

**UNIVERSIDADE PAULISTA
INSTITUTO DE CIÊNCIA E TECNOLOGIA
CURSO DE CIÊNCIA DA COMPUTAÇÃO**

ENZO MANOEL LOZANO
FELIPE DO PRADO DE CASTRO
IASMIN BOAVENTURA PEDROZO
MARIA LUIZA ALVES DA SILVA MACHADO
PAULO RICARDO DE OLIVEIRA FERNANDES
VICTOR FERNANDO RODRIGUES FRANCISCHELLI

SISTEMA AUTOMATIZADO PARA CÁLCULO DE ÂNGULO DE COBB:

Inteligência Artificial Aplicada à Detecção de Escoliose

São José dos Campos - SP

2025

ENZO MANOEL LOZANO
FELIPE DO PRADO DE CASTRO
IASMIN BOAVENTURA PEDROZO
MARIA LUIZA ALVES DA SILVA MACHADO
PAULO RICARDO DE OLIVEIRA FERNANDES
VICTOR FERNANDO RODRIGUES FRANCISCHELLI

SISTEMA AUTOMATIZADO PARA CÁLCULO DE ÂNGULO DE COBB:

Inteligência Artificial Aplicada à Detecção de Escoliose

**Trabalho de Conclusão de Curso apresentado
ao Instituto de Ciências e Tecnologia da
Universidade Paulista, como parte dos
requisitos necessários para a obtenção do título
de Bacharel em Ciência da Computação**

Orientador: Prof. Dr. Lamartine Nogueira
Frutuoso Guimarães.

São José dos Campos - SP

2025

CIP - Catalogação na Publicação

Lozano, Enzo Manoel

Sistema automatizado para cálculo de ângulo de Cobb : Inteligência artificial aplicada à detecção de escoliose / Enzo Manoel Lozano, Felipe Do Prado de Castro, Iasmin Boaventura Pedrozo, Maria Luiza Alves Da Silva Machado, Paulo Ricardo De Oliveira Fernandes, Victor Fernando Rodrigues Francischelli - 2025.

106 f. : il. color

Trabalho de Conclusão de Curso (Graduação) apresentado ao Instituto de Ciência Exatas e Tecnologia da Universidade Paulista, São José dos Campos, 2025.

Área de Concentração: Visão Computacional.

Orientador: Prof. Dr. Lamartine Nogueira Frutuoso Guimarães.

1. Escoliose. 2. Ângulo de Cobb. 3. Inteligência Artificial. 4. Visão Computacional. 5. YOLOv8. I. Castro, Felipe Do Prado de II. Pedrozo, Iasmin Boaventura. III. Machado, Maria Luiza Alves da Silva. IV. Fernandes, Paulo Ricardo de Oliveira. V. Francischelli, Victor Fernando Rodrigues. VI. Guimarães, Lamartine Nogueira Frutuoso (orientador). VII. Título.

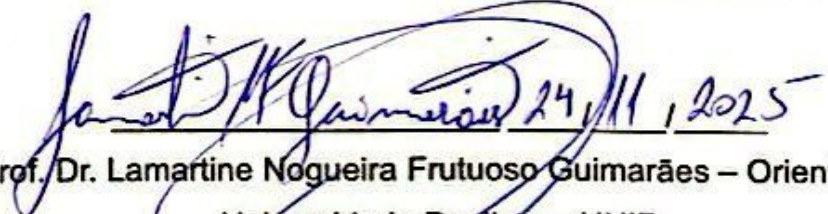
ENZO MANOEL LOZANO
FELIPE DO PRADO DE CASTRO
IASMIN BOAVENTURA PEDROZO
MARIA LUIZA ALVES DA SILVA MACHADO
PAULO RICARDO DE OLIVEIRA FERNANDES
VICTOR FERNANDO RODRIGUES FRANCISCHELLI

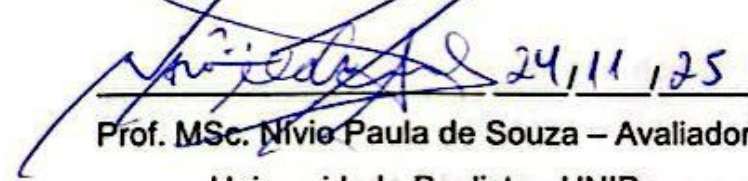
SISTEMA AUTOMATIZADO PARA CÁLCULO DE ÂNGULO DE COBB:
Inteligência Artificial Aplicada à Detecção de Escoliose

Trabalho de Curso apresentado ao Instituto de Ciências Exatas e Tecnologia da Universidade Paulista, como parte dos requisitos necessários para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof. Dr. Lamartine Nogueira Frutuoso Guimarães.

O trabalho foi considerado Aprovado com a nota note, média 9,5).
São José dos Campos, 19 de Novembro 2025.


Prof. Dr. Lamartine Nogueira Frutuoso Guimarães – Orientador
Universidade Paulista – UNIP


Prof. MSc. Nívia Paula de Souza – Avaliador
Universidade Paulista - UNIP

**Ao Curso de Ciência da Computação
da Universidade Paulista, às pessoas
que estiveram conosco nos obstáculos
e às nossas famílias.**

AGRADECIMENTOS

A Deus, por nossa vida, saúde e força.

Aos nossos familiares, pelo amor, incentivo e apoio incondicional.

Aos amigos, pela companhia e motivação nos momentos desafiadores.

Ao nosso orientador, Prof. Lamartine, pela paciência, ensinamentos e orientação dedicada, fundamentais para a realização deste trabalho.

Aos Profs. Fernando Mauro e André Kusumoto, pelos ensinamentos ao longo de todos estes anos.

E, de forma especial, à experiência pessoal de uma de nossas integrantes com a escoliose, que serviu como inspiração para a escolha do tema deste trabalho, transformando um desafio individual em um estudo científico que busca contribuir para a área da saúde.

“A tecnologia sozinha não é suficiente. É a combinação da tecnologia com as pessoas que faz a diferença.”

(Steve Jobs)

RESUMO

A escoliose é uma condição de saúde que afeta a curvatura da coluna vertebral, exigindo avaliações médicas precisas para diagnóstico e acompanhamento. Este trabalho tem como objetivo desenvolver uma aplicação utilizando visão computacional e inteligência artificial para calcular, a partir de imagens de raio-X, o ângulo de Cobb, fundamental para a análise dessa condição. Para isso, emprega-se a rede *You Only Look Once* versão 8 (YOLOv8), treinada com um conjunto público de imagens de raio-X, de modo a realizar a detecção automática das vértebras envolvidas no cálculo do ângulo. A aplicação é destinada a médicos, que poderão gerar relatórios com base nas imagens enviadas pelos pacientes, promovendo maior eficiência e precisão no processo. A metodologia envolve o desenvolvimento de um modelo de inteligência artificial integrado a uma interface que facilita o uso por profissionais da saúde. Espera-se que o sistema contribua para a simplificação do diagnóstico e do monitoramento da escoliose, otimizando o tempo dos médicos e melhorando a experiência dos pacientes.

Palavras-chave: Escoliose; Ângulo de Cobb; Inteligência Artificial; Visão Computacional; YOLOv8; Diagnóstico Médico.

ABSTRACT

Scoliosis is a health condition that affects the curvature of the spine, requiring accurate medical evaluations for diagnosis and follow-up. This work aims to develop an application using computer vision and artificial intelligence to calculate, from X-ray images, the Cobb angle, which is essential for analyzing this condition. For this purpose, the You Only Look Once version 8 (YOLOv8) network is employed, trained with a public dataset of X-ray images, in order to automatically detect the vertebrae involved in the angle calculation. The application is intended for physicians, who will be able to generate reports based on the images submitted by patients, thus promoting greater efficiency and precision in the process. The methodology involves the development of an artificial intelligence model integrated with an interface that facilitates use by healthcare professionals. The system is expected to contribute to simplifying the diagnosis and monitoring of scoliosis, optimizing physicians' time and improving patients' experience.

Keywords: Scoliosis; Cobb Angle; Artificial Intelligence; Computer Vision; YOLOv8; Medical Diagnosis.

LISTA DE ILUSTRAÇÕES

Figura 1 - Estatísticas do dataset.....	38
Figura 2 - Histograma de contagem de vértebras por imagem.....	39
Figura 3 - Output do dataset.....	40
Figura 4 - Software ImageJ para identificação do ângulo de inclinação da vértebra.....	41
Figura 5 - Arquitetura de Componentes do Sistema.....	52
Figura 6 - Fluxo de Processamento de Dados para Cálculo do Ângulo de Cobb.....	53
Figura 7 - Diagrama de Casos de Uso.....	56
Figura 8 - Modelo de Entidade e Relacionamento.....	61
Figura 9 - Diagrama de Classes - Backend.....	63
Figura 10 - Diagrama de Sequência IA.....	65
Figura 11 - Diagrama de Funções - Frontend.....	67
Figura 12 - Tela de Login.....	71
Figura 13 - Tela inicial (Médico logado).....	72
Figura 14 - Tela de envio de imagem (Médico logado).....	73
Figura 15 - Tela de imagem já selecionada (Médico logado).....	74
Figura 16 - Modal de imagem após análise (Médico logado).....	75
Figura 17 - Relatório do Raio-X analisado (Médico logado).....	76
Figura 18 - Tela de visualização de médicos cadastrados (Administrador logado)...	77
Figura 19 - Tela de visualização de pacientes cadastrados (Médico logado).....	78
Figura 20 - Tela de edição de médicos (Administrador logado).....	79
Figura 21 - Tela de edição de pacientes (Médico logado).....	80
Figura 22 - Modal para autenticação do médico (Administrador logado).....	81
Figura 23 - Confirmação de autenticação de médico (Administrador logado).....	82
Figura 24 - Curva Recall-Confidence do modelo YOLO.....	85
Figura 25 - Curva Precision-Recall do modelo YOLO.....	86
Figura 26 - Curva Precision-Confidence do modelo YOLO.....	87
Figura 27 - Curva Precision-Confidence do modelo YOLO.....	88
Figura 28 - Evolução da Loss por Época.....	89

LISTA DE TABELAS

Tabela 1 - Exemplo de registros do dataset de vértebras anotadas.....	41
Tabela 2 - Relação de Requisitos Funcionais.....	54
Tabela 3 - Requisitos Não Funcionais.....	55
Tabela 4 - Caso de uso referente a Enviar Radiografia.....	57
Tabela 5 - Caso de uso referente a Processar Imagem.....	58
Tabela 6 - Caso de uso referente a Calcular Ângulo de Cobb.....	59
Tabela 7 - Caso de uso referente a Exibir Relatório.....	60
Tabela 8 - Resultados das métricas por época no treinamento do YOLOv8.....	83

LISTA DE ABREVIATURAS E SIGLAS

API	Application Programming Interface
ASGI	Asynchronous Server Gateway Interface
CC BY 4.0	Creative Commons Attribution 4.0 International
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CSV	Comma Separated Value
EIA	Escoliose Idiopática do Adolescente
FAIR	Facebook AI Research
FMI	Fowlkes-Mallows Index
GPU	Graphics Processing Unit
HTTP	Hypertext Transfer Protocol
IA	Inteligência Artificial
ICC	Intraclass Correlation Coefficient
ID	Identificador
iOS	iPhone Operating System
IY	Índice de Youden
JSON	JavaScript Object Notation
MAE	Mean Absolute Error
mAP@50	Mean Average Precision com Intersection over Union de 50%
mAP@50-95	Mean Average Precision com Intersection over Union de 50% a 95%
MCC	Matthews Correlation Coefficient
MER	Modelo de Entidade e Relacionamento
MSE	Mean Squared Error
MSELoss	Mean Squared Error Loss
NumPy	Numerical Python
PDF	Portable Document Format

PIL	Python Imaging Library
RNA	Rede Neural Artificial
SEM	Standard Error of Measurement
SGBD	Sistema Gerenciador de Banco de Dados
YOLO	You Only Look Once
YOLOv8	You Only Look Once versão 8

SUMÁRIO

1 INTRODUÇÃO.....	17
1.1 Motivação.....	17
1.2 Definição do Problema.....	18
1.3 Objetivo Geral.....	18
1.4 Objetivos Específicos.....	19
2 REVISÃO BIBLIOGRÁFICA.....	20
2.1 Validation of an artificial intelligence-based method to automate Cobb angle measurement on spinal radiographs of children with adolescent idiopathic scoliosis.....	20
2.1.1 Introdução e Motivação.....	20
2.1.2 Metodologia.....	20
2.1.3 Resultados e Discussão.....	21
2.1.4 Comparação.....	21
2.2 Avaliação de desvios escolióticos com a utilização de Inteligência Artificial..	22
2.2.1 Introdução e Motivação.....	22
2.2.2 Metodologia.....	22
2.2.3 Resultados e Discussão.....	23
2.2.4 Comparação.....	23
2.3 Modelo computacional baseado em conhecimento para avaliação postural tridimensional.....	24
2.3.1 Introdução e Motivação.....	24
2.3.2 Metodologia.....	25
2.3.3 Resultados e Discussão.....	25
2.3.4 Comparação.....	25
3 METODOLOGIA.....	27
3.1 Inteligência Artificial	27
3.1.1 Inteligência Artificial na Área da Saúde.....	28
3.2 Visão Computacional	29
3.2.1 Pré-processamento de Imagens.....	29

3.2.2 Detecção de Objetos.....	31
3.2.3 Redes Neurais Convolucionais (CNNs).....	32
3.2.3.1 Aplicações da CNN na Medicina.....	33
3.2.3.2 Vantagens das CNNs para Diagnóstico por Imagem.....	34
3.3 Ângulo de Cobb.....	34
3.3.1 Definição e Cálculo do Ângulo de Cobb.....	35
3.3.2 Importância Clínica do Ângulo de Cobb.....	35
3.3.3 Desafios no Cálculo Manual e Necessidade de Automação.....	36
3.4 Datasets.....	37
3.4.1 Dataset de Detecção de Vértex.....	38
3.4.2 Dataset de Ângulo de Inclinação de Vértex.....	40
3.5 Ferramentas Utilizadas.....	42
3.5.1 Python.....	42
3.5.2 Pandas.....	42
3.5.3 NumPy.....	43
3.5.4 Pillow.....	44
3.5.5 PyTorch.....	44
3.5.6 YOLO.....	45
3.5.7 Ultralytics.....	46
3.5.8 FastAPI.....	46
3.5.9 Uvicorn.....	47
3.5.10 JSON.....	48
3.5.11 Node.js.....	48
3.5.12 React Native.....	49
3.5.13 Axios.....	49
3.5.14 Google Colab.....	50
3.5.15 jsPDF.....	51
4 IMPLEMENTAÇÃO.....	52
4.1 Arquitetura do Projeto.....	52

4.1.1 Fluxo de Processamento de Dados.....	53
4.2 Implementação do Software.....	54
4.2.1 Requisitos Funcionais.....	54
4.2.2 Requisitos Não Funcionais.....	54
4.2.3 Diagrama de Casos de Uso.....	55
4.2.3.1 Descrição dos Casos de Uso.....	56
4.3 Modelo de Entidade e Relacionamento.....	60
4.4 Diagramas de Classes.....	62
4.4.1 Backend.....	63
4.4.1.1 Classe CobbPipeline.....	63
4.4.1.2 Classe AngleCNN.....	64
4.4.1.3 Classe CalculoUtil.....	64
4.4.1.4 Classe ImagemUtil.....	64
4.4.1.5 Classe InferenciaUtil.....	65
4.4.1.6 Diagrama de Sequência.....	65
4.4.2 Frontend.....	66
4.4.2.1 Função LoginScreen.....	68
4.4.2.2 Função HomeScreen.....	68
4.4.2.3 Função PatientsScreen.....	68
4.4.2.4 Função PatientRegisterScreen.....	69
4.4.2.5 Função DoctorsScreen.....	69
4.4.2.6 Função DoctorRegisterScreen.....	69
4.4.2.7 Função SendImageScreen.....	70
4.4.2.8 Função ResultModal.....	70
4.5 Funcionamento e Interface.....	70
4.5.1 Tela de Login.....	71
4.5.2 Envio de Imagens.....	72
4.5.3 Geração de Relatórios.....	74
4.5.4 Gerenciamento de Usuários.....	77

5 TESTES E RESULTADOS	83
5.1 Modelo de Detecção de Vértexas	83
5.2 Modelo de Ângulo de Inclinação de Vértexas	88
5.3 API	89
5.3.1 [POST] /api/auth/login	90
5.3.2 [POST] /api/auth/register	90
5.3.3 [DELETE] /api/auth/{user_id}	90
5.3.4 [POST] /api/user	90
5.3.5 [PUT] /api/user/{user_id}	91
5.3.6 [GET] /api/user/{user_id}	91
5.3.7 [GET] /api/user/by_role/{role}	91
5.3.8 [GET] /api/user/patients/by_doctor/{doctor_id}	91
5.3.9 [DELETE] /api/user/{user_id}	92
5.3.10 Considerações Finais	92
6 CONCLUSÃO	93
7 TRABALHOS FUTUROS	94
7.1 Ampliação e Qualificação dos Datasets	94
7.2 Exploração de Novos Modelos de Inteligência Artificial	94
7.3 Previsão da Evolução da Escoliose	95
7.4 Integração Multimodal de Exames	95
7.5 Aprimoramento para Casos de Curvatura em “S”	95
7.6 Implementação de Login Seguro e Controle de Acesso	95
7.7 Fortalecimento da Segurança e Proteção de Dados	96
7.8 Geração e Edição de Relatórios Automatizados	96
7.9 Aprimoramento dos Relacionamentos do Modelo de Dados	96
REFERÊNCIAS	98

1 INTRODUÇÃO

A escoliose é uma condição médica caracterizada por uma curvatura lateral anormal da coluna vertebral, comumente diagnosticada através do ângulo de Cobb, um método amplamente utilizado para medir o grau de desvio (LEHNERT-SCHROTH; AUNER-GRÖBL, 2023). Nos Estados Unidos, estima-se que entre 6 e 9 milhões de pessoas vivam com escoliose, o que representa cerca de 2% a 3% da população (AANS, 2024). A condição pode manifestar-se em diferentes faixas etárias e origens, incluindo as formas idiopática, congênita e neuromuscular, exigindo monitoramento contínuo e intervenções apropriadas para evitar progressões graves.

Apesar dos avanços na medicina, o monitoramento e a avaliação da escoliose ainda apresentam desafios para os profissionais de saúde, em grande parte devido às restrições relacionadas à disponibilidade e eficiência dos recursos utilizados para diagnóstico e monitoramento contínuo da condição (FU et al., 2024). Métodos tradicionais de diagnóstico, como medições manuais do ângulo de Cobb, podem ser demorados, sujeitos a variabilidade entre observadores e dependentes de equipamentos especializados que nem sempre estão disponíveis. Com o aumento da prevalência da escoliose entre adolescentes, conforme indicado por uma revisão publicada no *Journal of Orthopaedic Surgery and Research* (FU et al., 2024), torna-se evidente a necessidade de tecnologias inovadoras para atender às demandas crescentes de diagnóstico e acompanhamento mais precisos.

Nesse contexto, este trabalho propõe o desenvolvimento de uma aplicação baseada em inteligência artificial e visão computacional, utilizando a rede YOLOv8 treinada com um conjunto público de imagens de raio-X para a detecção automática das vértebras. A partir dessa detecção, será possível calcular o ângulo de Cobb de forma mais rápida e padronizada.

1.1 Motivação

A motivação para este estudo está enraizada na busca por soluções tecnológicas que possam revolucionar o diagnóstico e o tratamento da escoliose, melhorando a qualidade de vida dos pacientes e otimizando os processos médicos.

O desenvolvimento de sistemas baseados em inteligência artificial tem demonstrado grande potencial na área da saúde, especialmente em aplicações de diagnóstico por imagem (KHALIFA; ALBADAWY, 2024). No contexto da escoliose, um sistema automatizado para detecção e cálculo do ângulo de Cobb pode proporcionar benefícios significativos, como maior precisão nos diagnósticos e redução do tempo de análise manual. Além disso, essa tecnologia pode ajudar na padronização do diagnóstico, reduzindo variações interobservadores e garantindo maior confiabilidade nos resultados.

A escolha do tema também foi motivada por uma experiência pessoal de um dos integrantes do grupo, que foi diagnosticado com escoliose na adolescência e passou por tratamento especializado. Essa vivência proporcionou uma compreensão mais profunda sobre os desafios enfrentados por pacientes e profissionais de saúde, reforçando a importância de desenvolver ferramentas tecnológicas que possam apoiar e agilizar o processo diagnóstico.

1.2 Definição do Problema

Diante do abordado, este trabalho propõe o desenvolvimento de um protótipo que utiliza visão computacional e IA para calcular automaticamente o ângulo de Cobb em imagens de raio-X, oferecendo uma solução que pode auxiliar profissionais médicos na avaliação da escoliose. A proposta inclui a análise e detecção precisa das curvaturas vertebrais e a geração de relatórios detalhados, promovendo maior eficiência e confiabilidade no processo de diagnóstico.

Ao abordar a integração entre tecnologia e saúde, este estudo busca não apenas contribuir para a área médica, mas também oferecer uma ferramenta que possa impactar diretamente a qualidade de vida dos pacientes. Dessa forma, o projeto se insere no contexto de inovações tecnológicas voltadas para a melhoria dos serviços de saúde, explorando soluções eficazes e acessíveis para desafios clínicos.

1.3 Objetivo Geral

O objetivo geral deste projeto é propor uma metodologia baseada em inteligência artificial para a análise de imagens de raios-X, possibilitando o cálculo automatizado do ângulo de Cobb e a detecção de escoliose de forma mais eficiente.

Através do uso de técnicas de visão computacional e aprendizado de máquina, pretende-se desenvolver uma solução que auxilie profissionais da saúde no diagnóstico e acompanhamento dessa condição, promovendo maior precisão e agilidade no atendimento clínico.

