c) Preferências (não implementadas)

Figura 17 – Tela de configurações do usuário. Fonte: Elaboração própria.

4.2.5 Execução

A execução do exercício consiste em duas etapas, pré-visualização e execução. A primeira etapa, é um vídeo executado antes de iniciar a execução de fato, como visto na Figura 18.

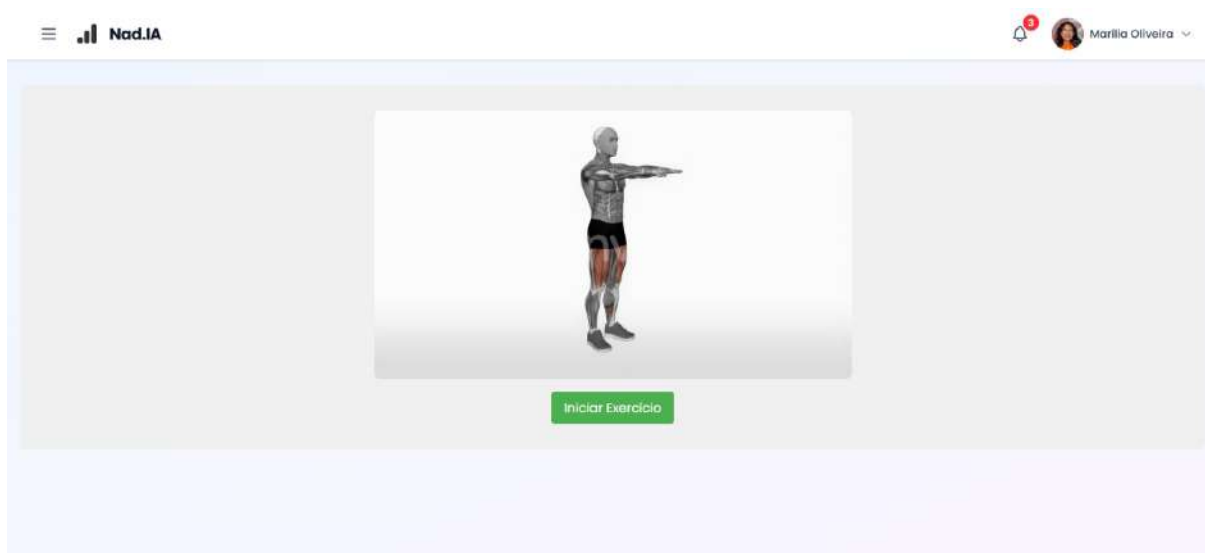


Figura 18 – Pré-visualização do exercício. Fonte: Elaboração própria.

Já a execução do exercício consistem em uma tela com:

- **Quadro de Visualização (Preview da Câmera):** Este componente mostra a execução do exercício
- **Painel de Controle de Tempo e Ações:** Localizado no canto superior direito.
 - **Tempo:** Exibe o cronômetro do treino em formato digital (00 : 00).
 - **Ações:** Contém os botões principais para a gestão da sessão: “Iniciar” (botão primário, em destaque) e “Resetar” (botão secundário).
- **Painel de Próximos Exercícios:** Abaixo do cronômetro, oferece uma visão antecipada da sequência do treino.
 - **Flexão de bíceps:** 3 séries de 2 repetições.
 - **Alongamento de pescoço:** 1 série de 2 repetições.
- **Painel de Exercício Atual:** Localizado na parte inferior da tela, detalha o exercício que está sendo realizado ou prestes a começar.
 - **Nome do Exercício:** Agachamento (visível também como um *tag* no canto).
 - **Detalhes:** “Agachamento, Contagem atual 0/1” (indicando que nenhuma repetição foi completada ainda).

- **Barra de Progresso:** Uma barra horizontal que visualmente representa o avanço do usuário no exercício atual.

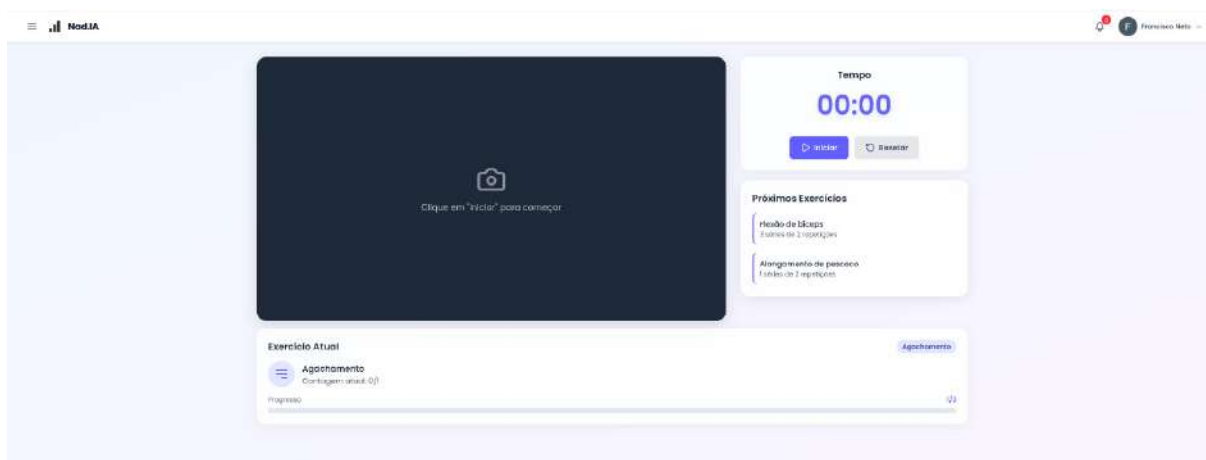


Figura 19 – Tela de execução. Fonte: Elaboração própria.