1.4 Objetivos Específicos

Os seguintes objetivos deverão ser verificados:

- Realizar um levantamento bibliográfico sobre o cálculo do ângulo de Cobb, sua aplicação no diagnóstico de escoliose e o uso de inteligência artificial na área médica, com foco em técnicas de visão computacional.
- Pesquisar e selecionar tecnologias adequadas de visão computacional e aprendizado de máquina para a análise automatizada de imagens radiológicas, considerando eficiência e precisão.
- Identificar e utilizar repositórios públicos confiáveis que disponibilizam imagens de raio-X da coluna vertebral anotadas com os valores do ângulo de Cobb, garantindo a qualidade e representatividade dos dados.
- Desenvolver e treinar um modelo de inteligência artificial, utilizando redes neurais convolucionais (CNNs, do inglês *Convolutional Neural Network*), para identificar os pontos de referência necessários ao cálculo do ângulo de Cobb.
- Implementar um protótipo funcional que realize o cálculo automatizado do ângulo de Cobb a partir de imagens de raios-X, integrando-o a uma interface de usuário voltada para médicos.
- Avaliar a eficácia e precisão do protótipo utilizando métricas de desempenho adequadas e validando o sistema com dados de teste.
- Documentar o processo de desenvolvimento, incluindo os métodos utilizados, resultados alcançados e conclusões, para disseminação do trabalho na comunidade acadêmica e médica.

2 REVISÃO BIBLIOGRÁFICA

Esta seção reúne e discute trabalhos relacionados ao tema deste projeto, com foco em estudos que abordam o uso de inteligência artificial e visão computacional na análise de imagens médicas e, em especial, na detecção e avaliação da escoliose. A partir dessa análise comparativa, busca-se evidenciar a relevância da solução proposta neste trabalho e como ela se diferencia das abordagens existentes.

2.1 Validation of an artificial intelligence-based method to automate Cobb angle measurement on spinal radiographs of children with adolescent idiopathic scoliosis

2.1.1 Introdução e Motivação

A escoliose idiopática do adolescente (EIA) é uma deformidade da coluna vertebral que requer medições precisas do ângulo de Cobb para um diagnóstico correto e para a definição do tratamento adequado (PENHA et al., 2018). Tradicionalmente, essa avaliação é realizada manualmente a partir de radiografias, procedimento que é demorado e sujeito a variações inter e intra observador, comprometendo a reprodutibilidade dos resultados (PRESTIGIACOMO et al., 2022). O estudo conduzido por Wong et al. (2023) tem como objetivo validar um algoritmo baseado em inteligência artificial capaz de automatizar a medição do ângulo de Cobb, com o intuito de aprimorar a eficiência, a precisão e a confiabilidade do processo diagnóstico.

2.1.2 Metodologia

No estudo realizado por Wong et al. (2023), foram avaliadas 330 radiografias de pacientes pediátricos com EIA. Deste total, 130 imagens foram utilizadas para o treinamento do modelo de inteligência artificial e as 200 restantes para sua validação. A etapa de validação envolveu a comparação entre os resultados obtidos automaticamente pelo algoritmo e as medições realizadas por um especialista com mais de 20 anos de experiência. A análise de desempenho foi baseada em métricas como o erro padrão de medição (SEM, do inglês *Standard Error of Measurement*), o coeficiente de correlação intraclass (ICC, do inglês *Intraclass Correlation*

Coefficient) e a porcentagem de medições dentro da margem clínica de aceitação ($\pm 5^\circ$), o que permitiu aferir a acurácia e confiabilidade do modelo.

2.1.3 Resultados e Discussão

O estudo realizado por Wong et al. (2023) demonstrou alto desempenho do algoritmo de inteligência artificial desenvolvido para a medição automatizada do ângulo de Cobb. Os resultados revelaram que 91% das medições automatizadas estavam dentro do intervalo clínico aceitável quando comparadas às medições manuais realizadas por especialistas. Além disso, o ICC alcançou 0,92, indicando excelente concordância, e o tempo médio de processamento por imagem foi de apenas 18 segundos. Esses dados evidenciam a precisão, confiabilidade e eficiência do método proposto, validando o uso de IA como uma ferramenta promissora para auxiliar no diagnóstico e monitoramento da escoliose.

2.1.4 Comparação

O estudo de Wong et al. (2023) validou clinicamente um algoritmo de inteligência artificial voltado à medição automatizada do ângulo de Cobb em radiografias de pacientes com EIA. Já o presente trabalho, embora inspirado na metodologia proposta por Wong et al. (2023), adota uma abordagem voltada à automação integral do processo, englobando desde a detecção das vértebras até a estimativa do ângulo de inclinação, utilizando dois modelos de aprendizado profundo complementares: um modelo YOLOv8 para detecção automática das vértebras em radiografias de coluna e uma CNN para regressão do ângulo de inclinação de cada vértebra individualmente.

No que diz respeito ao desempenho, o modelo de detecção (YOLOv8) demonstrou alta capacidade de identificar corretamente as vértebras, mesmo em imagens com variações anatômicas significativas. Já o modelo de regressão (CNN) obteve uma perda final de 38,91 e um erro médio absoluto de $5,65^\circ$, valores que se encontram dentro da faixa clínica de aceitação relatada por Wong et al. (2023), demonstrando consistência e precisão na estimativa dos ângulos de inclinação vertebral.

Enquanto o trabalho de Wong et al. (2023) concentra-se na validação clínica e estatística de um modelo único, o projeto atual destaca-se por integrar diferentes etapas do processo diagnóstico em um sistema unificado, com foco na aplicabilidade prática e acessibilidade tecnológica. A solução proposta busca automatizar completamente o fluxo de análise, desde o envio da imagem pelo profissional até a geração do relatório com o cálculo do ângulo de Cobb, reduzindo a necessidade de intervenção manual e o tempo de análise.

Dessa forma, os dois trabalhos se complementam: o de Wong et al. (2023) contribui com a base metodológica e evidências clínicas de validação, enquanto o presente estudo expande esses conceitos para um contexto de implementação prática, explorando a viabilidade técnica e o potencial de integração da inteligência artificial em sistemas de apoio ao diagnóstico da escoliose.

2.2 Avaliação de desvios escolióticos com a utilização de Inteligência Artificial

2.2.1 Introdução e Motivação

Oliveira et al. (2022) propuseram o uso de CNNs para detectar desvios escolióticos patológicos na coluna lombar em radiografias. O objetivo foi demonstrar que modelos de *deep learning* podem oferecer diagnósticos mais rápidos e consistentes, minimizando erros humanos e facilitando intervenções precoces. O estudo utilizou uma base de dados rotulada por especialistas e obteve resultados promissores em sensibilidade, especificidade e acurácia. Além disso, a proposta destaca o potencial da IA como ferramenta complementar ao olhar clínico, especialmente em regiões com acesso limitado a profissionais experientes ou infraestrutura diagnóstica avançada.

2.2.2 Metodologia

O estudo desenvolvido por Oliveira *et al.* (2022) foi classificado como uma pesquisa descritiva de caráter pré-diagnóstico. A metodologia envolveu a criação e o treinamento de uma CNN utilizando linguagem Python e bibliotecas específicas. Três diferentes cenários de arquitetura foram projetados: o primeiro baseado em um modelo de referência; o segundo com aprofundamento das conexões inter e intra

camadas; e o terceiro com a adição de camadas de transposição de matrizes. O banco de dados inicial consistia em 2.897 imagens, totalizando ao final mais de 579 mil imagens derivadas, utilizadas para treinamento e teste do algoritmo.

2.2.3 Resultados e Discussão

Os resultados obtidos indicaram uma performance robusta da CNN, especialmente no terceiro cenário, que apresentou os melhores indicadores de eficiência: acurácia de 96%, precisão de 98%, sensibilidade de 91% e especificidade de 99%. Além disso, foram calculados índices como o de *Fowlkes-Mallows* (FMI, do inglês *Fowlkes-Mallows Index*) de 0,934, o coeficiente de correlação de Matthews (MCC, do inglês *Matthews Correlation Coefficient*) de 0,894 e índice de Youden (IY) de 0,877, que confirmam a confiabilidade do modelo. Esses resultados sugerem que o algoritmo é altamente eficaz na detecção de desvios escolióticos na coluna lombar.

2.2.4 Comparação

A análise comparativa entre os estudos evidencia que, embora tanto Oliveira et al. (2022) quanto o presente trabalho utilizam inteligência artificial aplicada à área da saúde, seus objetivos e abordagens metodológicas diferem significativamente.

Enquanto Oliveira et al. (2022) desenvolveram uma CNN voltada à detecção de desvios escolióticos patológicos em radiografias da coluna lombar, com foco no reconhecimento da presença ou ausência da deformidade, o presente estudo propõe uma solução mais específica e quantitativa, ao automatizar a medição do ângulo de Cobb, parâmetro clínico fundamental para o diagnóstico e o acompanhamento da gravidade da escoliose.

O modelo de Oliveira et al. (2022) apresentou alta eficiência na classificação binária de desvios escolióticos, demonstrando resultados consistentes em termos de acurácia, precisão, sensibilidade e especificidade. De maneira semelhante, o sistema proposto neste trabalho mostrou desempenho robusto na detecção de vértebras, evidenciando excelente capacidade de localização e identificação das estruturas ósseas necessárias para o cálculo do ângulo de Cobb.

Além da etapa de detecção, este estudo também incorpora uma CNN de regressão responsável pela inferência do ângulo de Cobb a partir das vértebras identificadas. O modelo atingiu uma *loss* final de 38,91%, com variação média de 5,65° em relação aos valores reais de referência. Esses resultados demonstram que, embora o modelo ainda apresente margem de erro natural a aplicações em imagens médicas, ele é capaz de reproduzir com boa precisão a medição clínica, viabilizando seu uso como ferramenta auxiliar de apoio ao diagnóstico.

Outra distinção relevante está na aplicação prática. Enquanto o estudo de Oliveira et al. (2022) se concentra no desenvolvimento do algoritmo de detecção, este projeto integra inteligência artificial e visão computacional em um sistema completo, no qual o médico pode enviar a radiografia e receber automaticamente a imagem anotada com as vértebras detectadas e o ângulo calculado. Essa integração de módulos de detecção, regressão e visualização amplia o escopo da proposta, transformando-a em uma solução automatizada e acessível ao contexto clínico real, com potencial de otimizar o tempo de análise e reduzir a subjetividade da avaliação médica.

2.3 Modelo computacional baseado em conhecimento para avaliação postural tridimensional

2.3.1 Introdução e Motivação

Sant'Ana (2017) destaca a relevância da caracterização postural para um diagnóstico cinético funcional preciso, permitindo a identificação de alterações posturais e desequilíbrios musculares essenciais para o desenvolvimento de programas de reabilitação eficazes. Apesar do avanço tecnológico e da existência de programas computacionais que realizam análises tridimensionais da coluna vertebral a partir de imagens bidimensionais, esses métodos ainda apresentam limitações, inclusive nos sistemas mais recentes baseados em captura tridimensional (KUMAR et al., 2024). Dessa forma, torna-se necessário aprimorar as técnicas para avaliação postural, corrigindo suas limitações e melhorando a identificação de assimetrias, desvios na coluna e alterações articulares periféricas.

2.3.2 Metodologia

O estudo é uma pesquisa metodológica de desenvolvimento que teve como objetivo geral propor um modelo computacional baseado em conhecimento para avaliação postural tridimensional. Para isso, foram desenvolvidos e aperfeiçoados algoritmos para análise postural fundamentados em coordenadas cartesianas tridimensionais. O modelo computacional criado visa minimizar as limitações dos métodos existentes. A validação do modelo ocorreu por meio de um estudo piloto que avaliou sua aplicabilidade e desempenho.

2.3.3 Resultados e Discussão

Os resultados evidenciaram que o modelo computacional proposto por Sant'Ana (2017) supera diversas limitações das ferramentas atuais de avaliação biofotogramétrica. O sistema possibilita uma análise tridimensional mais detalhada e precisa das alterações posturais e desvios na coluna vertebral, contribuindo para um diagnóstico mais completo e fundamentado que pode apoiar programas de reabilitação personalizados.

2.3.4 Comparação

Ao comparar o presente trabalho com o de Sant'Ana (2017), observa-se que ambos compartilham o propósito de aprimorar a avaliação postural por meio de recursos computacionais, porém diferem quanto à finalidade e à natureza da análise.

O modelo proposto por Sant'Ana (2017) baseia-se em coordenadas tridimensionais para uma avaliação global da postura corporal, considerando variáveis musculares, articulares e de equilíbrio biomecânico. Seu foco está em análises tridimensionais exploratórias voltadas à reabilitação e prevenção de desequilíbrios posturais, o que o caracteriza como um sistema abrangente e preventivo.

Por outro lado, o presente estudo se dedica a um problema clínico específico, a escoliose, aplicando técnicas de visão computacional e aprendizado profundo em radiografias bidimensionais para automatizar o cálculo do ângulo de Cobb. Com Precisão de aproximadamente 92% e *recall* de 94% na detecção de vértebras, além

de uma variação média de $5,65^\circ$ na inferência do ângulo, o sistema demonstra potencial para suporte objetivo e quantitativo ao diagnóstico médico.

Enquanto Sant'Ana (2017) oferece uma abordagem tridimensional voltada à análise biomecânica e reabilitação, o presente trabalho foca na eficiência diagnóstica e na redução da subjetividade em avaliações radiográficas. Assim, ambos os estudos se complementam no campo da avaliação postural: o primeiro contribui com a compreensão cinético-funcional global do corpo, e o segundo com a automatização precisa de um indicador clínico essencial, consolidando a importância da inteligência artificial como ferramenta de apoio em diferentes dimensões da prática fisioterapêutica e médica.

3 METODOLOGIA

A metodologia adotada para o desenvolvimento deste trabalho combina fundamentos teóricos e práticos, de modo a estruturar as etapas necessárias para a construção da aplicação proposta. Inicialmente, são apresentados conceitos essenciais relacionados à inteligência artificial, visão computacional e ao ângulo de Cobb, que servem de base para a compreensão das técnicas aplicadas. Em seguida, detalha-se o conjunto de dados utilizado para treinamento e validação, bem como as ferramentas empregadas na implementação.

3.1 Inteligência Artificial

A Inteligência Artificial é um campo que busca desenvolver sistemas capazes de realizar tarefas onde que tradicionalmente, exigem a intervenção humana, como o raciocínio, aprendizado, reconhecimento de padrões, compreensão de linguagem natural e a tomada de decisões (STRYKER, C.; KAVLAKOGLU, E., 2024). Trata-se de uma área interdisciplinar, onde combina técnicas de computação com métodos estatísticos e análise de dados, além de incorporar conhecimentos de engenharia, linguística, neurociência, filosofia e psicologia. Essa integração permite que a IA não apenas automatize processos complexos, mas também contribua para o avanço tecnológico em diversos setores, incluindo saúde, indústria, educação e transporte (FREITAS et al., 2023).

No presente projeto, a inteligência artificial é aplicada na análise de imagens de raio-X, com o objetivo de detectar de forma precisa e automatizada alterações na coluna vertebral, como a escoliose, caracterizada pela curvatura anormal dessa estrutura óssea (VARELLA, 2017). Além disso, também foi utilizada para correção e revisão dos textos apresentados.

Para alcançar esse objetivo, o projeto utiliza algoritmos de aprendizado profundo capazes de identificar padrões específicos em imagens de raio-X da coluna vertebral. O modelo é treinado para detectar individualmente as vértebras e calcular o ângulo de Cobb, que representa o grau de inclinação das vértebras e constitui o padrão-ouro para o diagnóstico e acompanhamento da escoliose (NEGRINI et al., 2018). Dessa forma, a aplicação da inteligência artificial não apenas automatiza o

processo diagnóstico, mas também oferece suporte aos profissionais da saúde, garantindo maior eficiência e agilidade na análise dos exames.

3.1.1 Inteligência Artificial na Área da Saúde

Atualmente, a presença da IA é cada vez mais perceptível no cotidiano, desde *smartphones* até suporte educacional (COSTA, 2025). No campo da saúde, seu uso tem se expandido rapidamente, abrangendo desde diagnósticos simples até a personalização de tratamentos médicos (CTC TECH, 2025).

A inteligência artificial também está sendo utilizada em procedimentos cirúrgicos, auxiliando na busca por maior precisão e reduzindo significativamente o risco de erros (DAVENPORT; KALAKOTA, 2019). Na área da medicina, o uso da IA revela-se especialmente relevante, uma vez que a tecnologia pode ser aplicada por meio de redes neurais artificiais (RNAs), as quais otimizam a análise de diagnósticos clínicos mais simples e rotineiros, permitindo que os profissionais de saúde concentrem-se em casos mais complexos e no atendimento personalizado ao paciente, promovendo maior eficiência e precisão nos processos clínicos (AHUJA, 2019).

Com a utilização do *deep learning*, os sistemas de inteligência artificial alcançam níveis de precisão superiores à capacidade humana na análise de imagens médicas, sendo capazes de identificar padrões sutis que possibilitam a detecção precoce de doenças com maior exatidão (SANTOS et al., 2019). Considerando que referidos sistemas aprendem de forma autônoma a partir de grandes volumes de dados e laudos clínicos previamente analisados, tornam-se aptos a automatizar tarefas repetitivas, como a geração de laudos e a interpretação de exames de imagem, trazendo agilidade significativa aos processos diagnósticos e liberando os profissionais da saúde para decisões clínicas mais complexas e estratégicas (TELEMEDICINA, 2025).

Insta salientar que a inteligência artificial não possui como objetivo substituir o médico, mas atuar como uma ferramenta de apoio, oferecendo uma “segunda opinião” objetiva e baseada em dados (TELEMEDICINA, 2025). Dessa forma,

contribui para a escolha de tratamentos mais adequados e personalizados para cada paciente, ampliando a precisão e a segurança nas decisões clínicas.

3.2 Visão Computacional

A Visão Computacional é uma subárea da Inteligência Artificial, onde capacita os sistemas computacionais a extrair, analisar e interpretar informações provenientes de imagens digitais (SOUZA et al., 2024). No contexto deste projeto, essa tecnologia é aplicada ao processamento automatizado de radiografias da coluna vertebral, com o objetivo de apoiar a análise da escoliose.

A abordagem adotada utiliza modelos de *deep learning* para detectar individualmente as vértebras presentes nas imagens e então inferir o ângulo de inclinação de cada uma delas. A partir das vértebras identificadas, o sistema seleciona aquelas com maior inclinação para o cálculo do ângulo de Cobb. Essa estratégia, baseada no uso de modelos de detecção de objetos como o YOLOv8, permite uma análise precisa da curvatura vertebral. Em seguida, são traçadas linhas de referência sobre as vértebras detectadas, possibilitando o cálculo automático do ângulo de Cobb e fornecendo aos profissionais de saúde uma ferramenta confiável para avaliação do grau de escoliose.

3.2.1 Pré-processamento de Imagens

O pré-processamento de imagens corresponde ao conjunto de técnicas aplicadas a imagens digitais antes de sua utilização em modelos de visão computacional. Seu principal objetivo é padronizar os dados de entrada, reduzir ruídos e facilitar a extração de características relevantes pelos algoritmos de aprendizado de máquina. De acordo com Goodfellow, Bengio e Courville (2016), esse processo é essencial para aumentar a robustez dos modelos e estabilizar o treinamento das redes neurais, garantindo que a variabilidade dos dados não prejudique o desempenho do sistema.

Entre as técnicas que mais se utiliza, destaca-se a conversão para tons de cinza ou a normalização de cores, que acabam simplificando a informação visual, reduzindo a dimensão dos dados sem que haja perda significativa do conteúdo

relevante. O redimensionamento das imagens visto que é necessário que sejam iguais, pois as CNNs exigem entradas de tamanho fixo tornando o processo uniforme (GOODFELLOW; BENGIO; COURVILLE, 2016). A normalização dos valores de *pixel*, geralmente em intervalos como 0 e 1, contribui para estabilizar o aprendizado e evitar que diferenças de escala entre imagens causam instabilidades no treinamento (YUAN et al., 2019).

Outras abordagens que podemos incluir são a remoção de ruídos e o aumento de contraste, aplicáveis especialmente quando as imagens apresentam uma baixa qualidade ou iluminação o que acaba dificultando a interpretação. Adicionalmente, técnicas de *data augmentation* como rotação, corte, espelhamento e variação de brilho são amplamente utilizadas, principalmente em contextos médicos, onde os conjuntos de dados são limitados devido a restrições éticas e de disponibilidade. Essa estratégia expande artificialmente o conjunto de treinamento, aumentando a capacidade de generalização do modelo (SHORTEN; KHOSHGOFTAAR, 2019).

No contexto das imagens médicas, o pré-processamento assume papel particularmente relevante. Exames radiográficos, por exemplo, podem apresentar diferenças significativas em qualidade de captura, iluminação, contraste e padronização entre diferentes equipamentos. Conforme apontado por Litjens et al. (2017), a aplicação de técnicas de pré-processamento contribui para a padronização desses exames, possibilitando maior consistência na análise automatizada e melhorando a precisão dos modelos de detecção de anomalias, como no caso da escoliose. Dessa forma, o pré-processamento constitui etapa indispensável para garantir a confiabilidade do sistema proposto neste trabalho.

No desenvolvimento do modelo, foram aplicadas diferentes técnicas de pré-processamento voltadas especificamente para radiografias da coluna. Essas etapas mostraram-se essenciais para padronizar os dados de entrada e possibilitar a utilização eficiente pelos modelos de detecção e regressão.

- **Redimensionamento e normalização:** todas as imagens foram redimensionadas para a dimensão de entrada exigida pelas redes neurais. Em seguida, os valores dos pixels foram normalizados no intervalo

$[0,1][0,1][0,1]$, o que contribuiu para estabilizar e acelerar o processo de aprendizado.

- **Conversão para tons de cinza:** considerando que as radiografias não demandam informação de cor, as imagens foram convertidas para escala de cinza, reduzindo a dimensionalidade dos dados sem perda de informação relevante.

Data augmentation no YOLOv8: foram aplicadas técnicas de aumento de dados, como rotações, inversões e variações de brilho e contraste, com o objetivo de aumentar a robustez do modelo de detecção de vértebras diante de diferenças de posicionamento ou iluminação entre exames.

- **Organização dos dados:** para o modelo de regressão responsável pelo cálculo do ângulo de Cobb, os dados foram estruturados em arquivos CSV (*Comma Separated Values*), vinculando cada imagem ao ângulo anotado da vértebra correspondente. Essa organização possibilitou maior eficiência no treinamento e validação do modelo.