4.2.6 Tela de Visualização de Resultados

A Tela de Visualização de Resultados (Dashboard) é o ponto central para o engajamento e a manutenção da motivação do usuário, transformando os dados brutos de desempenho em insights acionáveis. A capacidade do usuário de visualizar seu progresso e resultados de forma consolidada é fundamental para validar o esforço e auxiliar na prevenção de dores ou lesões a longo prazo.

O painel apresenta os resultados, provenientes tanto dos dados coletados pelo monitoramento em tempo real, quanto das entradas manuais e atualizações de metas do usuário.

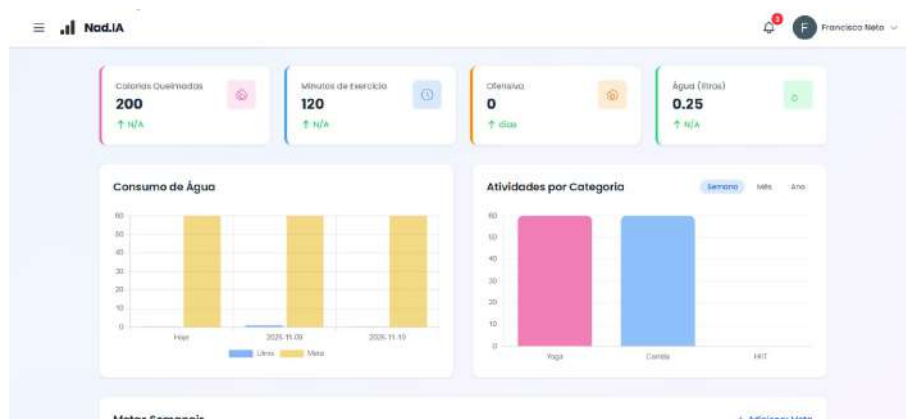
O painel superior Figura 20 e os gráficos oferecem uma visão geral do desempenho do usuário. As informações exibidas são:

- **Calorias Queimadas (Contagem Semanal):** Estimativa do gasto calórico total da semana, calculada com base no tipo e intensidade dos exercícios realizados.
- **Minutos de Exercício (Contagem Semanal):** Sumarização do tempo total dedicado aos treinos ao longo da semana.
- **Água (Litros):** Acompanhamento do consumo de água no dia, podendo ser atualizado manualmente pelo usuário.
- **Consumo de Água (Gráfico):** Gráfico de barras que exhibe o histórico diário do consumo de água versus a meta estabelecida.
- **Ofensiva (Dias Seguidos):** Exibe a sequência de dias consecutivos em que o usuário realizou alguma atividade, incentivando a consistência na prática.

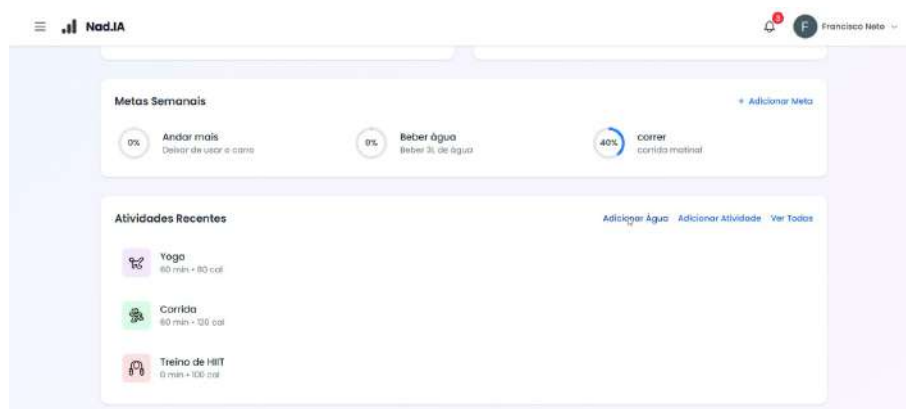
- **Atividades por Categoria:** Gráfico de barras que exhibe o agrupamento das atividades realizadas, segmentadas por categorias (e.g., Yoga, Corrida, HIIT).

O Dashboard foi aprimorado para fornecer maior autonomia ao usuário em seu acompanhamento de saúde e bem-estar, incorporando recursos de gestão de metas e registro de atividades externas. A funcionalidade de Metas Semanais permite que o usuário adicione e acompanhe objetivos personalizados, como "Andar mais", "Beber água 3L/dia" ou "Correr matinal". O progresso dessas metas é exibido de forma clara por meio de anéis de porcentagem, oferecendo um feedback visual imediato sobre o nível de cumprimento.