3.2.2 Detecção de Objetos

A detecção de objetos em visão computacional constitui uma das tarefas mais relevantes dentro da área de inteligência artificial aplicada ao processamento de imagens (GONZALEZ; WOODS, 2018). Esse processo tem como objetivo identificar, localizar e classificar automaticamente elementos de interesse em uma cena, geralmente por meio da utilização de caixas delimitadoras (*bounding boxes*) e rótulos que indicam a categoria de cada objeto (BROWNLEE, 2019). Em comparação com métodos tradicionais de reconhecimento, a detecção não apenas verifica a presença de um objeto, mas também determina sua posição espacial na imagem, o que permite análises mais detalhadas e contextualizadas (REDMON et al., 2016).

No âmbito da saúde, a detecção de objetos apresenta relevância significativa, sobretudo em exames radiológicos, como radiografias, tomografias e ressonâncias magnéticas. A automatização desse processo possibilita a identificação de órgãos, ossos e possíveis anomalias, o que contribui para diagnósticos mais rápidos, padronizados e com menor margem de erro humano (ESTEVA et al., 2019).

Conforme destacado por Litjens et al. (2017), os avanços em aprendizado profundo ampliaram a capacidade de interpretação automática de imagens médicas, oferecendo suporte efetivo aos profissionais da saúde na detecção precoce de alterações anatômicas.

Nas últimas décadas, observou-se um progresso significativo nessa área, principalmente com o surgimento de modelos de detecção baseados em CNNs. Entre as abordagens mais proeminentes, destaca-se a família de modelos *You Only Look Once* (YOLO), que introduziu uma estratégia inovadora ao unificar em uma única etapa a localização e a classificação de objetos, conciliando rapidez e precisão. Essa característica torna os modelos YOLO especialmente adequados para aplicações em tempo real, como no auxílio ao diagnóstico médico, em que a agilidade e a confiabilidade são aspectos críticos (BOCHKOVSKIY et al., 2020).

No presente projeto, a detecção de objetos foi implementada por meio do modelo YOLOv8, selecionado em virtude de sua rapidez e alta precisão, fatores especialmente relevantes em aplicações médicas. O foco dessa etapa consistiu na localização automática das vértebras em imagens radiográficas, viabilizando a identificação das estruturas essenciais para o cálculo do ângulo de Cobb.

Para o treinamento do modelo, as radiografias foram anotadas no Roboflow, atribuindo-se rótulos às vértebras de interesse. Essas anotações constituíram o conjunto de dados supervisionado utilizado pelo YOLOv8, que, após o treinamento, demonstrou capacidade de realizar a inferência automática sobre novas radiografias, localizando as vértebras com confiabilidade satisfatória. Assim, a detecção de vértebras desempenhou papel fundamental no *pipeline* do projeto, pois estabeleceu a transição entre as imagens radiográficas originais e a etapa de extração de características, permitindo a automatização da avaliação clínica.

3.2.3 Redes Neurais Convolucionais (CNNs)

As CNNs estão entre as arquiteturas mais aplicadas na inteligência artificial contemporânea, sobretudo em tarefas em que envolvem o reconhecimento e o aprendizado de padrões visuais (SOUZA et al., 2024). Essas redes são constituídas por camadas convolucionais, em que aplicam filtros para que possa se extrair características específicas como locais das imagens, bordas, texturas e formas

específicas, seguidas por camadas de *pooling*, que reduzem a dimensionalidade e destacam as informações mais relevantes. As camadas totalmente conectadas integram essas informações, permitindo que a rede realize tarefas de classificação, regressão ou tomada de decisão (SOUZA et al., 2024).

No presente projeto, as CNNs foram empregadas em duas etapas complementares do *pipeline* de processamento de imagens. Primeiramente, utilizou-se a arquitetura YOLOv8, responsável pela detecção automática das vértebras nas radiografias, identificando suas posições e extraindo informações estruturais relevantes para o cálculo do ângulo de Cobb. Em seguida, foi desenvolvida uma rede CNN de regressão, projetada para estimar o ângulo de inclinação individual de cada vértebra detectada, a partir das características visuais extraídas dessas regiões.

Por meio desse arranjo, a combinação entre uma CNN de detecção e uma CNN de regressão permitiu que o sistema realizasse tanto a localização anatômica das vértebras quanto a inferência quantitativa dos ângulos necessários para o cálculo do ângulo de Cobb. Dessa forma, as CNNs não apenas classificam ou segmentam imagens, mas também fornecem representações detalhadas que sustentam medições médicas essenciais, contribuindo para análises mais objetivas e precisas na prática clínica (LITJENS et al., 2017; ALZUBAIDI et al., 2021).

3.2.3.1 Aplicações da CNN na Medicina

O avanço da inteligência artificial tem impactado significativamente diversos setores, incluindo a área da saúde, e as CNNs destacam-se dentre as tecnologias emergentes mais relevantes nesse contexto.

As CNNs demonstram uma elevada eficácia na análise de imagens médicas, como radiografias, tomografias computadorizadas e imagens de ressonância magnética (COSTA, 2017). A incorporação dessas redes em sistemas de apoio ao diagnóstico médico permite a detecção automatizada de patologias, contribui para a triagem clínica e promove maior agilidade e precisão nos fluxos de trabalho hospitalares (MALEKI VARNOSFADERANI; FOROUZANFAR, 2024).

Diversos estudos demonstram casos de sucesso da aplicação de CNNs na medicina. Por exemplo, no diagnóstico por imagem de câncer, as CNNs têm se mostrado capazes de identificar lesões tumorais e indícios de neoplasias com maior rapidez e precisão do que análises exclusivamente humanas, reduzindo o tempo necessário para o diagnóstico e aumentando as chances de detecção precoce (ESTEVA et al., 2017). Outro exemplo é o uso de redes neurais convolucionais tridimensionais para análise de exames de ressonância magnética cerebral, como na Doença de Alzheimer, em que modelos multimodais 3D alcançaram alta precisão na classificação e monitoramento da progressão da doença (FU et al., 2024).

3.2.3.2 Vantagens das CNNs para Diagnóstico por Imagem

- **Precisão elevada:** Quando treinadas com *datasets* representativos, as CNNs atingem acurácia comparável ou superior à de especialistas humanos, garantindo resultados confiáveis em diagnósticos médicos (ESTEVA et al., 2017).
- **Automação da análise de imagens:** Reduz o trabalho manual, permitindo que profissionais da saúde concentrem atenção em casos complexos ou urgentes, incluindo radiografias e outras modalidades de imagem (COSTA, 2017).
- **Aprendizado contínuo e adaptabilidade:** O ajuste dinâmico de pesos e parâmetros permite que o modelo melhore com novos dados e se adapte a diferentes tipos de exame, mantendo a eficácia mesmo em contextos variados (ALZUBAIDI et al., 2021).
- **Redução de erros humanos:** Atuação como segunda opinião, aumentando a segurança do paciente e a confiabilidade do diagnóstico (LITJENS et al., 2017).

3.3 Ângulo de Cobb

O ângulo de Cobb é a principal medida quantitativa utilizada para avaliar a curvatura anormal da coluna vertebral em casos de escoliose. Seu cálculo baseia-se na análise de radiografias em que se traçam linhas sobre as vértebras mais inclinadas da curvatura (JONES; THUAIMER, 2013). A interseção entre essas linhas, ou entre suas perpendiculares, define o valor do ângulo. Desenvolvido por

John Robert Cobb em 1948, esse método se tornou a referência padrão em diagnósticos clínicos por sua simplicidade e eficiência na representação do desvio lateral da coluna (COBB, 1948).

Além de ser amplamente adotado na área médica, o ângulo de Cobb tem ganhado espaço em aplicações computacionais voltadas à saúde, pois oferece um parâmetro claro e objetivo que pode ser extraído por sistemas automatizados (FU et al., 2024). Isso o torna especialmente relevante no contexto de desenvolvimento de soluções baseadas em visão computacional, como a proposta deste trabalho, em que o objetivo é detectar e calcular automaticamente esse ângulo com o auxílio de inteligência artificial.

3.3.1 Definição e Cálculo do Ângulo de Cobb

O ângulo de Cobb é a medida mais utilizada para quantificar a curvatura lateral da coluna vertebral em casos de escoliose. Desde que foi proposto em 1948, tornou-se o padrão de referência mundial para essa avaliação (COBB, 1948).

O cálculo desse ângulo é feito com base em imagens de raio-X da coluna. O procedimento envolve a identificação da vértebra mais inclinada no início da curvatura e da vértebra mais inclinada ao final. A partir dessas duas vértebras, traçam-se linhas paralelas às suas placas terminais superiores e inferiores, respectivamente. Em seguida, constrói-se o ângulo entre as perpendiculares a essas linhas, sendo este o valor do ângulo de Cobb (MORRISSY et al., 1990).

O método permanece amplamente utilizado não apenas por sua eficácia, mas também por sua simplicidade técnica, permitindo que médicos realizem medições diretas em exames radiográficos com instrumentos básicos (ASHER, 2024). Sua padronização ao longo das décadas é um dos fatores que o tornam tão prevalente em protocolos de diagnóstico e acompanhamento da escoliose.

3.3.2 Importância Clínica do Ângulo de Cobb

O ângulo de Cobb é o principal critério utilizado na avaliação da gravidade da escoliose e na tomada de decisões clínicas quanto ao tratamento (ASHER, 2024). Ele fornece uma medida padronizada da curvatura lateral da coluna vertebral, sendo

amplamente adotado por ortopedistas e radiologistas para classificar a escoliose em graus leve, moderado ou severo. De acordo com a classificação tradicional, curvaturas inferiores a 10 graus geralmente não configuram escoliose clínica; de 10 a 25 graus são consideradas leves; de 25 a 45 graus, moderadas; e acima de 45 graus, severas, podendo indicar a necessidade de cirurgia (NEGRINI et al., 2018).

Essa classificação influencia diretamente a conduta médica. Em casos leves, a abordagem pode ser apenas o acompanhamento periódico com exames de imagem. Já nas formas moderadas, pode ser indicado o uso de coletes ortopédicos com o objetivo de impedir a progressão da curvatura. Para curvaturas severas, especialmente em pacientes em fase de crescimento, intervenções cirúrgicas são frequentemente recomendadas para evitar complicações respiratórias e posturais (KRÖBER et al., 2019).

Além disso, o ângulo de Cobb é utilizado para monitorar a evolução do quadro ao longo do tempo, permitindo comparar exames consecutivos e avaliar a resposta ao tratamento. Sua padronização e fácil aplicabilidade fazem com que seja, até hoje, a ferramenta mais confiável para avaliação clínica da escoliose (TEIXEIRA, 2023).

3.3.3 Desafios no Cálculo Manual e Necessidade de Automação

Apesar de ser amplamente utilizado na prática clínica, o cálculo manual do ângulo de Cobb apresenta diversos desafios que impactam sua precisão e reprodutibilidade (SPERANDIO; TAVARES; SANTOS, 2011). Um dos principais problemas é a variabilidade entre observadores, ou seja, diferentes profissionais podem obter valores distintos ao medir o mesmo raio-X. Isso ocorre devido à subjetividade envolvida na seleção das vértebras mais inclinadas da curvatura, que são essenciais para a construção das linhas de referência do método (PRUIJS et al., 1994).

Além disso, fatores como a qualidade da imagem, a posição do paciente no momento do exame e até o nível de experiência do avaliador influenciam diretamente no resultado. Estudos indicam que a diferença entre medições feitas por diferentes especialistas pode chegar a até 7 graus, o que é clinicamente

significativo, principalmente em casos limítrofes entre diferentes classificações de gravidade (MORRISSY et al., 1990).

Nesse contexto, a automação do cálculo do ângulo de Cobb torna-se uma necessidade crescente. O uso de tecnologias como visão computacional e inteligência artificial têm demonstrado grande potencial para reduzir a subjetividade e aumentar a confiabilidade dos diagnósticos (FU et al., 2024). Modelos baseados em CNNs têm se destacado pela capacidade de identificar automaticamente as vértebras envolvidas na curvatura e calcular com precisão o ângulo correspondente (FU et al., 2024).

O desenvolvimento de soluções automatizadas contribui para a padronização dos resultados, redução do tempo de análise e menor dependência da experiência do profissional. Nesse sentido, o presente trabalho propõe uma aplicação que utiliza técnicas de visão computacional integradas a modelos de inteligência artificial para realizar esse cálculo de forma automatizada, com o intuito de oferecer maior precisão e agilidade no apoio ao diagnóstico da escoliose.

3.4 Datasets

O conjunto de dados utilizado neste trabalho foi obtido a partir de um repositório público de imagens médicas, o *Mendeley Data* (FRAIWAN; AUDAT; MANASREH, 2022). A utilização de repositórios públicos é especialmente relevante em pesquisas com dados médicos, uma vez que esses dados são sensíveis e seu acesso direto é restrito devido a questões éticas e legais. O uso de *datasets* públicos permite que pesquisas sejam conduzidas de maneira ética, segura e reprodutível, garantindo transparência nos resultados.

Além disso, a utilização de imagens de um repositório público contribui para a comparabilidade de resultados com estudos similares, permitindo que futuras pesquisas avaliem a performance de diferentes modelos de inteligência artificial em dados consistentes e acessíveis à comunidade científica.

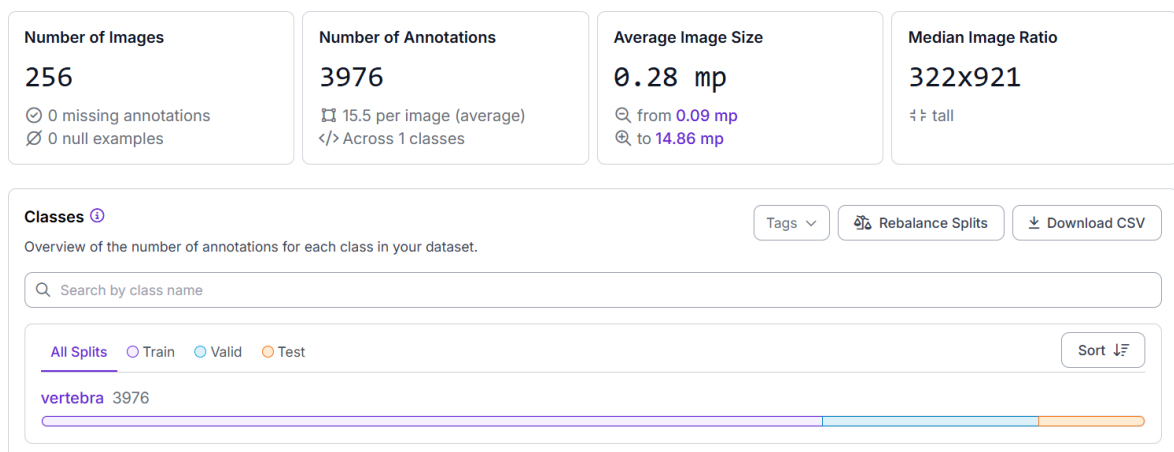
O *dataset* está licenciado sob CC BY 4.0 (*Creative Commons Attribution 4.0 International*), possibilitando seu uso para fins acadêmicos e comerciais, desde que seja devidamente citada a fonte original.

No contexto deste trabalho, foram construídos dois subconjuntos distintos a partir do *dataset* original, cada um destinado a uma etapa do *pipeline*: O primeiro subconjunto, preparado e anotado no Roboflow, foi utilizado para treinar a rede YOLOv8 na tarefa de detecção automática de vértebras, já o segundo subconjunto foi criado a partir dos recortes das vértebras detectadas e posteriormente anotado com o ângulo de inclinação de cada vértebra, servindo como base para o treinamento de uma rede CNN.

3.4.1 Dataset de Detecção de Vértebras

Para a rotulagem das imagens, foi utilizada a ferramenta Roboflow, na qual as vértebras presentes nas radiografias foram marcadas manualmente pelo grupo. O *dataset* final contou com 256 imagens de raio-X, sendo 185 de pacientes com escoliose e 71 de pacientes sem a condição. Como o objetivo do modelo era a identificação individual das vértebras, a única classe estabelecida foi denominada “vértebra”. Ao todo, foram anotadas 3.976 vértebras, resultando em uma média de 15,5 vértebras por imagem, conforme ilustrado nas estatísticas disponibilizadas pela ferramenta e ilustrado na Figura 1.

Figura 1 - Estatísticas do dataset

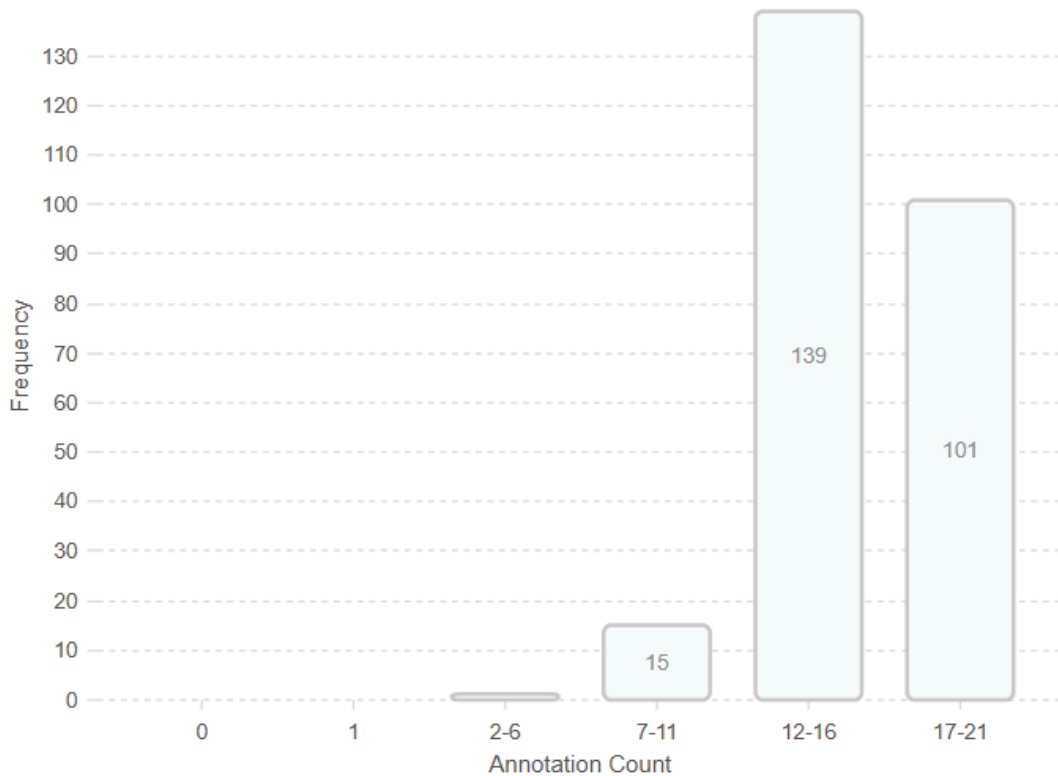


Fonte: Autores, 2025.

A Figura 2 apresenta a distribuição da quantidade de vértebras por imagem, evidenciando que a maior parte das radiografias concentrou-se entre 12 e 16 vértebras (139 imagens) e entre 17 e 21 vértebras (101 imagens). Observa-se ainda um número reduzido de radiografias com 7 a 11 vértebras (15 imagens) e casos

esporádicos com apenas 2 a 6 vértebras. Esses dados demonstram que, embora o conjunto seja relativamente limitado em tamanho, ele possui consistência e qualidade suficientes para treinar modelos de detecção, visto que a grande maioria das imagens contempla uma quantidade adequada de vértebras anotadas.

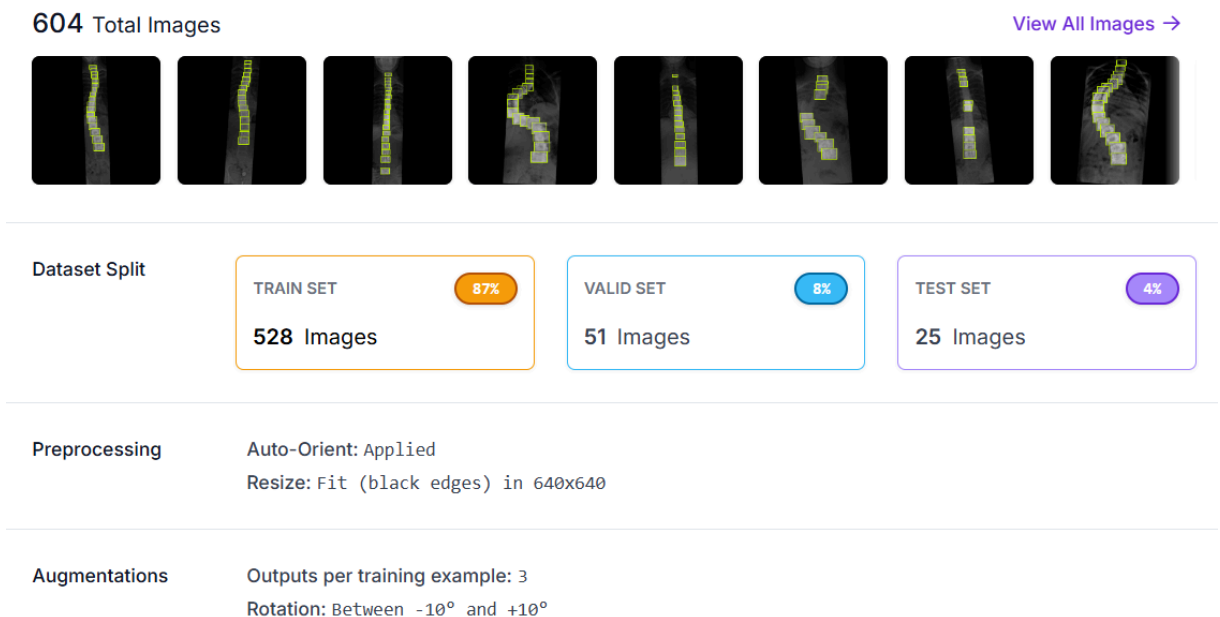
Figura 2 - Histograma de contagem de vértebras por imagem



Fonte: Autores, 2025.