Complementarmente, a seção de Atividades Recentes e Registro Manual não se limita a exibir o histórico dos treinos monitorados pelo Nad.IA, mas também oferece botões de ação como "Adicionar Água" e "Adicionar Atividade". Essa capacidade permite que o usuário registre atividades físicas realizadas fora da plataforma (como Yoga ou Corrida) e o consumo de água, garantindo, assim, um panorama completo de seu bem-estar geral, mesmo para exercícios que não são submetidos à análise da Visão Computacional.



(a) Painel Superior: Métricas de Desempenho, Consumo de Água e Atividades por Categoria.



(b) Painel Inferior: Metas Semanais e Atividades Recentes (Registro manual).

Figura 20 – Tela de Visualização de Resultados (Dashboard). As informações consolidam dados automáticos (VC) e manuais (metas e atividades). Fonte: Elaboração própria.

4.3 RESULTADOS OBTIDOS

A avaliação da plataforma Nad.IA foi conduzida por meio de um teste de usabilidade exploratório, aplicado a uma amostra de conveniência composta por 9 usuários, com idades entre 17 e 50 anos. Os participantes foram selecionados com base no interesse em atividades físicas mediadas por tecnologia e na disponibilidade, de tempo e espaço, para a realização do teste (critérios de inclusão), sem restrições prévias de nível de condicionamento. A coleta de dados ocorreu em ambiente não controlado, simulando o uso doméstico real, através de dispositivos diversos (4 smartphones, 4 laptops e 1 tablet). Do ponto de vista ético, todos os voluntários foram informados sobre a natureza da pesquisa e concordaram com a participação, embora o estudo não tenha passado por comitê de ética formal por tratar-se de uma avaliação de interface sem coleta de dados sensíveis de saúde.

É fundamental ressaltar que este estudo não possui a pretensão de validar a eficácia clínica ou técnica absoluta da plataforma. A principal ameaça à validade dos resultados reside na ausência de uma homologação formal dos exercícios por um profissional de Educação Física (Personal Trainer) durante os testes, o que limita a comprovação da acurácia biomecânica dos movimentos sugeridos pela Visão Computacional (VC). Além disso, devido ao tamanho reduzido da amostra ($n=9$), os dados não constituem uma base massiva, configurando-se como um estudo de caso focado na percepção de usabilidade, na viabilidade técnica limitada da VC e no impacto subjetivo sobre a motivação e prevenção de lesões. Os resultados detalhados seguem nas subseções abaixo.

4.3.1 Usabilidade e Funcionalidades Centrais

A percepção geral de usabilidade do Nad.IA (em uma escala de 1 a 5) revelou um alto nível de aceitação, com a maioria das métricas obtendo média aproximada em 4.5.

Avaliação Média de Usabilidade e Funcionalidades	
Prevencao Lesões (1-5)	4.67
Feedback Tempo Real (1-5)	4.56
Facilidade Geral (1-5)	4.44
Contagem Repetições (1-5)	4.33

Conforme mostrado na Tabela 4.3.1, a funcionalidade de **Contagem Automática de Repetições** e o potencial de **Prevenção de Lesões** (avaliação de crença) excelentes médias ($\sim 4,7$) e ($\sim 4,3$), respectivamente, seguidos pela **Facilidade de Uso Geral** e o **Feedback em Tempo Real** ($\sim 4,4$).

4.3.1.1 Facilidade de Ações Chave

A usabilidade em ações operacionais específicas da interface também foi amplamente positiva. A Figura 21 detalha a distribuição das respostas para ações críticas:

Facilidade Percebida para Ações Chave na Plataforma

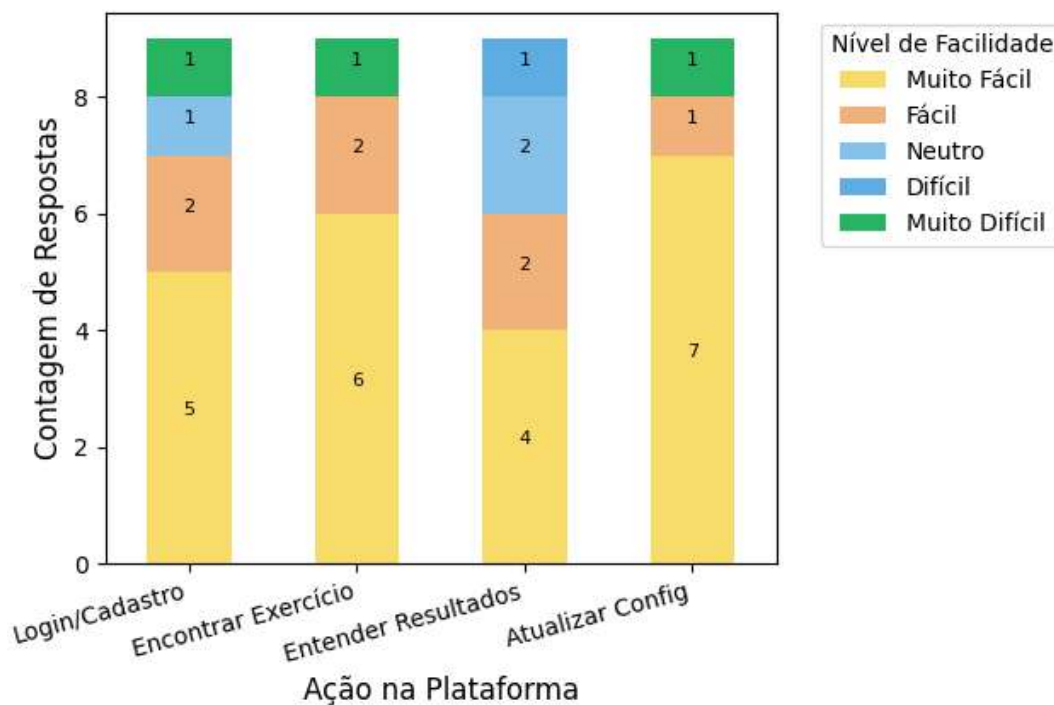


Figura 21 – Facilidade Percebida para Ações Chave na Plataforma (Contagem).

Em todas as ações, a grande maioria dos usuários avaliou a experiência como "**Muito Fácil**" ou "**Fácil**". O processo de **Atualizar Configurações** foi a única ação com uma avaliação "Muito Difícil" (1 usuário), mas também uma das que mais obteve "Muito Fácil" (4 usuários), sugerindo um ponto de fricção específico a ser investigado.

4.3.2 Avaliação da Visão Computacional e Desempenho

As funcionalidades de VC e o desempenho em tempo real foram validados, mas também apontaram para as limitações técnicas do protótipo (Capítulo 5.2).

- **Eficácia do Feedback (Correção):** A totalidade dos usuários (9 de 9) considerou que o feedback ajudou a executar o movimento de forma mais correta.
- **Alertas de Segurança:** A maioria dos usuários (6 de 9) recebeu alertas de segurança (e.g., ajuste de posição). Destes, 5 classificaram o alerta como "claro e útil", validando a funcionalidade de análise de alinhamento corporal.

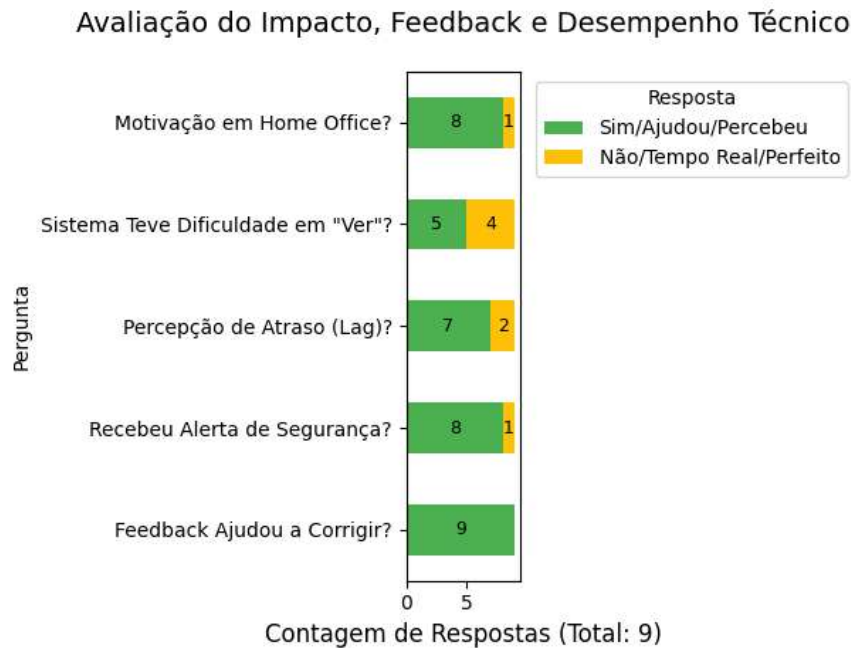


Figura 22 – Avaliação do Impacto, Feedback e Desempenho Técnico (Contagem).

- **Latência (Lag):** Seis dos 9 usuários perceberam um atraso (*lag*) significativo entre o movimento e a resposta do sistema. Contudo, todos os 6 que perceberam o atraso afirmaram que ele "*não chegou a atrapalhar*" a experiência, corroborando a análise de que o uso de *WebSockets* amorteceu o impacto da latência, mesmo sem GPU dedicada.
- **Dificuldade do Sistema:** Aproximadamente, metade dos usuários (4 de 9) relatou que o sistema teve dificuldade em "ver" os movimentos em algum momento, indicando a necessidade de maior precisão no reconhecimento de pose, como sugerido na seção de Trabalhos Futuros.