Após a etapa de anotação, o *dataset* foi exportado diretamente pelo Roboflow, conforme ilustrado na Figura 3, já configurado para o treinamento no modelo YOLOv8. No pré-processamento, foi aplicada a auto-orientação das imagens e o redimensionamento para 640x640 *pixels*, utilizando o método *fit*, que adiciona bordas pretas quando necessário, garantindo padronização no tamanho de entrada da rede. Além disso, foram empregadas técnicas de *data augmentation*, com a geração de três variações para cada imagem original, incluindo rotações entre -10° e $+10^\circ$.

Figura 3 - Output do dataset



Fonte: Autores, 2025.

Com essas configurações, o conjunto final passou a conter 604 imagens, que foram automaticamente divididas em três subconjuntos: treinamento, validação e teste, seguindo a proporção sugerida pela ferramenta. Essa preparação assegurou que o *dataset* estivesse padronizado e diversificado o suficiente para melhorar a robustez do modelo durante o treinamento e avaliação.

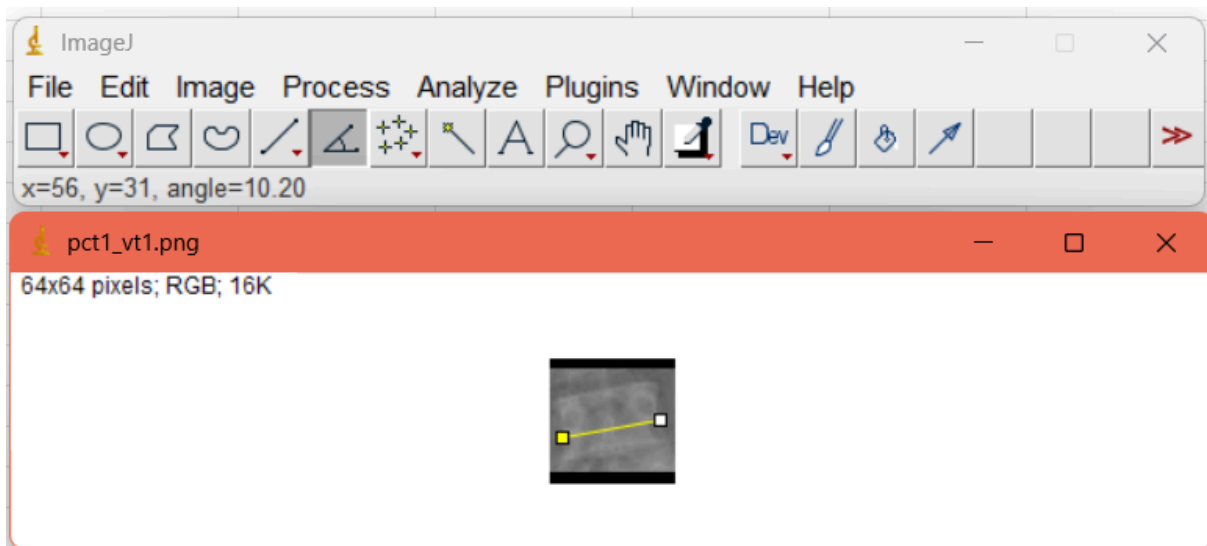
3.4.2 Dataset de Ângulo de Inclinação de Vértabras

Para a segunda etapa do *pipeline*, foi criado um conjunto de dados específico para o treinamento da rede CNN, destinado à análise da inclinação das vértebras. Diferentemente do *dataset* utilizado pela YOLOv8, que considerou radiografias de pacientes com e sem escoliose, este conjunto foi composto apenas pelas vértebras extraídas de imagens com escoliose, de modo a priorizar a diversidade de inclinações e aumentar a relevância clínica do treinamento.

A construção deste *dataset* ocorreu em duas etapas principais. Primeiramente, foi utilizado um algoritmo em Python para realizar o recorte individual das vértebras a partir das *bounding boxes* previamente anotadas no Roboflow. Em seguida, cada recorte foi aberto no software ImageJ, onde, por meio da ferramenta

de medição de ângulos, foi determinada manualmente a inclinação da vértebra em relação ao eixo horizontal, conforme ilustrado na Figura 4.

Figura 4 - Software ImageJ para identificação do ângulo de inclinação da vértebra



Fonte: Autores, 2025.

Os valores anotados foram registrados em um arquivo CSV, no qual cada linha associa o nome do arquivo de imagem da vértebra ao seu respectivo ângulo de inclinação, conforme ilustrado na Tabela 1. Os ângulos foram arredondados para números inteiros para facilitar o treinamento.

Tabela 1 - Exemplo de registros do dataset de vértebras anotadas

arquivo	ângulo
pct1_vt1.png	10
pct1_vt2.png	14
pct1_vt3.png	13
pct2_vt1.png	-26
pct2_vt2.png	-31
pct2_vt3.png	-29
pct51_vt1.png	-17
pct51_vt2.png	-10
pct51_vt3.png	-13

Fonte: Autores, 2025.

Ao final do processo, o *dataset* resultante contou com 779 vértebras recortadas e anotadas, formando uma base estruturada (imagem + valor do ângulo) para o treinamento da CNN. Esse conjunto permitiu que a rede aprendesse a estimar automaticamente a inclinação das vértebras, etapa essencial para o cálculo do ângulo de Cobb.

3.5 Ferramentas Utilizadas

Esta seção apresenta as principais ferramentas empregadas no desenvolvimento do projeto, abrangendo tanto os recursos de software quanto as plataformas de apoio utilizadas para treinamento dos modelos, processamento das imagens e implementação da aplicação.

3.5.1 Python

A linguagem de programação Python, criada por Guido van Rossum no início da década de 1990, destaca-se por sua sintaxe simples, legibilidade e ampla versatilidade em diversas áreas computacionais (RASCHKA; MIRJALILI, 2019). Seu ecossistema de bibliotecas especializadas, como NumPy, SciPy, Scikit-learn, Matplotlib, Pydicom e *frameworks* de aprendizado de máquina como PyTorch e TensorFlow, têm favorecido seu uso em aplicações científicas e biomédicas, incluindo processamento de imagens médicas e visão computacional (EKMEKCI; MCANANY; MURA, 2016). Essa combinação de simplicidade e robustez tornou o Python uma das linguagens mais adotadas em pesquisas voltadas à análise de imagens e desenvolvimento de sistemas inteligentes na área da saúde, por esses motivos foi a escolhida para o desenvolvimento do modelo deste projeto.

3.5.2 Pandas

A biblioteca Pandas é uma ferramenta de código aberto desenvolvida para a linguagem Python que fornece estruturas de dados rápidas, flexíveis e adaptáveis, além de um conjunto robusto de ferramentas para análise e manipulação de dados. Criada originalmente por Wes McKinney em 2008, o Pandas tornou-se uma das bibliotecas mais populares para ciência de dados, oferecendo estruturas como Series e DataFrame, que facilitam o tratamento de dados tabulares e heterogêneos (EKMEKCI; MCANANY; MURA, 2016). Sua capacidade de integração com outras

bibliotecas do ecossistema científico do Python, como NumPy, Matplotlib e Scikit-learn, permite que operações de limpeza, filtragem, agregação e transformação de dados sejam realizadas de forma eficiente e reprodutível (EKMEKCI; MCANANY; MURA, 2016).

No projeto, o Pandas foi utilizado para leitura de arquivos CSV contendo o mapeamento entre imagens de vértebras e os ângulos previamente anotados. Esses dados foram essenciais para treinar a CNN responsável por prever o ângulo de inclinação de cada vértebra. O carregamento da base foi feito de forma simples e estruturada, permitindo associar cada imagem ao valor de ângulo correspondente para divisão entre conjuntos de treino e teste.

3.5.3 NumPy

O NumPy (*Numerical Python*) é uma biblioteca essencial para computação científica em Python, fornecendo suporte para *arrays* multidimensionais e operações matemáticas de alto desempenho (OLIPHANT, 2015). Sua eficiência em operações vetoriais e matriciais o torna fundamental para o pré-processamento de imagens médicas, permitindo manipulação de matrizes de *pixels*, normalização e redimensionamento antes do uso em modelos de aprendizado de máquina, além de facilitar a integração com outras bibliotecas de inteligência artificial utilizadas neste trabalho, como PyTorch e Pillow.

Para o cálculo do ângulo de Cobb, as coordenadas das vértebras detectadas foram convertidas em pontos e, a partir delas, traçaram-se as retas representativas das placas terminais superior e inferior. A biblioteca NumPy foi utilizada nesse processo, em especial a função `arctan2`, que permite calcular a inclinação de uma reta levando em consideração corretamente os quadrantes do plano cartesiano, evitando ambiguidades comuns em operações trigonométricas. A diferença absoluta entre as inclinações obtidas dessas duas retas corresponde ao valor final do ângulo de Cobb.

Além disso, a biblioteca foi empregada em etapas de manipulação numérica durante o treinamento da CNN. As imagens, após abertas em escala de cinza, foram convertidas para *arrays* NumPy normalizados entre 0 e 1, garantindo compatibilidade com tensores PyTorch. Além disso, a biblioteca foi usada para

cálculos estatísticos, como a média do erro absoluto ao final da avaliação do modelo.

3.5.4 Pillow

O Pillow é uma biblioteca Python amplamente utilizada para processamento e manipulação de imagens, sendo uma continuação do projeto original *Python Imaging Library* (PIL) (CLARK, 2010). Além de operações básicas, como redimensionamento, recorte, rotação e conversão de cores, o Pillow também oferece recursos mais avançados, incluindo aplicação de filtros, ajustes de brilho, contraste e nitidez, além de suporte à manipulação de canais individuais de cores. Devido à sua simplicidade, eficiência e compatibilidade com outras bibliotecas do ecossistema Python, como NumPy e *frameworks* de aprendizado de máquina, o Pillow é amplamente adotado em projetos de visão computacional e ciência de dados (ANNALA et al., 2018).

A biblioteca *Pillow* foi utilizada para leitura e pré-processamento das imagens individuais das vértebras. As imagens foram abertas utilizando a função `Image.open()`, convertidas para tons de cinza e transformadas em *array* NumPy para posterior normalização e uso pela rede neural.

3.5.5 PyTorch

O PyTorch é uma biblioteca de código aberto para aprendizado profundo desenvolvida pelo *Facebook AI Research (FAIR)*, amplamente utilizada em pesquisa e aplicações práticas de inteligência artificial (SAWARKAR; ARREMSETTY, 2021). Ela oferece suporte para a criação e treinamento de redes neurais, incluindo operações sobre tensores multidimensionais, que são estruturas de dados similares a *arrays*, mas otimizadas para execução em CPU (*Central Processing Unit*) e GPU (*Graphics Processing Unit*). Entre suas funcionalidades destacam-se *autograd*, que permite o cálculo automático de gradientes para otimização de modelos, e uma interface flexível que facilita a experimentação com arquiteturas complexas de redes neurais (SAWARKAR; ARREMSETTY, 2021). O *PyTorch* é reconhecido por sua simplicidade e performance, permitindo que desenvolvedores construam, treinem e implementem modelos de aprendizado profundo de forma eficiente, sendo

amplamente adotado em visão computacional, processamento de linguagem natural e outras áreas de inteligência artificial (SUBRAMANIAN, 2018).

O PyTorch foi o *framework* principal para o desenvolvimento do modelo de CNN, voltado à predição do ângulo de inclinação das vértebras. Foram implementadas camadas convolucionais para extração de características, camadas totalmente conectadas para regressão do ângulo e técnicas como *dropout* para reduzir o *overfitting*. A biblioteca também forneceu os módulos de otimização (Adam), função de perda (`MSELoss`, do inglês *Mean Squared Error Loss*) e ferramentas para criação do *dataset*, o `Dataset` e `DataLoader`, que automatizaram a leitura e divisão em lotes durante o treinamento e avaliação.

3.5.6 YOLO

O YOLO é um algoritmo de detecção de objetos em tempo real que revolucionou a área de visão computacional ao combinar rapidez e precisão na identificação de múltiplas classes de objetos em imagens (REDMON et al., 2016). Diferentemente de métodos tradicionais que realizam a detecção em múltiplas etapas, o YOLO realiza a predição de *bounding boxes* e classes em uma única passagem pela CNN, otimizando o processo e permitindo aplicações em tempo real, mesmo em dispositivos com recursos limitados. Suas versões subsequentes, como YOLOv3 (YOLO versão 3), YOLOv4 (YOLO versão 4) e YOLOv5 (YOLO versão 5), aprimoraram a arquitetura original, introduzindo técnicas como redes residuais, detecção em múltiplas escalas e melhorias na velocidade e acurácia (BOCHKOVSKIY et al., 2020).

No projeto, o YOLO foi empregado para detectar individualmente as vértebras presentes nas imagens de raio-X. As coordenadas das caixas delimitadoras retornadas pelo modelo foram utilizadas diretamente nas etapas seguintes do *pipeline*, permitindo identificar as vértebras com maior inclinação para o cálculo do ângulo de Cobb. Essa abordagem garante precisão e eficiência na detecção de estruturas essenciais para a análise do grau de curvatura da coluna.

3.5.7 Ultralytics

A Ultralytics é uma empresa especializada em ferramentas de visão computacional, sendo amplamente reconhecida por suas implementações do modelo YOLO, uma abordagem de detecção de objetos em tempo real. A biblioteca Ultralytics fornece uma interface simples e eficiente para treinamento, validação e inferência de modelos YOLO, permitindo o uso de imagens e vídeos em diferentes formatos e resoluções. Além disso, inclui recursos para pré-processamento de dados, aumento de imagens (*data augmentation*) e exportação de modelos treinados para diferentes ambientes, facilitando a integração em projetos de inteligência artificial aplicados a imagens (KUKARTSEV et al., 2024).

No projeto, a biblioteca Ultralytics foi utilizada para implementar e treinar o YOLOv8 voltado à detecção das vértebras nas imagens de raio-X. A biblioteca permite carregar pesos pré-treinados, configurar parâmetros de treinamento como taxa de aprendizado, tamanho de lote e número de épocas, além de executar inferências diretamente no *pipeline*. As caixas delimitadoras obtidas com o modelo foram integradas ao processo de pós-processamento para seleção das vértebras de maior inclinação, fornecendo as informações necessárias para o cálculo do ângulo de Cobb de forma automatizada.

3.5.8 FastAPI

FastAPI é um micro *framework* Python moderno que simplifica a criação de APIs (*Application Programming Interface*) web utilizando a programação em Python. Ele foi desenvolvido para que os desenvolvedores conseguissem criar APIs de forma rápida e fácil, garantindo desempenho ideal e com um fácil gerenciamento sem que comprometa a qualidade e eficiência do código (RAMÍREZ, 2025). O FastAPI é amplamente reconhecido por reunir algumas características que o tornam uma escolha eficiente e moderna para o desenvolvimento (RAMÍREZ, 2025). Seu alto desempenho está associado ao uso de bibliotecas como Pydantic e à compatibilidade com ecossistema ASGI (*Asynchronous Server Gateway Interface*), promovendo também uma base sólida com o Starlette, que acaba aproveitando os recursos assíncronos (RAMÍREZ, 2025). Outro diferencial é a geração automática de documentação interativa, o que agiliza e facilita o trabalho dos desenvolvedores

(RAMÍREZ, 2025). Além disso, a clareza na validação de dados das requisições, apoiadas pelo Pydantic, contribui para que obtenha respostas mais detalhadas e fáceis de serem compreendidas, enquanto a simplicidade da sintaxe irá garantir uma curva de aprendizado tranquila e rápida com a implementação de *endpoints* (RAMÍREZ, 2025).

No projeto, o FastAPI foi utilizado para desenvolver a API responsável por receber imagens de raio-X enviadas pelos médicos, processar essas imagens com os modelos de detecção de vértebras e cálculo do ângulo de Cobb, e retornar os resultados em formato JSON (*JavaScript Object Notation*). Para permitir o envio de arquivos via requisições HTTP (*Hypertext Transfer Protocol*), a biblioteca `python-multipart` foi incluída como dependência, possibilitando que o FastAPI recebesse *uploads* em formato *multipart/form-data* e integrasse as imagens ao *pipeline* de processamento de forma eficiente.

3.5.9 Uvicorn

O Uvicorn é um servidor *web* baseado no ASGI, desenvolvido em Python, que acaba se destacando por sua leveza e alta eficiência sobre o tratamento de aplicações assíncronas (UVICORN, 2025). Feito sobre bibliotecas como *uvloop* e a *httptools*, ele alcança desempenho comparado a soluções já consolidadas como a do Node.js (LEAPCELL, 2025). Por sua versatilidade acaba se tornando a ferramenta apropriada para o desenvolvimento, oferecendo recursos como o recarregamento automático de código, quanto para um ambiente de produção, onde demonstra força e capacidade de lidar de forma estável múltiplos processos (TILKOV; VINOSKI, 2010). A sua compatibilidade com os *frameworks* mais modernos, como por exemplo o FastAPI, concretiza sua posição como uma das principais ferramentas quando se busca velocidade, escalabilidade e confiabilidade (UVICORN, 2025).

O Uvicorn foi o servidor ASGI escolhido para o projeto, sendo utilizado para executar a aplicação FastAPI. Durante o desenvolvimento, utilizou-se o parâmetro `--reload` para habilitar o *hot-reload*, permitindo que alterações no código fossem aplicadas automaticamente sem reiniciar o servidor. Em ambiente de implantação, a

aplicação é executada com múltiplos *workers* e, quando aplicável, posicionada atrás de um *proxy* reverso para garantir segurança e balanceamento de carga.

3.5.10 JSON

O JSON é um formato leve e amplamente utilizado para que haja troca de dados entre sistemas (JSON.ORG, 2023). Uma das suas principais características é a simplicidade na representação de informações em pares de chave e valor, o que acaba facilitando tanto a leitura por humanos quanto para o processamento por máquinas (RFC 8259, 2017). Por ser independente de linguagem, porém compatível com a maioria delas, o JSON se tornou um padrão nas aplicações *web*, APIs e serviços que exigem uma comunicação rápida e estruturada (MDN, 2024).

No projeto, o JSON foi utilizado como formato de saída da API, permitindo que os resultados das inferências, como as coordenadas das vértebras detectadas, os ângulos calculados e metadados relacionados ao paciente ou à requisição, fossem retornados de forma estruturada e de fácil interpretação pelo sistema. Essa abordagem facilitou a integração da API com o *frontend*.

3.5.11 Node.js

O Node.js é um ambiente de execução que dá a possibilidade de rodar aplicações desenvolvidas em JavaScript fora do navegador, oferecendo ao desenvolvedor uma plataforma eficiente e voltada para o alto desempenho (GEEKSFORGEEEKS, 2019). Possui uma arquitetura baseada em modelos assíncrono e orientado a eventos, permitindo assim lidar com múltiplas operações de entrada e saída sem que bloqueie o processamento (NODESOURCE, 2025). Essa característica o torna bastante utilizado em aplicações que exigem escalabilidade e resposta em tempo real, além de que, o Node.js acaba fornecendo um ecossistema robusto de pacotes acessíveis por meio do npm o que acaba facilitando a integração de bibliotecas e ferramentas durante o desenvolvimento de sistemas mais modernos (NODE.JS, 2024).

Na aplicação desenvolvida, o Node.js atuou como interpretador principal do código, reconhecendo a sintaxe baseada em JavaScript utilizada pelo React Native. Apesar disso, o uso de JavaScript de forma isolada não foi significativo, sendo

empregado apenas em arquivos auxiliares (*helpers*) em casos pontuais. Dessa forma, o foco esteve na utilização do React Native e das bibliotecas integradas, enquanto o Node.js garantiu a execução correta e a infraestrutura necessária para o funcionamento do projeto.

3.5.12 React Native

O React Native é um *framework* JavaScript de código aberto, criado pelo Facebook, voltado para o desenvolvimento de aplicativos móveis nativos para Android e iOS (*iPhone Operating System*), e Desktop, quando integrado a tecnologias como Electron (REACT NATIVE, 2024; ELECTRON, 2021). Assim como o React, ele adota o conceito de componentes reutilizáveis e uma abordagem declarativa, mas, ao invés de utilizar o Virtual DOM e elementos HTML, traduz seus componentes em elementos nativos de cada plataforma, garantindo melhor desempenho e experiência próxima à de aplicações desenvolvidas de forma tradicional (REACT NATIVE, 2024). Além disso, o React Native pode ser integrado ao React Navigation, ampliando sua flexibilidade e consolidando-se como uma das ferramentas mais relevantes para o desenvolvimento mobile multiplataforma na atualidade (REACT NAVIGATION, 2025).

No desenvolvimento do aplicativo, utilizou-se o React Native para a construção de todas as telas, garantindo uma interface responsiva e nativa para as plataformas móveis. Para a navegação entre telas, foi empregada a biblioteca `React Navigation`, que possibilitou a definição de rotas e a transição dinâmica entre diferentes interfaces. Além disso, implementou-se a biblioteca `Axios` para a realização das requisições à API, permitindo a comunicação entre o aplicativo e os serviços externos de forma eficiente.

3.5.13 Axios

O Axios é uma biblioteca JavaScript de cliente HTTP, baseada em *promises*, amplamente utilizada em aplicações *web* e React Native para realizar requisições a serviços *REST* (*GET*, *POST*, *PUT*, *DELETE*, entre outros) (AXIOS, 2023). A biblioteca oferece recursos como configuração de instâncias (*baseURL*, *headers*, *timeout*), interceptores, envio de JSON e arquivos (*multipart/form-data*) e

cancelamento de requisições, fornecendo uma solução flexível e eficiente para comunicação entre *frontend* e *backend* (AXIOS, 2023).

No projeto, o Axios foi utilizado por meio da classe `ApiService`, que centraliza toda a comunicação entre o *frontend* e o *backend* (Python). Através dessa classe, a biblioteca provê funcionalidades como injeção automática de token (padrão *Bearer*) nos *headers*, padronização de erros mapeando códigos HTTP para mensagens de negócio, controle de tempo de requisição, cancelamento de chamadas associadas ao ciclo de vida das telas e suporte a uploads de imagens de raio-X em formato *multipart/form-data* para análise automática do ângulo de Cobb. Dessa forma, o Axios foi integrado de maneira segura e organizada ao *pipeline* do aplicativo, facilitando a manutenção e extensibilidade do código.

3.5.14 Google Colab

O Google Colab é um serviço gratuito onde permite que o desenvolvedor escreva e execute o código em Python diretamente do navegador, sem que haja a necessidade de instalar *software* (GOOGLE, 2024). É baseado em Jupyter Notebook, combinado com código, texto e gráficos, é utilizado principalmente para o aprendizado de máquina (MARQUES, 2023). Por conta de ser acessível somente com uma conta do Google, tornou-se uma ferramenta bastante popular em áreas como ciência de dados, aprendizado de máquina, análise estatística e ensino de programação (GOOGLE, 2024).

O Google Colab foi utilizado como principal ambiente de desenvolvimento e treinamento dos modelos de aprendizado profundo empregados no projeto. O suporte a recursos de aceleração via GPU possibilitou a execução dos experimentos com maior eficiência computacional em comparação a máquinas locais.

Uma das funcionalidades que também contribuiu para o andamento do trabalho foi a integração direta com o Google Drive, que permite armazenar e acessar o *dataset* de forma prática e segura durante as sessões de treinamento. Por meio dessa integração, foi possível carregar as imagens anotadas, registrar os resultados e gerenciar os arquivos necessários para o desenvolvimento.

Após a etapa de treinamento, os pesos finais dos modelos foram exportados do Colab e integrados ao projeto principal. Esse procedimento foi aplicado tanto ao modelo de detecção de vértebras (YOLOv8) quanto à rede CNN para regressão do ângulo de Cobb, garantindo que os modelos pudessem ser utilizados localmente na API desenvolvida.

3.5.15 jsPDF

A jsPDF é uma biblioteca escrita em JavaScript utilizada para gerar documentos no formato PDF diretamente pelo navegador (FOUNDER, 2025). Ela possibilita a criação de arquivos com textos, imagens, gráficos e tabelas de maneira dinâmica, sem necessidade de *softwares* externos ou processamento no servidor (PINHO, 2019). Sua estrutura leve e compatibilidade com a maioria dos navegadores tornam o jsPDF uma ferramenta amplamente empregada em aplicações *web* que necessitam exportar ou imprimir informações de forma automatizada (FOUNDER, 2025).

No projeto, o jsPDF foi utilizado para a geração de relatórios em formato PDF, a partir dos dados processados pela aplicação. Os resultados das análises incluindo valores numéricos e imagens geradas pela etapa de processamento são convertidos em um documento estruturado, no qual os elementos são adicionados dinamicamente por meio de funções da biblioteca. Essa implementação possibilita a renderização e o *download* direto dos relatórios na aplicação, dispensando a necessidade de processamento no servidor e garantindo maior eficiência e integração entre as camadas do sistema.

4 IMPLEMENTAÇÃO

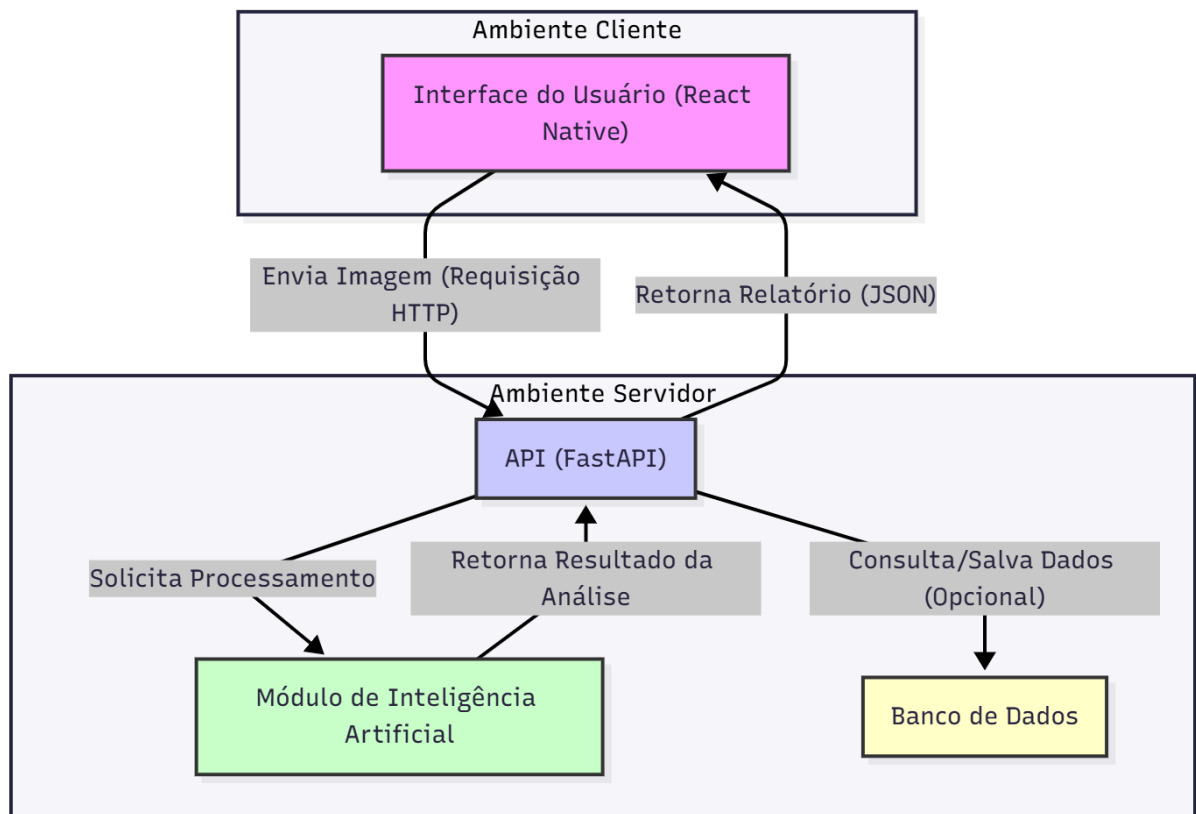
Nesta seção são apresentadas as etapas de implementação do sistema, abrangendo a arquitetura do projeto, os casos de uso, os requisitos funcionais e não funcionais definidos pelo grupo, o modelo entidade de entidade e relacionamento do banco de dados, os diagramas de classes e a interface da aplicação desenvolvida.

4.1 Arquitetura do Projeto

A arquitetura da solução foi projetada no modelo cliente-servidor, separando a aplicação em duas camadas principais: o *frontend* (interface do usuário) e o *backend* (lógica do servidor). Esta abordagem foi escolhida por sua flexibilidade e clara separação de responsabilidades, facilitando a manutenção do sistema.

A Figura 5 ilustra a macro arquitetura do sistema, composta por quatro componentes principais que interagem para realizar a análise das imagens de raio-X.

Figura 5 - Arquitetura de Componentes do Sistema



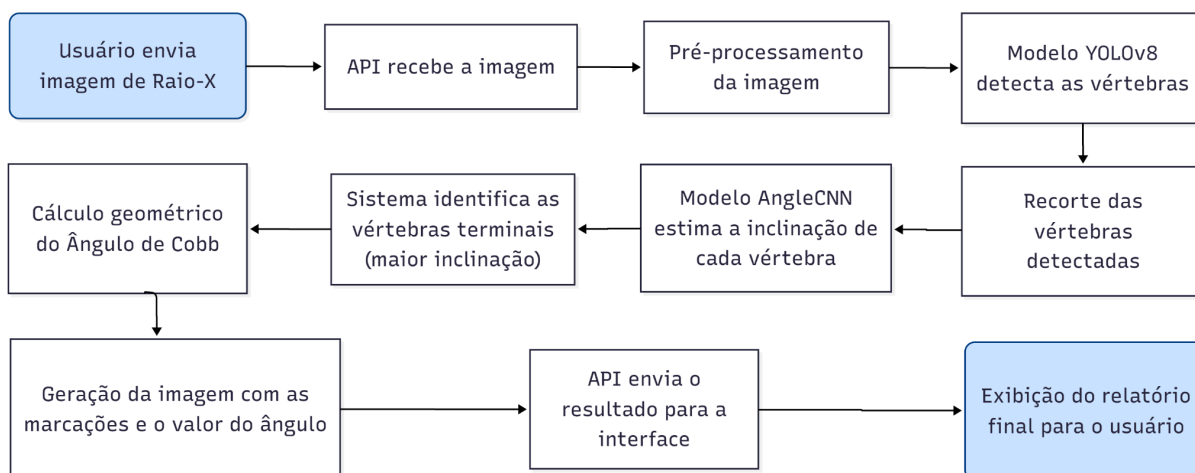
Fonte: Autores, 2025.

Os componentes do sistema funcionam de forma integrada. A **Interface do Usuário**, desenvolvida em React Native, permite ao médico realizar o *upload* de imagens e visualizar os resultados. A **API (FastAPI)** atua como orquestrador central, recebendo requisições, gerenciando o fluxo de dados e comunicando-se com os demais módulos. O **Módulo de Inteligência Artificial** é o núcleo de processamento, utilizando o modelo YOLOv8 para detectar as vértebras e uma CNN customizada para estimar seus ângulos. Por fim, o **Banco de Dados** armazena as informações de pacientes e relatórios, garantindo a persistência e o acesso ao histórico de exames.

4.1.1 Fluxo de Processamento de Dados

O fluxo de processamento de dados, ilustrado na Figura 6, detalha a sequência de operações desde a entrada da imagem até a apresentação do resultado.

Figura 6 - Fluxo de Processamento de Dados para Cálculo do Ângulo de Cobb



Fonte: Autores, 2025.

O processo operacional inicia-se com o *upload* da imagem do raio-x pelo usuário. No *backend*, a imagem passa por um pré-processamento para padronização. Em seguida, o modelo YOLOv8 localiza as vértebras, que são individualmente analisadas pela CNN para estimar seus ângulos de inclinação. Um algoritmo então identifica as vértebras terminais com base nas inclinações mais extremas para realizar o cálculo geométrico do ângulo de Cobb. O resultado,

contendo a imagem anotada com as linhas e o valor angular, é retornado para a interface, onde o usuário pode visualizar e gerar o relatório final.

4.2 Implementação do Software

4.2.1 Requisitos Funcionais

Requisitos funcionais são descrições dos serviços que o sistema deve oferecer, detalhando como ele deve reagir a entradas específicas e como deve se comportar ou não em determinadas situações (SOMMERVILLE, 2019). Abaixo, na Tabela 2, a relação de requisitos funcionais deste trabalho.

Tabela 2 - Relação de Requisitos Funcionais

Relação de Requisitos Funcionais	
ID	Descrição
RF01	O sistema deve permitir o cadastro de médicos com validação de login com e-mail e senha.
RF02	Médicos devem cadastrar pacientes de acordo com informações pessoais.
RF03	Deve ser possível realizar o <i>upload</i> de imagens de raio-X para processamento.
RF04	O sistema deve processar as imagens e calcular o ângulo de Cobb usando o modelo de IA.
RF05	O médico deve poder revisar os relatórios.
RF06	Relatórios devem incluir a imagem processada e ângulos calculados.
RF07	O sistema deve permitir ao médico realizar o <i>download</i> do relatório gerado.

Fonte: Autores, 2025.

4.2.2 Requisitos Não Funcionais

Os requisitos não funcionais representam limitações ou condições associadas aos serviços e funcionalidades do sistema, como restrições de tempo, condições para o processo de desenvolvimento e requisitos estabelecidos por normas ou regulamentações (SOMMERVILLE, 2019). A Tabela 3 apresenta os requisitos não funcionais.

Tabela 3 - Requisitos Não Funcionais

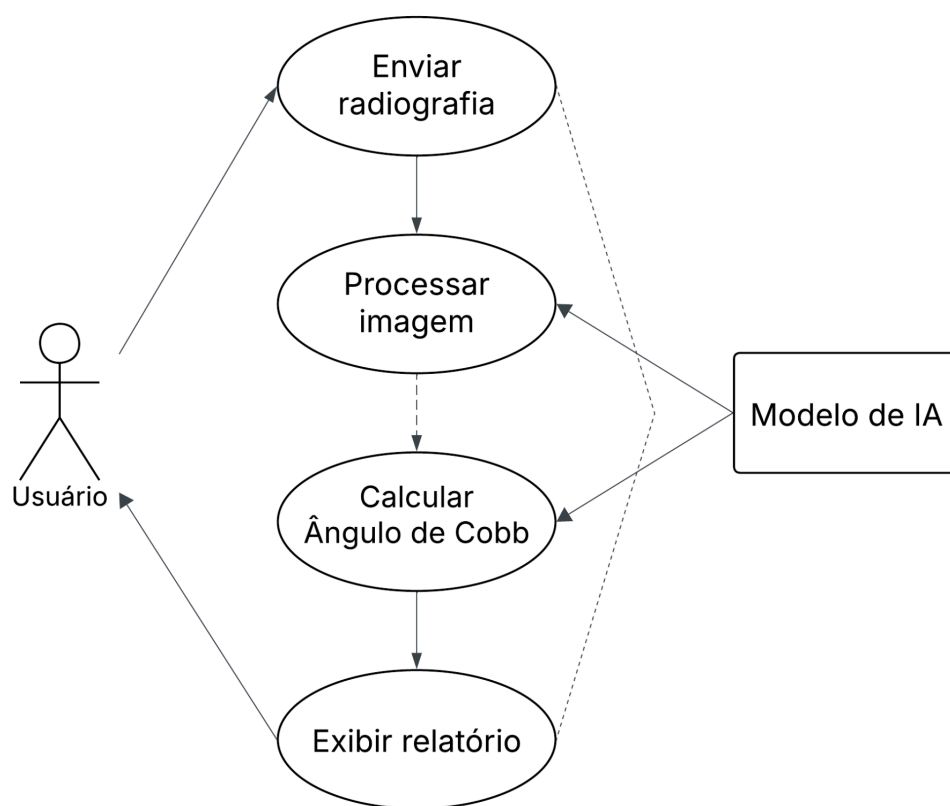
Relação de Requisitos Não Funcionais	
ID	Descrição
RNF01	O sistema deve processar imagens e retornar os resultados em até 10 segundos para garantir eficiência.
RNF02	Interfaces devem ser simples e intuitivas para facilitar a utilização por médicos e pacientes.
RNF03	Dados de imagens e relatórios devem ser manipulados com cuidado para evitar vazamentos ou uso indevido.
RNF04	O modelo de IA deve ter uma confiabilidade mínima de 90% de precisão no cálculo dos ângulos de Cobb para representar a viabilidade da solução.
RNF05	A estrutura do sistema deve ser modular para facilitar futuras manutenções e expansões.

Fonte: Autores, 2025.

4.2.3 Diagrama de Casos de Uso

O diagrama de caso de uso é utilizado para representar as funcionalidades oferecidas pelo sistema e a interação entre os atores e o sistema (SOMMERVILLE, 2019). Ele serve como uma ferramenta visual que descreve os principais cenários de uso, mostrando como o usuário e o modelo de IA interagem com o protótipo para realizar tarefas específicas. A Figura 7 apresenta cenários de casos de uso relevantes para o sistema.

Figura 7 - Diagrama de Casos de Uso



Fonte: Autores, 2025.

4.2.3.1 Descrição dos Casos de Uso

No que segue, serão apresentadas as descrições dos cenários descritos no diagrama de casos de uso apresentado, destacando como os objetivos são alcançados a partir das interações modeladas.

A Tabela 4 apresenta o caso de uso referente a **Enviar Radiografia**.

Tabela 4 - Caso de uso referente a Enviar Radiografia

Nome do Caso de Uso	Enviar Radiografia
Ator Principal	Usuário
Resumo	O usuário realiza o envio da imagem de raio-X do paciente para o sistema para que a IA possa iniciar o processo de análise.
Pré-Condições	1. O médico deve estar autenticado no sistema. 2. O paciente deve estar previamente cadastrado.
Pós-Condições	A imagem de raio-X fica disponível para processamento.
Fluxo Básico	1. O médico acessa o sistema. 2. O médico seleciona o paciente. 3. O médico anexa e envia a imagem de raio-X. 4. O sistema confirma o recebimento da imagem.
Fluxos Alternativos	3a. Caso o envio falhe, o sistema exibe uma mensagem de erro e solicita novo envio.
Restrições e Validações	O arquivo deverá ser enviado em um formato suportado (.png, .jpg ou .jpeg).

Fonte: Autores, 2025.

A Tabela 5 apresenta o caso de uso referente a **Processar Imagem**.

Tabela 5 - Caso de uso referente a Processar Imagem

Nome do Caso de Uso	Processar Imagem
Ator Principal	IA
Resumo	A imagem enviada é processada automaticamente pela IA, gerando um arquivo otimizado para análise.
Pré-Condições	A imagem deve ter sido recebida com sucesso.
Pós-Condições	A imagem está pronta para ser analisada pela IA.
Fluxo Básico	1. A IA recebe a imagem. 2. Aplica técnicas de pré-processamento. 3. Gera uma imagem otimizada para análise.
Fluxos Alternativos	2a. Se a imagem estiver corrompida, o sistema notifica o usuário para reenviar.

Fonte: Autores, 2025.

A Tabela 6 apresenta o caso de uso referente a **Calcular Ângulo de Cobb**.

Tabela 6 - Caso de uso referente a Calcular Ângulo de Cobb

Nome do Caso de Uso	Calcular Ângulo de Cobb
Ator Principal	IA
Resumo	A IA examina a radiografia para identificar traços anatômicos relevantes, detectar escoliose e calcular o ângulo de Cobb a partir das vértebras envolvidas.
Pré-Condições	A imagem deve estar processada corretamente e disponível para análise.
Pós-Condições	O diagnóstico e o ângulo de Cobb são calculados, armazenados e prontos para exibição ao usuário.
Fluxo Básico	1. A IA carrega a imagem processada. 2. Detecta as estruturas vertebrais. 3. Identifica a presença de escoliose. 4. Seleciona as vértebras superior e inferior da curva. 5. Traça linhas de referência e calcula o ângulo de Cobb. 6. Gera o diagnóstico baseado no valor do ângulo.
Fluxos Alternativos	2a. Se a imagem não for clara, a IA pode solicitar novo processamento ou alertar o médico. 6a. Se o cálculo do ângulo for inconclusivo, o sistema notifica o médico para análise manual.
Restrições e Validações	O sistema só consegue calcular o ângulo de Cobb. A análise é limitada à coluna vertebral.

Fonte: Autores, 2025.

A Tabela 7 apresenta o caso de uso referente a **Exibir Relatório**.

Tabela 7 - Caso de uso referente a Exibir Relatório

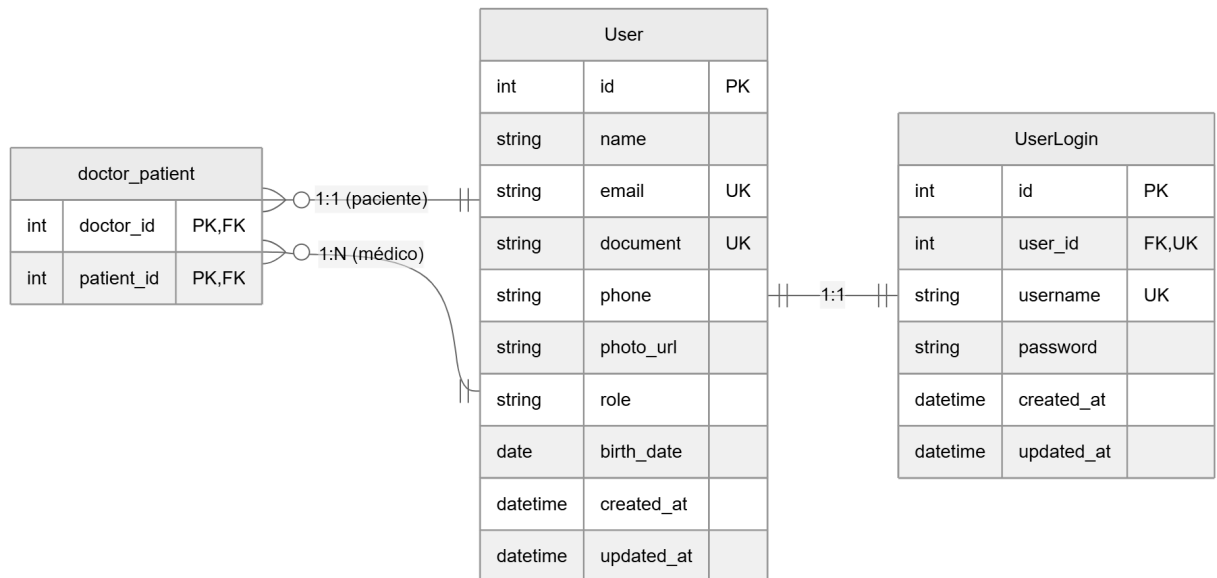
Nome do Caso de Uso	Exibir Relatório.
Ator Principal	Usuário
Resumo	O sistema exibe o relatório final com as informações do diagnóstico, imagem analisada e grau de curvatura.
Pré-Condições	O diagnóstico e análise devem estar concluídos.
Pós-Condições	Relatório disponível para acompanhamento clínico, se necessário.
Fluxo Básico	1. O sistema gera o relatório com base nos dados da IA. 2. O médico acessa o relatório.
Restrições e Validações	1. O médico pode visualizar somente relatórios dos seus pacientes.

Fonte: Autores, 2025.

4.3 Modelo de Entidade e Relacionamento

Neste subcapítulo, será apresentado o Modelo de Entidade e Relacionamento (MER) do *software* que está ilustrado na Figura 8 e é responsável por estruturar o banco de dados de forma que o modelo incorpore informações semânticas importantes sobre o mundo real (CHEN, 1977).

Figura 8 - Modelo de Entidade e Relacionamento



Fonte: Autores, 2025.

No parágrafo a seguir, será explicada cada tabela de dados representados anteriormente, com seus campos e relacionamentos.