4.3.3 Impacto e Perspectivas de Melhoria

Os resultados obtidos demonstram que o Nad.IA possui um elevado potencial de aceitação no cenário de *home office*. A adesão de 100% dos participantes quanto à motivação e percepção de segurança na prevenção de lesões indica que a ferramenta preenche uma lacuna importante na saúde ocupacional remota. Contudo, a análise profunda dos dados revela nuances entre a expectativa do usuário e as limitações tecnológicas atuais.

4.3.3.1 Análise das Funcionalidades e Percepção de Valor

A análise qualitativa permitiu identificar os pilares de valor do sistema sob a ótica do usuário:

1. **Acessibilidade e Autonomia:** A flexibilidade de execução em ambiente doméstico foi o ponto mais citado. Isso sugere que o Nad.IA atua como um facilitador de saúde, removendo barreiras logísticas (tempo e deslocamento) que frequentemente levam ao sedentarismo no trabalho remoto.
2. **Interatividade Cognitiva:** O feedback em tempo real e as instruções auditivas não foram vistos apenas como auxílio técnico, mas como uma companhia para o exercício. Apesar de a literatura apontar que o feedback imediato não é, estatisticamente, impactante (ADOLF et al., 2025) demonstrou ser um dos principais recursos do Nad.IA.

4.3.3.2 Discussão de Limitações e Recomendações

Embora a recepção tenha sido positiva, a fase de testes e o debate acadêmico subsequente trouxeram à tona limitações críticas que devem ser discutidas para evitar a superficialidade descritiva:

1. **Limitação computacional:** Observou-se que a experiência do usuário está intrinsecamente ligada à capacidade de processamento em tempo real do servidor. Esta limitação técnica comprometeu os sistemas de feedback e contagem, uma vez que a latência no processamento dos frames impediu a atualização fidedigna dos dados em determinados intervalos.
2. **Evolução para Ecossistema de Saúde:** As sugestões de gamificação, vídeos explicativos e integração com bioimpedância indicam que os usuários não percebem o Nad.IA apenas como um corretor pontual, mas como um potencial hub de monitoramento de saúde. Isso aponta para um caminho de desenvolvimento onde a inteligência artificial deixa de ser meramente funcional e passa a ser estratégica na gestão da saúde individual.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento de um sistema completo de análise de movimentos corporais em tempo real, explorando técnicas de visão computacional e processamento de imagens para contagem e avaliação de exercícios físicos. Além do processamento de imagens propriamente dito, foi desenvolvida uma interface web que permite ao usuário acompanhar seu progresso de forma simples e intuitiva, integrando visualização em tempo real com funcionalidades de acompanhamento de desempenho. Ao longo do desenvolvimento, foram identificados avanços significativos, desafios técnicos e oportunidades de melhorias futuras.

A abordagem determinística adotada para análise de movimentos demonstrou eficácia principalmente na contagem automática de repetições. No entanto, a precisão do feedback fornecido ao usuário durante a execução dos exercícios apresentou limitações. A análise baseada em um número reduzido de pontos corporais resultou em dificuldades para o algoritmo distinguir determinados movimentos similares, comprometendo a qualidade do retorno em tempo real.

5.1 PRINCIPAIS APRENDIZADOS

O desenvolvimento deste projeto proporcionou aprendizados sobre aspectos técnicos e arquiteturais. O processamento de imagens em tempo real revelou-se consideravelmente mais complexo do que inicialmente previsto, demandando atenção especial à integração de diversas tecnologias. A implementação de comunicação via WebSockets, o sincronismo entre backend e frontend, e a manutenção da fluidez visual durante o processamento dos frames exigiram soluções cuidadosas e otimizações constantes.

Outro aprendizado fundamental foi a importância de uma arquitetura bem planejada desde as fases iniciais do projeto. A definição de uma estrutura sólida não apenas facilitou o desenvolvimento incremental da solução, como também contribuiu significativamente para a manutenibilidade do código e o desempenho geral do sistema, permitindo identificar e resolver gargalos com maior agilidade.

5.2 LIMITAÇÕES TÉCNICAS

A principal limitação técnica enfrentada foi a ausência de recursos de processamento gráfico dedicado (GPU) para análise dos frames em tempo real. Esta restrição resultou em atrasos perceptíveis entre a captura do movimento e o retorno do feedback ao usuário, impactando negativamente a experiência de uso do sistema.

Adicionalmente, a indisponibilidade de GPU impossibilitou a realização de *fine-tuning* do modelo YOLO, o que teria permitido personalizar o modelo para as especificidades do projeto, melhorando potencialmente a precisão na detecção e classificação dos movimentos analisados.

5.3 TRABALHOS FUTUROS

Com base nas experiências e limitações identificadas ao longo deste trabalho, diversas oportunidades de aprimoramento e expansão do sistema foram identificadas, abrindo caminhos promissores para pesquisas futuras.