- User:** Representa todos os usuários do sistema, incluindo médicos e pacientes, diferenciados pelo atributo *role*. Os campos armazenam informações pessoais e de identificação, como *name*, *email*, *document*, *phone* e *birth_date*, além de dados administrativos como *created_at* e *updated_at*. A entidade participa de dois relacionamentos:
 - Um relacionamento um para um (1:1) com “UserLogin”, responsável por armazenar as credenciais de acesso;
 - Um relacionamento um para muitos (1:N), mediado pela tabela “doctor_patient”, no qual um médico pode estar vinculado a vários pacientes;
- UserLogin:** Armazena as credenciais de autenticação dos usuários. Cada registro de login está associado a apenas um usuário, e essa relação é garantida por uma chave estrangeira que também possui restrição de unicidade (*user_id*). Com isso, estabelece-se um relacionamento 1:1 entre “User” e “UserLogin”, permitindo a separação entre dados pessoais e dados de autenticação, o que aumenta a segurança e facilita a modularidade do

sistema. Os atributos incluem *username*, *password* e os metadados de criação e atualização;

- **doctor_patient**: Registra as associações entre médicos e pacientes. Embora estruturalmente seja uma tabela associativa típica de relacionamentos muitos para muitos (N:N), no protótipo desenvolvido opera como um relacionamento 1:N, pois cada paciente é vinculado a apenas um médico no momento de sua criação. Dessa forma, um médico pode atender vários pacientes, mas cada paciente aparece apenas uma vez na tabela, estabelecendo assim uma relação “1 médico → N pacientes”. Essa modelagem permite manter flexibilidade estrutural enquanto respeita a regra de negócio vigente no protótipo.

É importante destacar que, para o protótipo desenvolvido neste trabalho, não foi utilizado um Sistema Gerenciador de Banco de Dados (SGBD) tradicional. Em vez disso, as entidades e relacionamentos foram implementados diretamente em Python utilizando SQLAlchemy, conforme os arquivos apresentados anteriormente. Essa abordagem permitiu uma rápida prototipação, mantendo a estrutura lógica compatível com o MER projetado e facilitando eventuais migrações para um SGBD no futuro.

4.4 Diagramas de Classes

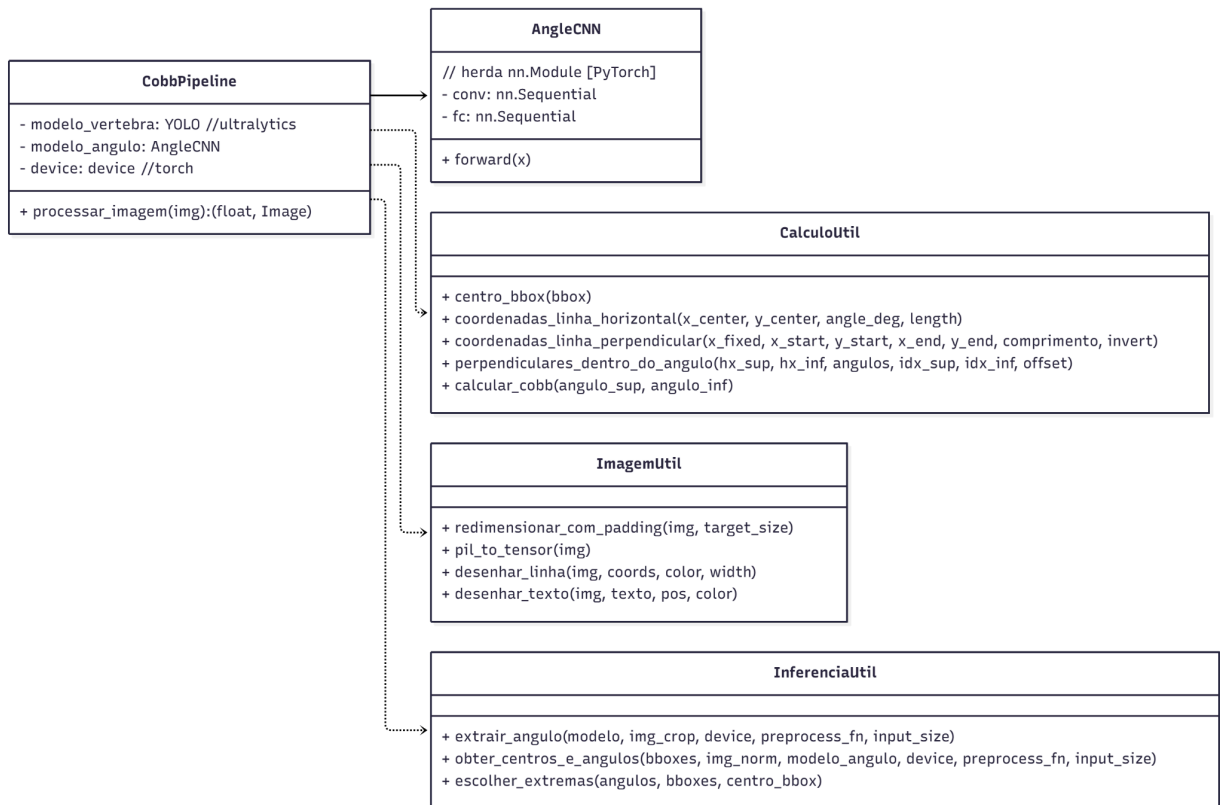
Os diagramas de classes representam a estrutura estática do sistema, mostrando as classes existentes, seus atributos, métodos e os relacionamentos entre elas. Esse tipo de diagrama é fundamental para compreender como os diferentes componentes do software se organizam e interagem em nível estrutural, facilitando a manutenção, evolução e reutilização do código (BELL, 2004b).

O sistema é composto por duas camadas principais: *backend*, responsável pelo processamento das imagens e cálculo do ângulo de Cobb, e *frontend*, responsável pela interação com o usuário. Para fins de documentação, o diagrama de classes está dividido para evidenciar essas duas camadas.

4.4.1 Backend

A Figura 9 apresenta o diagrama de classes referente ao *backend* da aplicação desenvolvida, que será detalhadamente explicado nesta seção.

Figura 9 - Diagrama de Classes - *Backend*



Fonte: Autores, 2025.

O *backend* é organizado em torno da classe **CobbPipeline**, responsável por integrar os modelos de detecção de vértebras e regressão de ângulos. Além disso, a classe utiliza diversas classes utilitárias para manipulação de imagens, inferência e cálculos geométricos. Cada classe possui responsabilidades bem definidas, garantindo modularidade, reutilização de código e maior clareza na manutenção e evolução do projeto.

4.4.1.1 Classe CobbPipeline

A classe **CobbPipeline** representa o núcleo do projeto, orquestrando a execução das etapas de processamento da imagem, detecção de vértebras, inferência dos ângulos e cálculo do ângulo de Cobb. Ela possui como atributos os modelos **YOLO** (para detecção de vértebras) e **AngleCNN** (para regressão dos

ângulos), além do dispositivo de processamento (`torch.device`). O método principal `processar_imagem(img: Image)` recebe uma imagem de radiografia, aplica pré-processamento, realiza a detecção e inferência, calcula o ângulo de Cobb e retorna o resultado juntamente com a imagem anotada.

4.4.1.2 Classe AngleCNN

A classe `AngleCNN` define a arquitetura de CNN responsável por inferir o ângulo de cada vértebra previamente detectada. A rede é composta por blocos convolucionais (`conv`) e camadas totalmente conectadas (`fc`), estruturados em sequência para extrair características visuais e realizar regressão do ângulo. O método `forward(x)` processa um tensor de entrada e retorna a previsão do ângulo.

4.4.1.3 Classe CalculoUtil

A classe `CalculoUtil` contém funções auxiliares para cálculo geométrico do ângulo de Cobb e definição das linhas horizontais e perpendiculares a partir das caixas delimitadoras das vértebras. Entre os métodos disponíveis destacam-se: `centro_bbox()`, para determinar o centro de uma caixa delimitadora; `coordenadas_linha_horizontal()`, `coordenadas_linha_perpendicular()` e `perpendiculares_dentro_do_angulo()`, utilizadas na construção das linhas geométricas; e `calcular_cobb()`, responsável pelo cálculo final do ângulo.

4.4.1.4 Classe ImagemUtil

A classe `ImagemUtil` fornece funcionalidades para manipulação de imagens, como redimensionamento com padding (`redimensionar_com_padding()`), conversão de imagens PIL para tensores (`pil_to_tensor()`), desenho de linhas (`desenhar_linha()`) e inserção de textos (`desenhar_texto()`). Essas funções suportam a visualização dos resultados do cálculo do ângulo de Cobb e o pré-processamento necessário para os modelos.

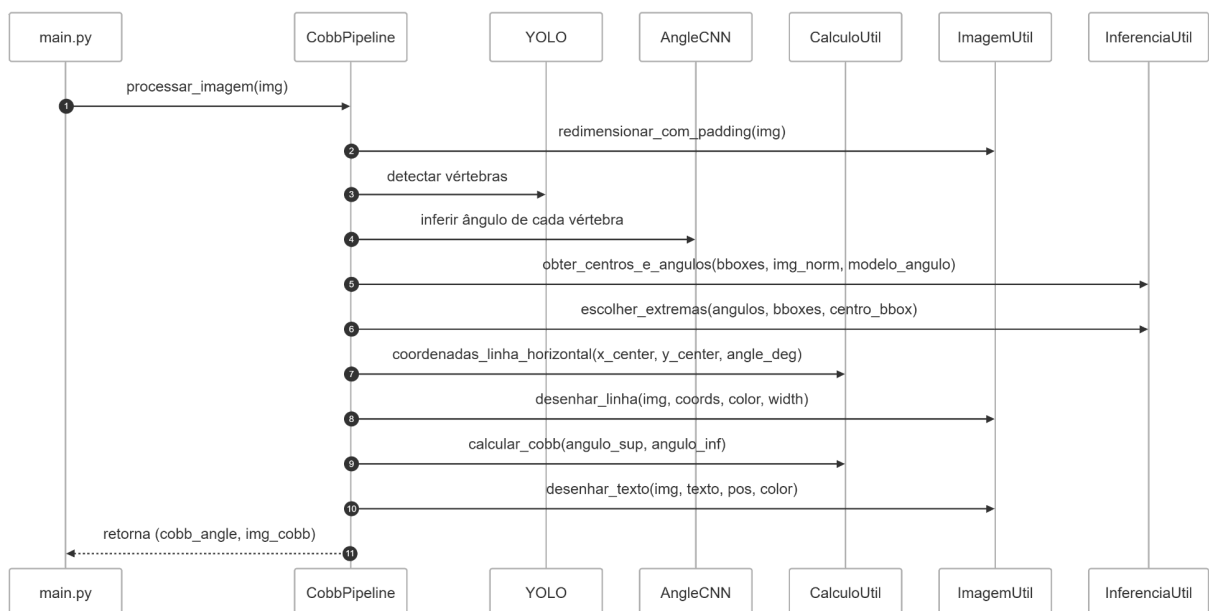
4.4.1.5 Classe InferenciaUtil

A classe `InferenciaUtil` concentra métodos para inferência com os modelos treinados. Entre as funções principais estão: `extrair_angulo()`, que processa uma imagem recortada para prever o ângulo de uma vértebra; `obter_centros_e_angulos()`, que integra informações das caixas delimitadoras e prediz os ângulos correspondentes; e `escolher_extremas()`, que seleciona as vértebras superior e inferior para o cálculo do ângulo de Cobb.

4.4.1.6 Diagrama de Sequência

O diagrama de sequência representa a interação entre os componentes do sistema ao longo do tempo, destacando a ordem de execução das operações e a troca de informações entre classes e módulos do sistema (BELL, 2004a). A Figura 10 ilustra o fluxo do processamento iniciado a partir da chamada da classe principal `CobbPipeline`, que coordena a execução das demais classes envolvidas na detecção de vértebras, inferência de ângulos e cálculo do ângulo de Cobb.

Figura 10 - Diagrama de Sequência IA



Fonte: Autores, 2025.

Inicialmente, o *script* principal (`main.py`) instancia a classe `CobbPipeline`, fornecendo como parâmetros o modelo de detecção de vértebras, a rede de regressão de ângulos e o dispositivo de execução. Em seguida, a função

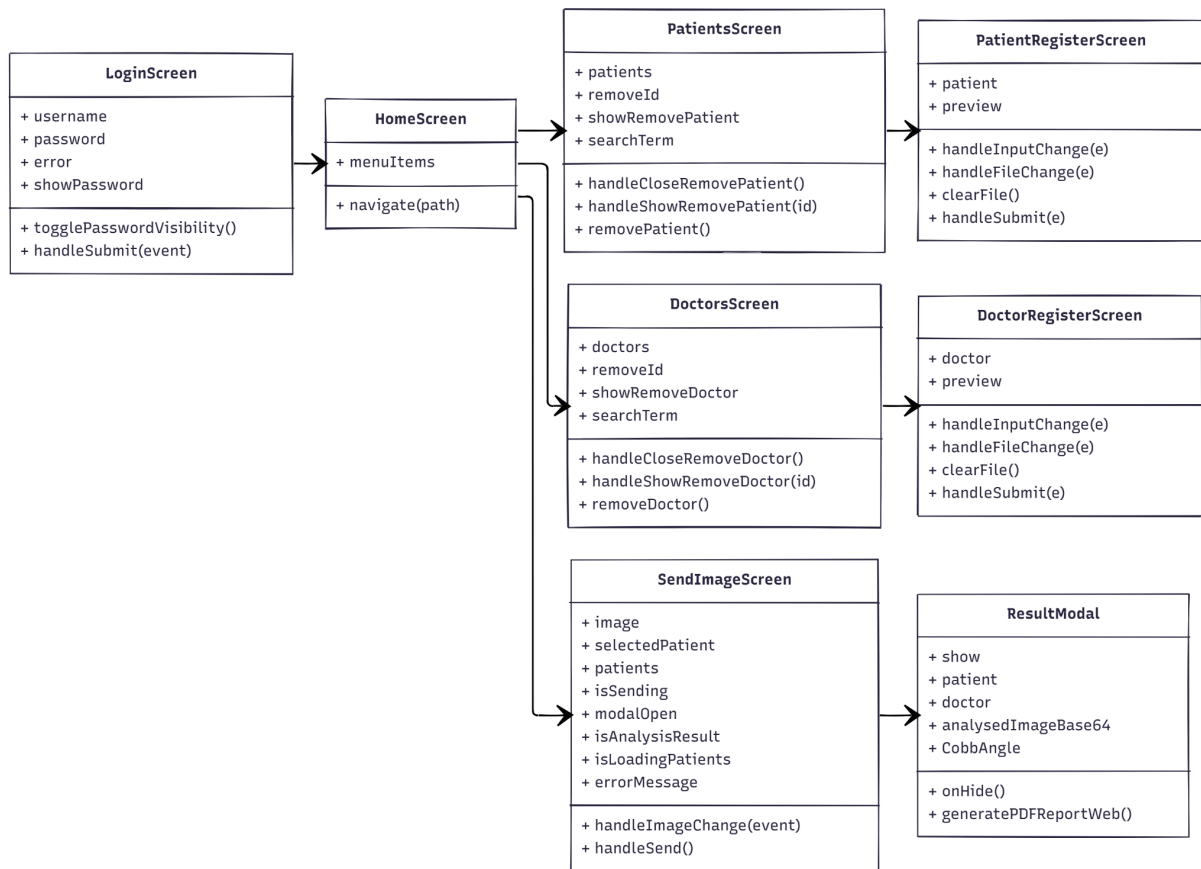
`processar_imagem()` é chamada para iniciar o fluxo de processamento da imagem.

A primeira etapa consiste na normalização da imagem, realizada pela classe `ImagemUtil`, que redimensiona e ajusta o raio-X para o formato esperado pelo modelo. Em seguida, a classe `YOLO` é acionada para identificar as vértebras presentes na imagem, e cada vértebra detectada é analisada pela `AngleCNN` para estimar seu respectivo ângulo.

Com os ângulos obtidos, a classe `InferenciaUtil` é responsável por determinar quais vértebras correspondem às extremidades superior e inferior da curvatura, utilizando funções de apoio para cálculo e organização dos dados. A classe `CalculoUtil`, por sua vez, gera as linhas de referência horizontais e calcula o ângulo de Cobb a partir das informações fornecidas. Por fim, a `ImagemUtil` é novamente utilizada para desenhar linhas e anotações na imagem processada, e o resultado final, o valor do ângulo de Cobb juntamente com a imagem anotada, é retornado ao *script* principal.

4.4.2 Frontend

A Figura 11 apresenta o diagrama de funções referente ao *frontend* da aplicação desenvolvida, que será detalhadamente explicado nesta seção.

Figura 11 - Diagrama de Funções - *Frontend*

Fonte: Autores, 2025.

O *frontend* do Sistema OrthoAI foi implementado em React Native, com o objetivo de fornecer uma interface completa, prática e moderna para o gerenciamento de pacientes, exames e imagens de raio-X. A aplicação integra-se a um módulo de IA responsável pela detecção de escoliose por meio do cálculo do ângulo de Cobb.

O diagrama evidencia as principais telas e componentes da aplicação, como LoginScreen, HomeScreen, PatientsScreen, PatientsRegisterScreen, DoctorsScreen, DoctorsRegisterScreen, SendImageScreen e ResultModal. Cada função (function) possui responsabilidades específicas voltadas à navegação, manipulação de dados e exibição de resultados, assegurando modularidade, reutilização de código e facilidade de manutenção do sistema.

4.4.2.1 Função LoginScreen

A tela de login é representada pela função `LoginScreen`, que tem como objetivo principal autenticar o usuário antes de permitir o acesso às funcionalidades do aplicativo. Essa função possui os atributos `username` e `password`, que ficam responsáveis por armazenar as credenciais inseridas pelo usuário, além do atributo `error`, que indica mensagens de erro em caso de falha no login e `showPassword`, que controla a visibilidade da senha. Entre os métodos, destaca-se `togglePasswordVisibility()`, responsável por alternar entre exibir e ocultar a senha, e `handleSubmit(event)`, que processa o envio do formulário de login. Após a autenticação bem-sucedida, o usuário é redirecionado para a tela inicial do sistema (`HomeScreen`).

4.4.2.2 Função HomeScreen

A tela inicial do aplicativo é representada pela função `HomeScreen`, que atua como ponto central de navegação dentro do sistema. Sua principal funcionalidade é apresentar ao usuário o menu com as opções disponíveis e permitir o redirecionamento para as demais funcionalidades do aplicativo. A função possui o atributo `menuItems`, que armazena os itens do menu acessível pelo usuário, e o método `navigate(path)`, responsável por conduzir o usuário à tela correspondente à funcionalidade selecionada. A partir da `HomeScreen`, o usuário pode navegar para as seguintes telas: `PatientsScreen`, para visualização e gerenciamento de pacientes; `DoctorsScreen`, para visualização e gerenciamento de médicos; e `SendImageScreen`, para envio de imagens de exames para análise.

4.4.2.3 Função PatientsScreen

A função `PatientsScreen` representa a tela de gerenciamento de pacientes, permitindo ao usuário visualizar, editar ou remover pacientes do sistema. Os atributos da função incluem `patients`, que armazena a lista de pacientes, `removeId`, que indica o paciente a ser removido, e `searchTerm`, utilizado para filtrar a lista de pacientes. O atributo `showRemovePatient` controla a exibição da interface de remoção. Os métodos `handleCloseRemovePatient()` e

`handleShowRemovePatient(id)` gerenciam a abertura e o fechamento da interface de remoção, enquanto `removePatient()` e `handleSubmit()` processam as remoções e alterações dos pacientes. A tela também permite navegar para `PatientRegisterScreen`, para cadastro ou edição de informações de pacientes.

4.4.2.4 Função `PatientRegisterScreen`

A tela de cadastro ou edição de pacientes é representada pela função `PatientRegisterScreen`, que possui os atributos `patient`, responsável por armazenar os dados do paciente, e `preview`, que permite visualizar imagens ou documentos carregados. Os métodos `handleInputChange(e)` e `handleFileChange(e)` que processam a entrada de dados e arquivos, `clearFile()` limpa arquivos selecionados, e `handleSubmit(e)` envia as informações para o sistema.

4.4.2.5 Função `DoctorsScreen`

A função `DoctorsScreen` representa a tela de gerenciamento de médicos, possibilitando ao Administrador visualizar, editar ou remover médicos do sistema. Entre seus atributos estão `doctors`, que armazena a lista de médicos, `removeId`, `searchTerm`, e `showRemoveDoctor`, que controla a exibição da interface de remoção. Os métodos `handleCloseRemoveDoctor()`, `handleShowRemoveDoctor(id)` e `removeDoctor()` são responsáveis pelo gerenciamento e execução das ações de remoção. A tela permite ainda navegar para `DoctorScreen`, para cadastro ou edição de médicos.

4.4.2.6 Função `DoctorRegisterScreen`

A tela de cadastro ou edição de médicos é representada pela função `DoctorRegisterScreen`, que contém os atributos `doctor`, `username`, `password`, `preview`, além de *flags* como `showAuthMessage` e `showPassword`, utilizadas para controle da interface. Os métodos `togglePasswordVisibility()`, `handleInputChange(e)`,

`handleFileChange(e)`, `clearFile()`, `handleSubmit(e)`, `handleRemoveAuth()` e `addAuth()` permitem gerenciar dados do médico, autenticação e arquivos relacionados.

4.4.2.7 Função `SendImageScreen`

A função `SendImageScreen` representa a tela de envio de imagens para análise. Seus atributos incluem `image`, que armazena a imagem a ser enviada, `selectedPatient`, referente ao paciente selecionado, `patients`, `isSending`, que indica o envio em andamento, `modalOpen`, `analysisResult`, `isLoadingPatients` e `errorMessage`. Entre os métodos, `handleImageChange(event)` e `handleSend()` gerenciam o envio da imagem e o tratamento do resultado da análise.

4.4.2.8 Função `ResultModal`

A função `ResultModal` representa o *modal* responsável por exibir o resultado da análise das imagens. Entre seus atributos estão o `show`, que indica se o *modal* está aberto, `patient` e `doctor`, que armazenam informações do paciente e do médico, `analysedImageBase64`, contendo a imagem processada, e `cobbAngle`, referente ao ângulo Cobb calculado. Os métodos `onHide()` e `generatePDFReportWeb()` permitem, respectivamente, fechar o *modal* e gerar o relatório em *Portable Document Format* (PDF).

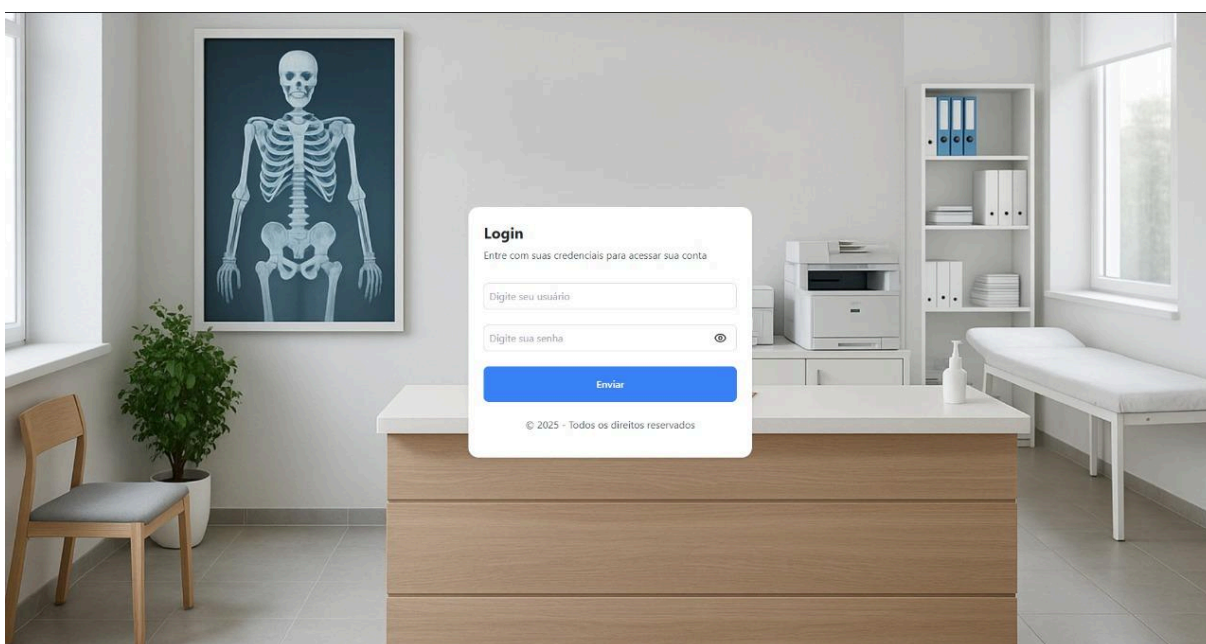
4.5 Funcionamento e Interface

O sistema foi desenvolvido com o objetivo de assegurar desempenho satisfatório, usabilidade e responsividade, de modo a apoiar a rotina do profissional médico como ferramenta de suporte durante consultas e na elaboração de relatórios subsequentes. A interface apresenta um fluxo de navegação claro que evidencia as relações entre as telas e o fluxo de dados entre o *frontend* e o *backend*. Esta seção irá apresentar as funcionalidades e interface do sistema, alimentado com informações fictícias para fins de visualização.

4.5.1 Tela de Login

O acesso ao sistema é realizado por meio da tela de login, ilustrada na Figura 12, que autentica o usuário antes de permitir o uso das funcionalidades do aplicativo. Nessa tela, o usuário insere suas credenciais. O sistema também apresenta mensagens de erro em caso de falha na autenticação e permite alternar a visibilidade da senha para maior conveniência.

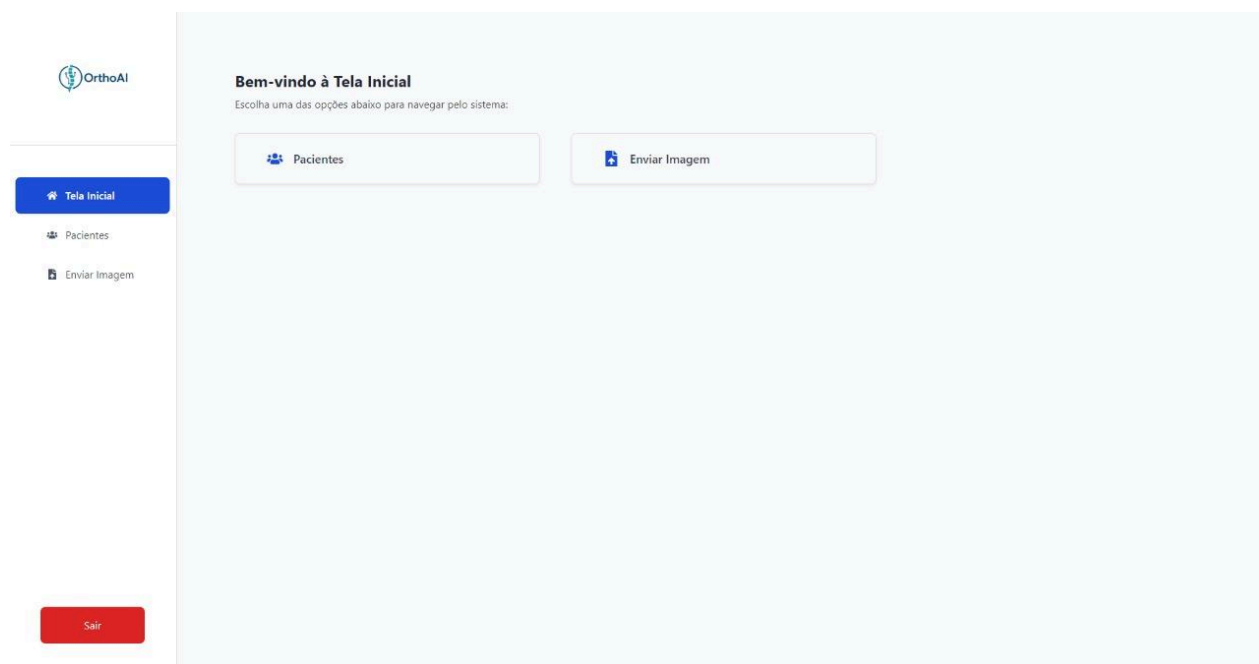
Figura 12 - Tela de Login



Fonte: Autores, 2025.

Após a validação das credenciais, o usuário é redirecionado à tela inicial ilustrada na Figura 13, garantindo segurança e controle de acesso às informações do sistema.

Figura 13 - Tela inicial (Médico logado)



Fonte: Autores, 2025.

4.5.2 Envio de Imagens

A Figura 14 apresenta a tela de envio de imagem disponível para o médico logado no sistema. Nessa interface, o usuário pode selecionar arquivos de raio-X diretamente de seu dispositivo, garantindo que apenas imagens válidas sejam enviadas para análise. A disposição dos elementos da tela foi planejada para proporcionar uma experiência intuitiva, permitindo que o médico realize o envio de forma rápida e eficiente, reduzindo possíveis erros e agilizando o processo de preparação das imagens para processamento automatizado.

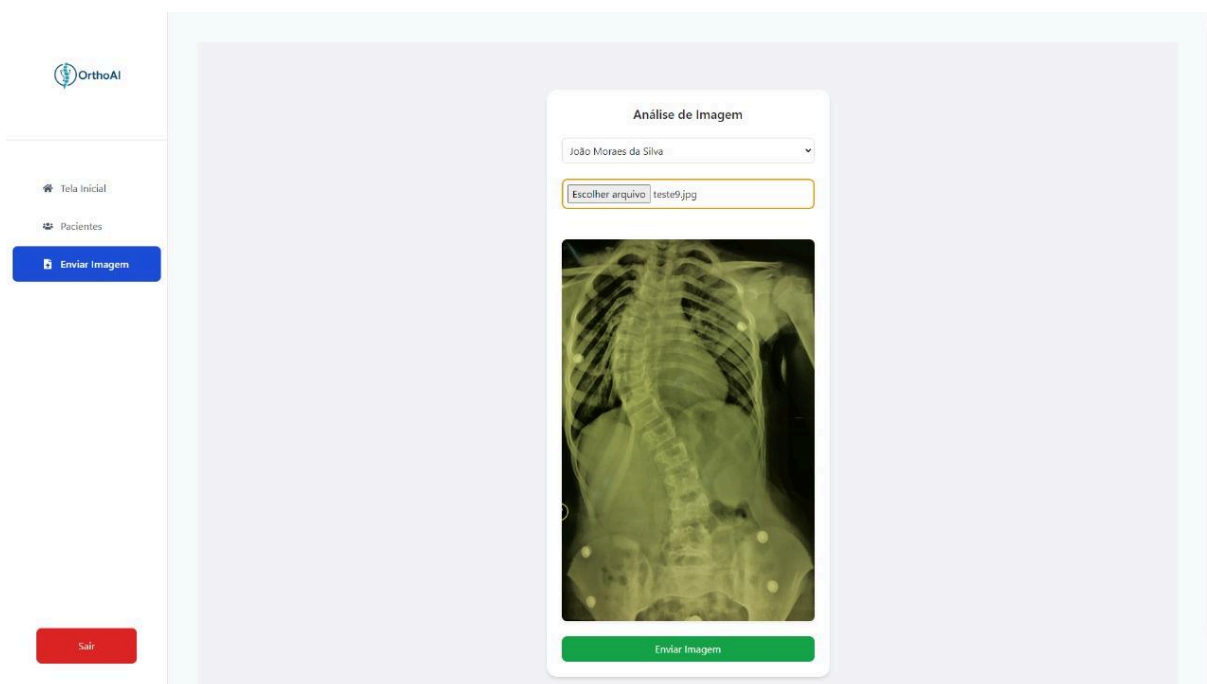
Figura 14 - Tela de envio de imagem (Médico logado)

The screenshot displays the OrthoAI web application interface. On the left, a sidebar contains the OrthoAI logo at the top, followed by navigation links: 'Tela Inicial', 'Pacientes', 'Enviar Imagem' (highlighted in blue), and 'Sair' (in a red box at the bottom). The main content area is a light blue rectangle. Centered within this area is a white modal box titled 'Análise de Imagem'. This modal contains a dropdown menu labeled 'Selecione o paciente', a file selection button labeled 'Escolher arquivo' next to the text 'Nenhum arquivo escolhido', and a grey button at the bottom labeled 'Enviar Imagem'.

Fonte: Autores, 2025.

A Figura 15 ilustra a tela após a seleção da imagem pelo usuário. Nessa etapa, o sistema exibe a pré-visualização do arquivo escolhido, permitindo ao médico confirmar que a imagem correta foi selecionada antes de iniciar a análise automatizada. Esse recurso oferece maior segurança e controle sobre o processo, assegurando que a inteligência artificial realize a detecção das vértebras e o cálculo do ângulo de Cobb com base na imagem adequada, otimizando a precisão dos resultados clínicos.

Figura 15 - Tela de imagem já selecionada (Médico logado)



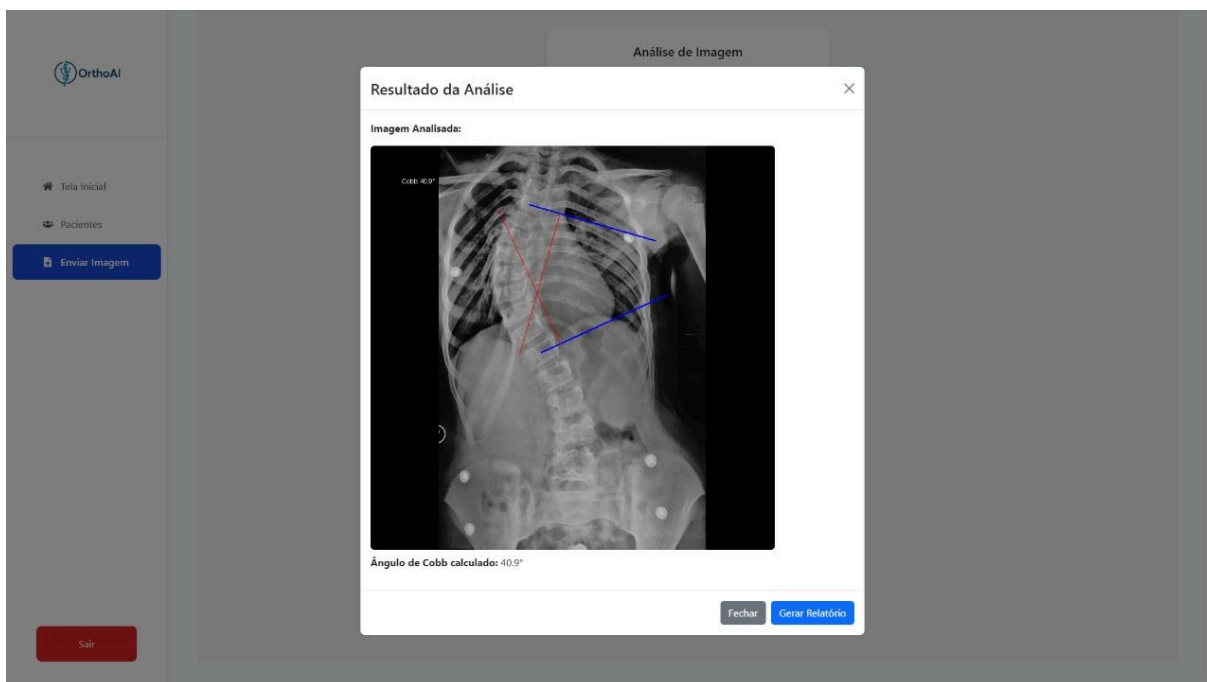
Fonte: Autores, 2025.

Durante o envio, é possível acompanhar o progresso do processo, garantindo que os dados sejam transmitidos de forma segura.

4.5.3 Geração de Relatórios

Após a análise automatizada das imagens de raio-X, o sistema permite que o usuário gere relatórios em formato PDF, conforme ilustrado na Figura 16. Esses relatórios incluem informações detalhadas, como o cálculo do ângulo de Cobb e a avaliação da escoliose, oferecendo uma documentação precisa e organizada dos resultados obtidos pela inteligência artificial. A disponibilização dessas informações de forma estruturada contribui para uma interpretação mais clara e segura pelos profissionais de saúde.

Figura 16 - Modal de imagem após análise (Médico logado)



Fonte: Autores, 2025.

Conforme demonstrado na Figura 17, os relatórios podem ser baixados e consultados a qualquer momento, garantindo praticidade no acompanhamento clínico e preservação do histórico de exames. Essa funcionalidade facilita o registro contínuo dos resultados, permitindo que médicos e pacientes monitorem a evolução da escoliose de maneira confiável e eficiente, promovendo maior controle sobre o tratamento e a tomada de decisões clínicas.

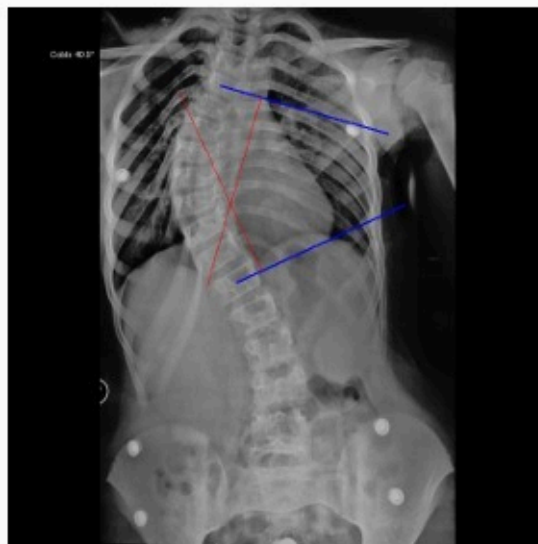
Figura 17 - Relatório do Raio-X analisado (Médico logado)



Relatório de Análise



Nome: João Moraes da Silva
Data de Nascimento: 10/10/2015
Médico Responsável: Gregory House
Ângulo de Cobb: 40.9°



Este relatório apresenta a análise realizada por IA. A interpretação médica profissional é indispensável para diagnóstico e tratamento.

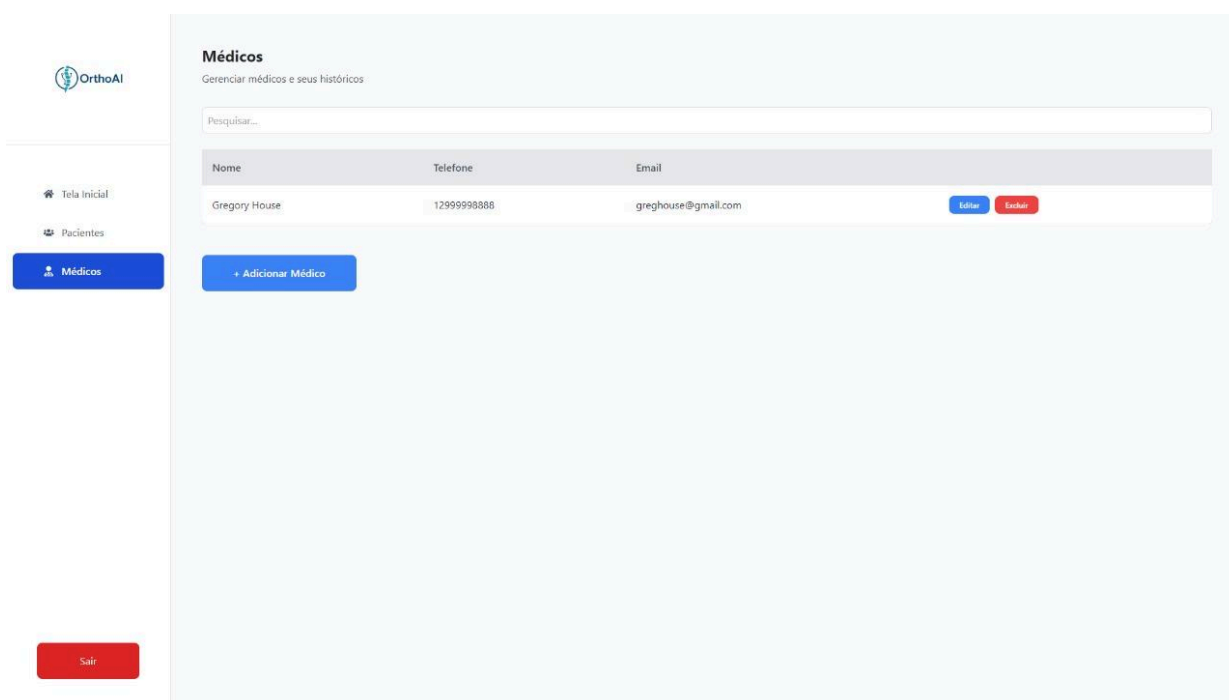
Ortho AI - Sistema de Análise de Imagens de Raio-X

Fonte: Autores, 2025.

4.5.4 Gerenciamento de Usuários

A Figura 18 apresenta a tela de visualização dos médicos cadastrados, disponível para o administrador logado no sistema. Por meio dessa interface, é possível consultar informações cadastrais completas de cada profissional, incluindo registro profissional e dados de contato, promovendo uma gestão organizada e confiável do corpo clínico. A tela ainda permite a identificação rápida de médicos ativos e facilita a supervisão administrativa das equipes que utilizam o sistema, garantindo maior controle sobre o gerenciamento de usuários.

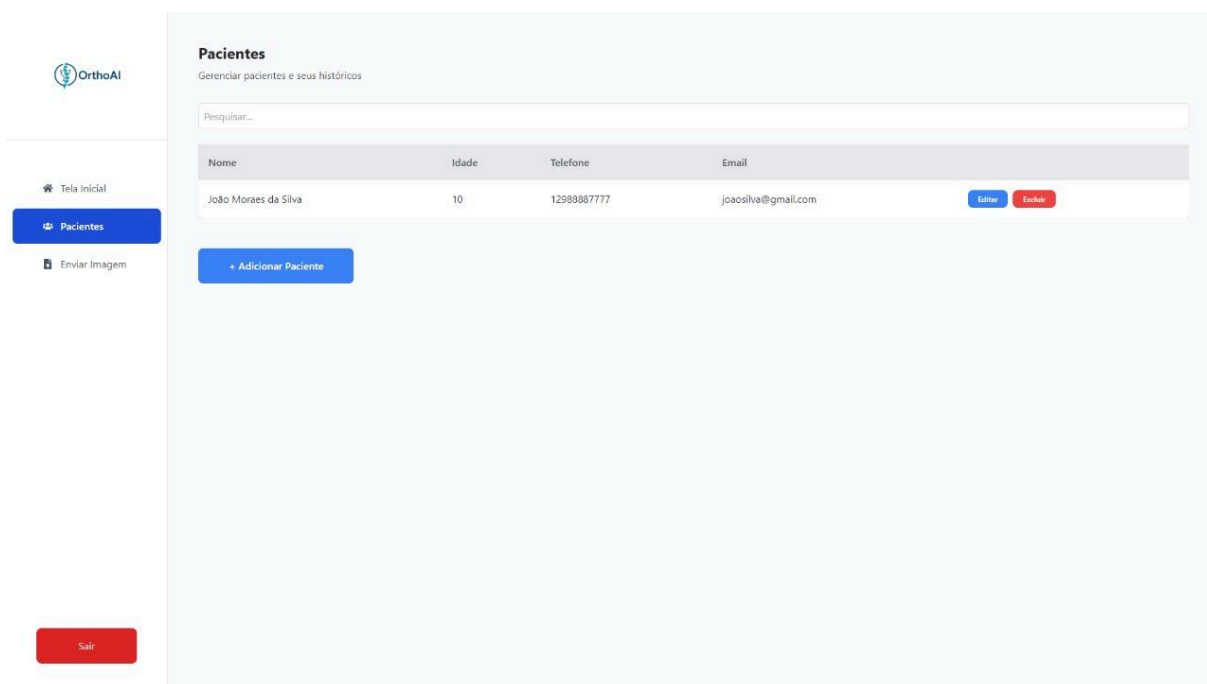
Figura 18 - Tela de visualização de médicos cadastrados (Administrador logado)



Fonte: Autores, 2025.

A Figura 19 ilustra a tela de visualização dos pacientes cadastrados, na qual o médico pode acessar informações detalhadas sobre cada paciente, como dados pessoais e informações para acompanhamento clínico. Essa funcionalidade proporciona uma visão abrangente do banco de dados de pacientes, possibilitando decisões administrativas e clínicas fundamentadas em informações precisas e atualizadas.

Figura 19 - Tela de visualização de pacientes cadastrados (Médico logado)



Fonte: Autores, 2025.