Uma direção fundamental consiste no desenvolvimento de uma infraestrutura robusta para extração, armazenamento e análise de dados dos usuários. Atualmente, o sistema foca principalmente na análise em tempo real, mas a criação de um repositório estruturado de dados históricos, indo além do monitoramento básico como consumo de água e perda de peso, permitiria o acompanhamento da evolução individual de cada usuário ao longo do tempo. Esta infraestrutura possibilitaria a identificação de padrões de desempenho, tendências de progresso e até mesmo a detecção de possíveis riscos de lesão por sobrecarga ou execução inadequada recorrente. A partir desses dados, seria possível gerar relatórios personalizados e visualizações que auxiliem tanto os usuários quanto profissionais de educação física no planejamento e ajuste de rotinas de treinamento.

Outro avanço significativo seria o desenvolvimento e treinamento de um modelo de aprendizado de máquina especializado na categorização automática de movimentos do corpo humano. A abordagem determinística atual, embora funcional para casos específicos, apresenta limitações quando confrontada com variações individuais de execução, diferentes biotipos e novos tipos de exercícios. Um modelo treinado com dados diversificados seria capaz de aprender representações mais ricas dos movimentos, reduzindo a dependência de regras fixas e limiares manualmente ajustados. Esta transição para uma abordagem baseada em aprendizado permitiria maior generalização do sistema, facilitando a inclusão de novos exercícios sem a necessidade de reprogramação extensiva e melhorando a precisão na detecção de sutilezas na execução dos movimentos.

Complementarmente ao modelo de classificação, propõe-se o desenvolvimento de um sistema de feedback inteligente e contextualizado. A integração dos resultados do modelo de classificação com uma base de conhecimento estruturada de orientações técnicas permitiria a geração automática de feedbacks personalizados e pedagogicamente adequados. Este sistema poderia não apenas identificar erros na execução, mas também priorizar correções baseadas em critérios de segurança e efetividade, adaptar a linguagem do feedback ao nível de experiência do usuário e oferecer progressões ou regressões de exercícios quando apropriado. A incorporação de técnicas de processamento de linguagem natural poderia, inclusive, permitir a geração de explicações mais detalhadas sobre o motivo de

determinada correção, contribuindo para o processo educativo do usuário.

Adicionalmente, melhorias na infraestrutura computacional representam uma necessidade crítica para evolução do sistema. A incorporação de processamento em GPU não apenas reduziria significativamente a latência do feedback, como também viabilizaria o uso de modelos mais sofisticados e a análise de um número maior de pontos corporais simultaneamente. Esta evolução técnica abriria portas para análises biomecânicas mais detalhadas e precisas.

O aprimoramento do mecanismo de envio das imagens representa um potencial e relevante ponto de pesquisa. Abordagens que empregam compressão de imagem podem aliviar a sobrecarga de dados tanto no envio quanto no processamento subsequente, esta última não foi implementada por limitações de prazo. Além disso, o pré-processamento dos frames no lado do cliente (client-side) constitui uma possível e promissora alternativa. Neste último ponto, duas abordagens principais poderiam ser seguidas:

1. O envio de zonas de interesse (ZOIs), onde o backend receberia apenas os dados essenciais para o processamento do exercício específico.
2. A realização da estimativa de pose integralmente no lado do cliente, resultando no envio dos keypoints ao backend.

Embora o presente trabalho não tenha adotado a segunda abordagem, visto que a alocação do processamento para o cliente poderia comprometer o feedback em tempo real, em função de possíveis limitações do hardware do usuário, novas pesquisas e testes nesse sentido são sempre encorajados e bem-vindos.

Por fim, a expansão do sistema para contemplar aspectos de gamificação e engajamento social poderia aumentar significativamente a adesão dos usuários. Elementos como desafios, conquistas, comparações com histórico pessoal e funcionalidades colaborativas poderiam transformar o sistema em uma plataforma mais completa de apoio à prática de atividades físicas.

O desenvolvimento deste sistema demonstrou o potencial das técnicas de visão computacional aplicadas ao contexto de atividades físicas, ao mesmo tempo em que evidenciou os desafios inerentes ao processamento em tempo real e à necessidade de recursos computacionais adequados para soluções mais robustas e precisas. As direções futuras propostas representam caminhos naturais de evolução que poderiam transformar este protótipo em uma ferramenta verdadeiramente abrangente e efetiva de auxílio à prática segura e eficiente de exercícios físicos.

REFERÊNCIAS

- ADOLF, J. et al. Offistretch: Camera-based real-time feedback for daily stretching exercises. **The Visual Computer**, v. 41, p. 1555–1571, 2025. Publicado online em 28 de maio de 2024. Disponível em: <https://doi.org/10.1007/s00371-024-03450-y>.
- CARVALHO, T. N.; LESSA, M. R. **Sedentarismo no ambiente de trabalho: os prejuízos da postura sentada por longos períodos**. 2025. Trabalho acadêmico / Relatório. Acesso em: 19 set. 2025.
- GUISELINI, M. **A importância da mobilidade e da flexibilidade para a saúde**. 2025. <https://revistaempresariofitness.com.br/personal-trainer/a-importancia-da-mobilidade-e-da-flexibilidade-para-a-saude/>. Acesso em: 20/09/2025.
- LACOMBE, P. **Qual a importância do alinhamento postural e como melhorá-lo?** 2020. <https://patricialacombe.com.br/blog/qual-a-importancia-do-alinhamento-postural-e-como-melhora-lo/>. Acesso em: 20/09/2025.
- LIN, T.-Y. et al. **Microsoft COCO: Common Objects in Context**. 2015. Disponível em: <https://arxiv.org/abs/1405.0312>.
- MO, L. et al. **Human Physical Activity Recognition Based on Computer Vision With Deep Learning Model**. 2016. <https://ieeexplore.ieee.org/document/7520541>. Acesso em: 06 jun. 2025.
- NETWORK, B. N. **Corporate Wellness Market Trends in Digital Health Solutions and Workplace Engagement**. 2025. Digital Journal, Press Release. Relatório menciona que o mercado global de wellness corporativo deve alcançar USD 120,23 bilhões até 2032, com CAGR de 7,1% a partir de 2025. Acesso em: 21 set. 2025. Disponível em: <https://www.digitaljournal.com/pr/news/binary-news-network/corporate-wellness-market-trends-digital-1414715389.html>.
- ORTOPEDIA, R. P. da. **Prevenção de acidentes domésticos: cuidados importantes para evitar quedas e outros acidentes em casa**. 2023. <https://portaldortaortopedia.com.br/ortopedia/prevencao/>. Acesso em: 20/09/2025.
- POWERER. **OpenPose + Torchvision + YoloV7 + Mediapipe [Vídeo]**. 2023. <https://www.youtube.com/watch?v=uedp3CnXWmM>. Acessado em: 18 set. 2025.
- SILVA, I. N. F. da; CUNHA, N. C. Ginástica laboral: Benefícios e melhoria da qualidade de vida – estudo de caso. **Revista GeTeC**, v. 6, n. 13, 2017. Acesso em: 19 set. 2025.
- SOUSA, D. S. S. **Seguimento do Movimento Humano Usando Visão Computacional – Aplicação na Análise da Marcha**. Dissertação (Mestrado) — Faculdade de Engenharia da Universidade do Porto, Porto, Portugal, 2008. Orientador: João Manuel R. S. Tavares; Coorientador: Miguel Velhote Correia. Disponível em: <https://repositorio-aberto.up.pt/handle/10216/13013>.
- ULTRALYTICS. **Documentação ultralytics**. 2025. <https://docs.ultralytics.com/tasks/pose/>. Acessado em: 18 set. 2025.

ULTRALYTICS. **ultralytics**. 2025. <https://github.com/ultralytics/ultralytics?tab=readme-ov-file#-models>. Acessado em: 18 set. 2025.

WELLS, J. et al. A systematic review of the impact of remote working referenced to the concept of work–life flow on physical and psychological health. **Workplace Health & Safety**, v. 71, n. 11, p. 507–521, 2023. Acesso em: 19 set. 2025.

World Health Organization. **Musculoskeletal conditions**. 2022. Disponível em: <https://www.who.int/news-room/fact-sheets/detail/musculoskeletal-conditions>, Acesso em: 21 set. 2025.

WU, J.; AL. et. Prevalence of work-related musculoskeletal disorders among workers in the automobile manufacturing industry in china: a systematic review and meta-analysis. **BMC Public Health**, v. 23, 2023. Acesso em: 20 set. 2025.

ZHANG, A. et al. **Dive into Deep Learning**. 2025. https://pt.d2l.ai/chapter_computer-vision/semantic-segmentation-and-dataset.html. Acessado em: 18 set. 2025.

APÊNDICES

Avaliação da Plataforma Nad.IA

(Pesquisa de TCC)

Olá! Este formulário faz parte da pesquisa do Trabalho de Conclusão de Curso (TCC) sobre o **Nad.IA**, um sistema de acompanhamento de exercícios físicos com feedback em tempo real.

O objetivo é coletar sua opinião sobre a experiência de uso da plataforma. Suas respostas são anônimas e fundamentais para a avaliação do projeto. O preenchimento leva cerca de 5 minutos.

Acesse a plataforma em: <https://nadia-two.vercel.app/>

Obrigado pela sua participação!

* Indica uma pergunta obrigatória

1. Qual sua faixa etária? *

Marcar apenas uma oval.

- ☐ Até 25 anos
- ☐ 26 a 35 anos
- ☐ 36 a 45 anos
- ☐ 46 anos ou mais

2. Atualmente, qual seu regime de trabalho principal? *

Marcar apenas uma oval.

- ☐ Trabalho 100% remoto (home office)
- ☐ Trabalho híbrido (parte remoto, parte presencial)
- ☐ Trabalho 100% presencial
- ☐ Não estou trabalhando no momento
- ☐ Outro

3. Com que frequência você pratica exercícios físicos ou ginástica laboral? *

Marcar apenas uma oval.

- ☐ Diariamente
- ☐ Algumas vezes por semana
- ☐ Raramente
- ☐ Nunca

4. Em qual dispositivo você utilizou o Nad.IA? *

Marque todas que se aplicam.

- ☐ Notebook (com webcam)
- ☐ Computador de mesa (com webcam)
- ☐ Celular
- ☐ Tablet

Usabilidade e Interface

5. Em uma escala de 1 a 5, como você avalia a facilidade de uso geral da plataforma Nad.IA? *

Marcar apenas uma oval.

1	2	3	4	5
☐	☐	☐	☐	☐

6. Avalie a facilidade para realizar as seguintes ações *

Marcar apenas uma oval por linha.

	Muito Difícil	Difícil	Neutro	Fácil	Muito Fácil
Realizar o login/ cadastro	☐	☐	☐	☐	☐
Encontrar um exercício na tela principal	☐	☐	☐	☐	☐
Entender as informações na tela de resultados/ progresso	☐	☐	☐	☐	☐
Atualizar suas informações na tela de "Configuração do Usuário"	☐	☐	☐	☐	☐

Avaliação das Funcionalidades Principais

7. Sobre a funcionalidade de CONTAGEM AUTOMÁTICA de repetições: *

Marcar apenas uma oval.

	1	2	3	4	5	
Nad	☐	☐	☐	☐	☐	Muito precisa

8. Sobre o FEEDBACK EM TEMPO REAL (mensagens de texto como "Muito bem!", "Continue!"): *

Marcar apenas uma oval.

1 2 3 4 5

Inúti ☐ ☐ ☐ ☐ ☐ Extremamente útil (ajudou a guiar o exercício)

9. Você sentiu que o feedback do Nad.IA ajudou você a executar o movimento de forma mais correta? *

Marcar apenas uma oval.

- ☐ Sim, ajudou bastante.
☐ Sim, ajudou um pouco.
☐ Não fez diferença.
☐ Não, senti que o feedback estava incorreto.

10. Você recebeu algum alerta de segurança (ex: "Mais de uma pessoa na câmera" ou "Ajuste sua posição")? *

Marcar apenas uma oval.

- ☐ Sim, e o alerta foi claro e útil.
☐ Sim, mas não entendi o alerta.
☐ Não, não recebi alertas.

11. Você percebeu algum atraso (lag) significativo entre o seu movimento e a resposta do sistema? *

Marcar apenas uma oval.

- ☐ Sim, um atraso perceptível que atrapalhou a experiência
☐ Sim, um pequeno atraso, mas que não chegou a atrapalhar.
☐ Não, o sistema pareceu funcionar em tempo real.

12. O sistema teve dificuldade em "ver" você ou seus movimentos em algum momento (ex: travou ou não recebeu o feedback)? *

Marcar apenas uma oval.

- ☐ Sim, frequentemente.
- ☐ Sim, algumas vezes.
- ☐ Não, funcionou perfeitamente.

13. Você acredita que usar o Nad.IA regularmente ajudaria a prevenir dores ou lesões causadas pela má execução de exercícios? *

Marcar apenas uma oval.

	1	2	3	4	5	
Disc	☐	☐	☐	☐	☐	Concordo totalmente

14. (Pense no contexto de home office) Você se sentiria mais motivado a fazer pausas para ginástica laboral se tivesse o Nad.IA disponível? *

Marcar apenas uma oval.

- ☐ Sim, com certeza.
- ☐ Talvez.
- ☐ Não, isso não mudaria minha motivação.

15. Qual funcionalidade do Nad.IA você achou MAIS valiosa?

16. Qual funcionalidade do Nad.IA você acha que MAIS precisa de melhorias?

Comentários Finais

17. Você encontrou algum bug (erro) ou teve alguma dificuldade que não foi mencionada?

18. Você tem alguma sugestão de melhoria ou funcionalidade que gostaria de ver no Nad.IA?

Este conteúdo não foi criado nem aprovado pelo Google.

Google Formulários

	INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DA PARAÍBA
	Campus Campina Grande - Código INEP: 25137409
	R. Tranquílino Coelho Lemos, 671, Dinamérica, CEP 58432-300, Campina Grande (PB)
	CNPJ: 10.783.898/0003-37 - Telefone: (83) 2102.6200

Documento Digitalizado Ostensivo (Público)

Versão Final do TCC

Assunto:	Versão Final do TCC
Assinado por:	Francisco Neto
Tipo do Documento:	Avaliação
Situação:	Finalizado
Nível de Acesso:	Ostensivo (Público)
Tipo do Conferência:	Cópia Simples

Documento assinado eletronicamente por:

- Francisco Bezerra da Silva Neto, DISCENTE (202111250014) DE BACHARELADO EM ENGENHARIA DE COMPUTAÇÃO - CAMPINA GRANDE, em 12/01/2026 18:55:38.

Este documento foi armazenado no SUAP em 12/01/2026. Para comprovar sua integridade, faça a leitura do QRCode ao lado ou acesse <https://suap.ifpb.edu.br/verificar-documento-externo/> e forneça os dados abaixo:

Código Verificador: 1727627
Código de Autenticação: fdcb6326d0