A Figura 20 demonstra a tela de edição de médicos, permitindo ao administrador atualizar dados cadastrais, corrigir informações inconsistentes e complementar registros existentes. Tal funcionalidade assegura a manutenção da acurácia das informações, fortalecendo a confiabilidade do sistema na gestão do corpo clínico e garantindo que os dados utilizados para análise e geração de relatórios estejam sempre corretos e consistentes.

Figura 20 - Tela de edição de médicos (Administrador logado)

The screenshot shows the 'Editar Médico' (Edit Doctor) interface. On the left is a sidebar with the OrthoAI logo and navigation links: 'Tela Inicial', 'Pacientes', and 'Médicos'. The main area is titled 'Editar Médico' and contains a form for editing a doctor's profile. At the top of the form is a circular profile picture of a man. Below the picture are several input fields: 'Nome Completo' (Gregory House), 'Email' (greghouse@gmail.com), 'Telefone' (1299999888), 'Documento' (12345678910), 'Data de Nascimento' (12/12/1980), and 'Data de Cadastro' (12/10/2025). A note below the registration date states 'Este campo não pode ser alterado'. At the bottom of the form are three buttons: 'Adicionar autenticação' (blue), 'Cancelar' (red), and 'Salvar Alterações' (green). A red 'Sair' button is located in the bottom left corner of the sidebar area.

Fonte: Autores, 2025.

A Figura 21 apresenta a tela de edição de pacientes, oferecendo recursos análogos aos disponíveis para edição de informações dos médicos. Por meio dessa interface, o médico pode modificar dados cadastrais, atualizar histórico clínico ou corrigir informações incorretas. Essa ferramenta é essencial para garantir a integridade e a confiabilidade do banco de dados, assegurando que as informações utilizadas pela inteligência artificial e pelos relatórios gerados reflitam a realidade clínica dos pacientes de forma precisa e segura.

Figura 21 - Tela de edição de pacientes (Médico logado)

OrthoAI

← Editar Paciente

Nome Completo: João Moraes da Silva

Email: joaosilva@gmail.com

Telefone: 12988887777

Documento: 10987654321

Data de Nascimento: 10/10/2015

Data de Cadastro: dd/mm/aaaa
Este campo não pode ser alterado

Cancelar Salvar Alterações

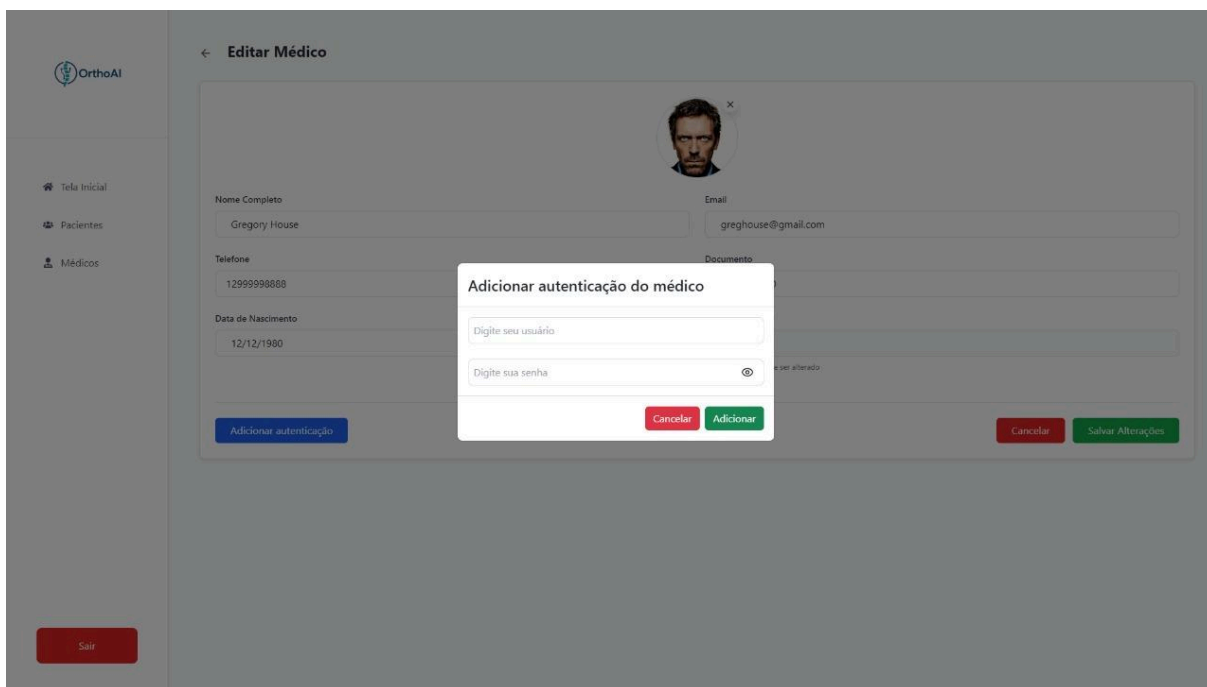
Sair

Fonte: Autores, 2025.

Durante o processo de autenticação de médicos, o sistema exibe mensagens de confirmação para cada modificação, a fim de evitar alterações indevidas ou erros de validação. A autenticação requer credenciais válidas (usuário e senha), assegurando a integridade no acesso às funcionalidades restritas.

A Figura 22 apresenta a janela *modal* utilizada no processo de autenticação de médicos, na qual o administrador insere as credenciais de acesso (usuário e senha) do profissional. Essa interface foi projetada para garantir que apenas usuários autorizados tenham acesso às funcionalidades restritas do sistema, assegurando a integridade e a segurança das informações sensíveis. Além disso, a disposição clara dos campos de entrada e dos elementos de interação contribui para a minimização de erros durante o processo de login.

Figura 22 - Modal para autenticação do médico (Administrador logado)



Fonte: Autores, 2025.

A Figura 23 ilustra o mecanismo de confirmação implementado pelo sistema, exibindo a mensagem de alerta que solicita a validação da ação antes de prosseguir com a autenticação. Esse recurso tem como objetivo prevenir alterações indevidas ou ações acidentais, promovendo maior confiabilidade e controle sobre operações críticas. A validação prévia de cada modificação garante que o acesso às informações sensíveis seja realizado de forma segura e auditável, reforçando a robustez do sistema frente a possíveis inconsistências ou falhas humanas.

Figura 23 - Confirmação de autenticação de médico (Administrador logado)

OrthoAI

Editar Médico

Atenção!

O médico em questão já possui um registro de autenticação. Tem certeza que deseja excluir o atual e criar uma nova autenticação?

Não **Sim**

Nome Completo: Gregory House

Email: greghouse@gmail.com

Telefone: 12999998888

Data de Nascimento: 12/12/1980

Adicionar autenticação

Cancelar Salvar Alterações

Sair

Fonte: Autores, 2025.

5 TESTES E RESULTADOS

Nesta seção, serão apresentados os testes e resultados obtidos na análise de desempenho do modelo de inteligência artificial e das APIs desenvolvidas neste estudo.

5.1 Modelo de Detecção de Vértex

O treinamento do modelo YOLOv8 foi monitorado por métricas de desempenho com o objetivo de avaliar a precisão da detecção de vértebras para o cálculo do ângulo de Cobb. As métricas consideradas foram:

- **Precisão (*Precision*):** Mede a proporção de detecções corretas em relação ao total de detecções realizadas pelo modelo. Um valor alto indica que o modelo erra pouco ao identificar objetos (HICKS et al., 2022).
- ***Recall*:** Indica a capacidade do modelo de detectar todos os objetos presentes nas imagens. Quanto maior, mais objetos reais foram corretamente identificados (HICKS et al., 2022).
- ***Mean Average Precision com Intersection over Union de 50% (mAP@50)*:** Representa a média da precisão em diferentes classes considerando um limiar de sobreposição moderado (50%), fornecendo uma avaliação geral da capacidade de detecção do modelo (HICKS et al., 2022).
- ***Mean Average Precision com Intersection over Union de 50% a 95% (mAP@50-95)*:** Similar ao mAP@50, mas calculado para *Intersection over Union* variando de 50% a 95%, oferecendo uma avaliação mais rigorosa da precisão do modelo em diferentes limiares (HICKS et al., 2022).

Essas estatísticas estão representadas na Tabela 8.

Tabela 8 - Resultados das métricas por época no treinamento do YOLOv8

(continua)

Época	Precisão	<i>Recall</i>	mAP@50	mAP@50-95
1	0,832	0,400	0,644	0,351
10	0,903	0,940	0,960	0,657
20	0,897	0,958	0,967	0,664

(conclusão)

Época	Precisão	<i>Recall</i>	mAP@50	mAP@50-95
30	0,908	0,955	0,970	0,676
36	0,921	0,948	0,970	0,677
40	0,916	0,935	0,965	0,671
48	0,916	0,941	0,965	0,669

Fonte: Autores, 2025.

Após a definição das métricas, é possível observar a evolução do desempenho do modelo ao longo do treinamento. A precisão começou em 0,832 na primeira época e aumentou gradualmente, estabilizando em torno de 0,916 nas últimas épocas, indicando que o modelo aprendeu a identificar corretamente a maioria das vértebras detectadas.

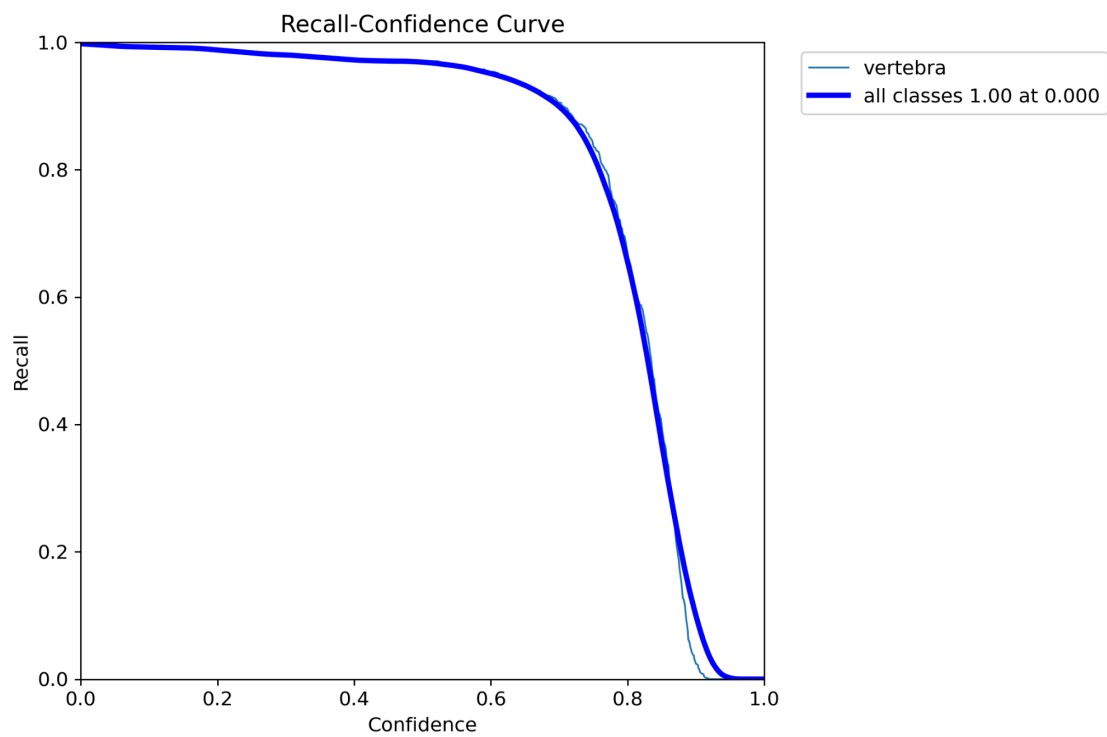
O *recall* apresentou crescimento mais acentuado, saindo de 0,400 na primeira época e atingindo aproximadamente 0,941 na última, demonstrando que o modelo foi capaz de localizar quase todas as vértebras presentes nas imagens.

O mAP@50 apresentou valores consistentes a partir da época 30, alcançando 0,970 na época 36, o que evidencia a alta capacidade do modelo em detectar corretamente as vértebras com um limiar moderado de sobreposição.

Já o mAP@50-95, métrica mais rigorosa, também apresentou melhora constante, indicando que o modelo mantém boa qualidade de detecção mesmo sob critérios mais estritos de sobreposição.

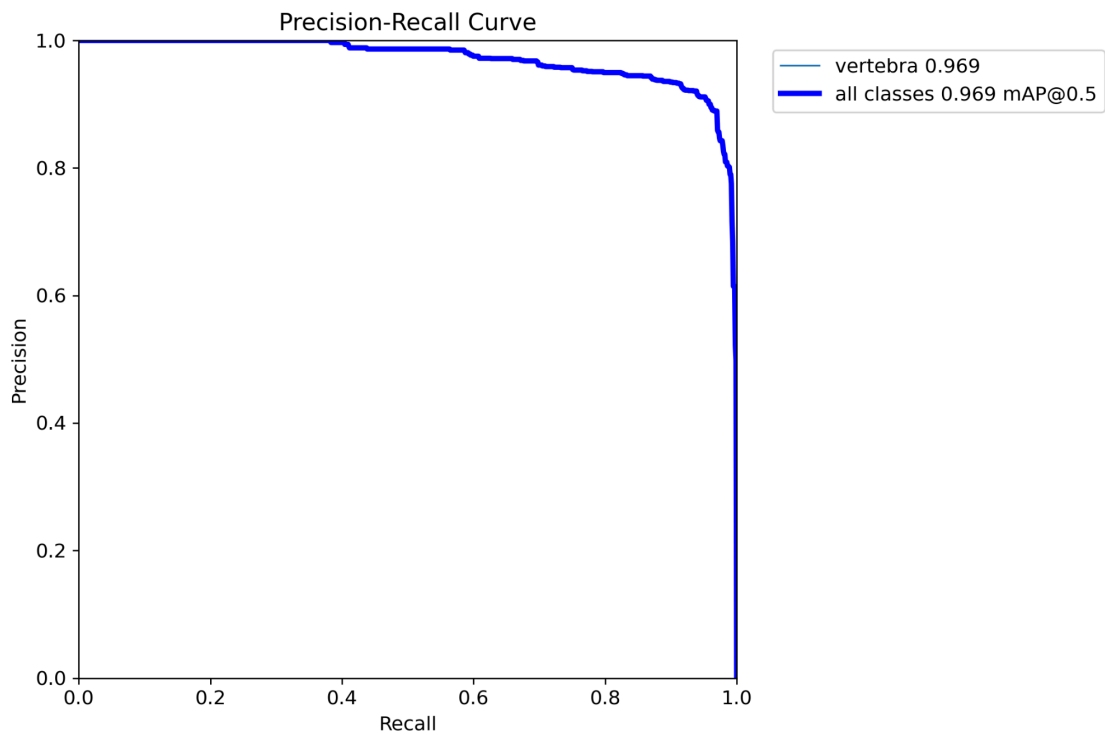
O treinamento foi interrompido automaticamente por *early stopping* na época 48, uma vez que não houve melhoria significativa nas métricas em 12 épocas consecutivas. O melhor modelo foi salvo na época 36, apresentando o equilíbrio ideal entre precisão e *recall*, com valores de precisão consistentes.

Para complementar a análise quantitativa, foram gerados gráficos de métricas durante o treinamento, permitindo uma interpretação visual do desempenho do modelo. A Figura 24 mostra a curva *Recall-Confidence*, que evidencia a relação entre a confiança das predições e a taxa de recuperação, mostrando como o modelo equilibra detecção e confiança das predições.

Figura 24 - Curva *Recall-Confidence* do modelo YOLO

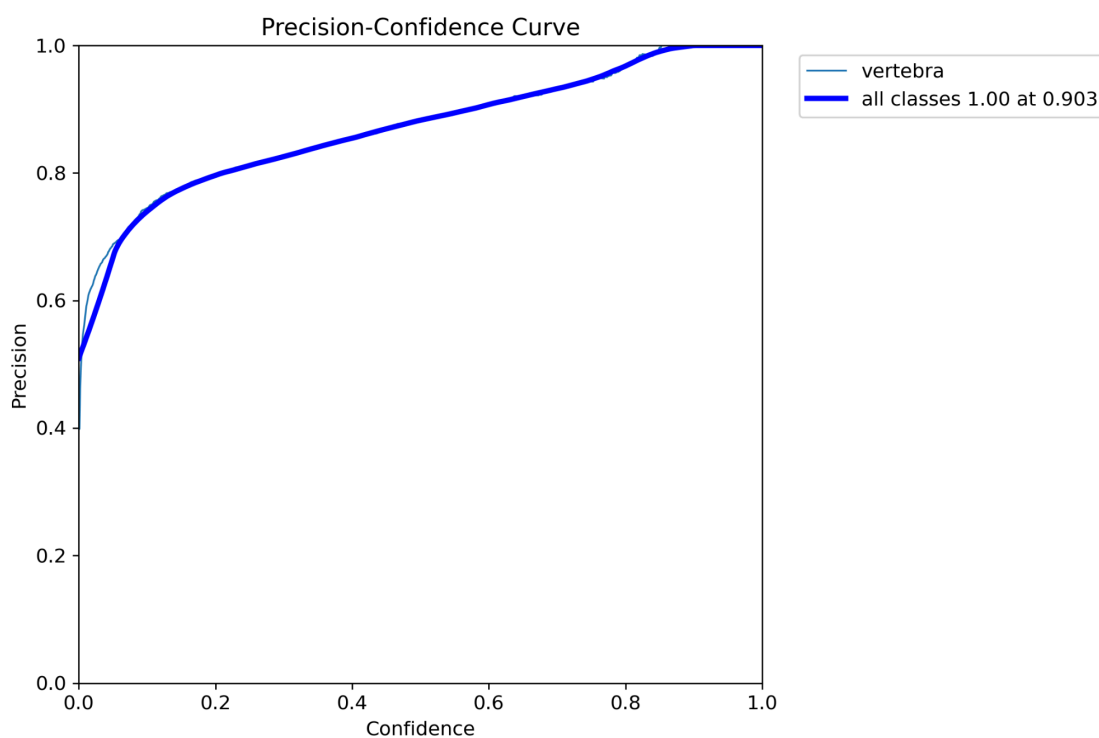
Fonte: Autores, 2025.

A Figura 25 ilustra a correlação entre precisão e *recall*, com $mAP@0.5$ de 0,969, permitindo identificar o ponto de melhor equilíbrio entre essas duas métricas.

Figura 25 - Curva *Precision-Recall* do modelo YOLO

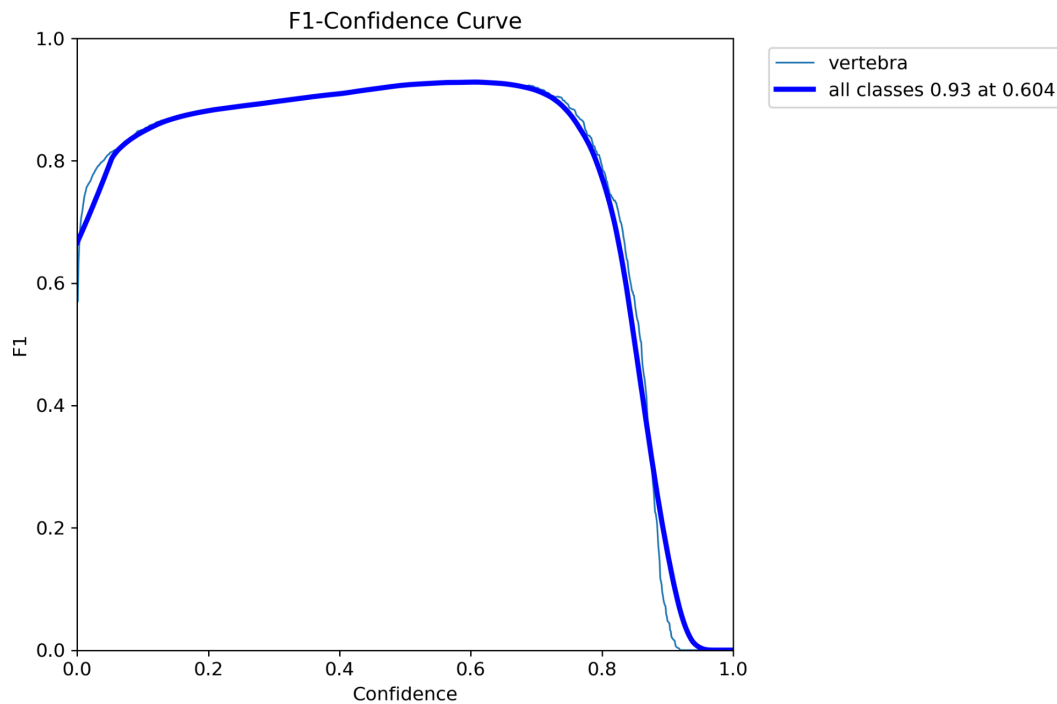
Fonte: Autores, 2025.

A curva *Precision-Confidence* representada na Figura 26 avalia a precisão do modelo em diferentes limiares de confiança, indicando em quais intervalos o modelo é mais confiável.

Figura 26 - Curva *Precision-Confidence* do modelo YOLO

Fonte: Autores, 2025.

Já a Figura 27 mostra o equilíbrio entre precisão e *recall*, considerando o valor máximo da métrica F1, sendo útil para identificar a confiança ideal das predições.

Figura 27 - Curva *Precision-Confidence* do modelo YOLO

Fonte: Autores, 2025.

5.2 Modelo de Ângulo de Inclinação de Vértexes

Para a etapa de predição do ângulo de inclinação das vértebras, foi desenvolvida uma CNN com duas camadas convolucionais seguidas de camadas totalmente conectadas, configurada para um problema de regressão.

O modelo foi treinado por 50 épocas, utilizando o otimizador Adam com taxa de aprendizado de 0,001 e função de perda *Mean Squared Error* (MSE).

Durante o treinamento, observou-se uma redução consistente da perda, indicando aprendizado progressivo da rede. O valor da *loss* caiu de aproximadamente 374,35 na primeira época para 38,91 na última, mostrando uma melhora significativa na capacidade de ajuste do modelo aos dados.

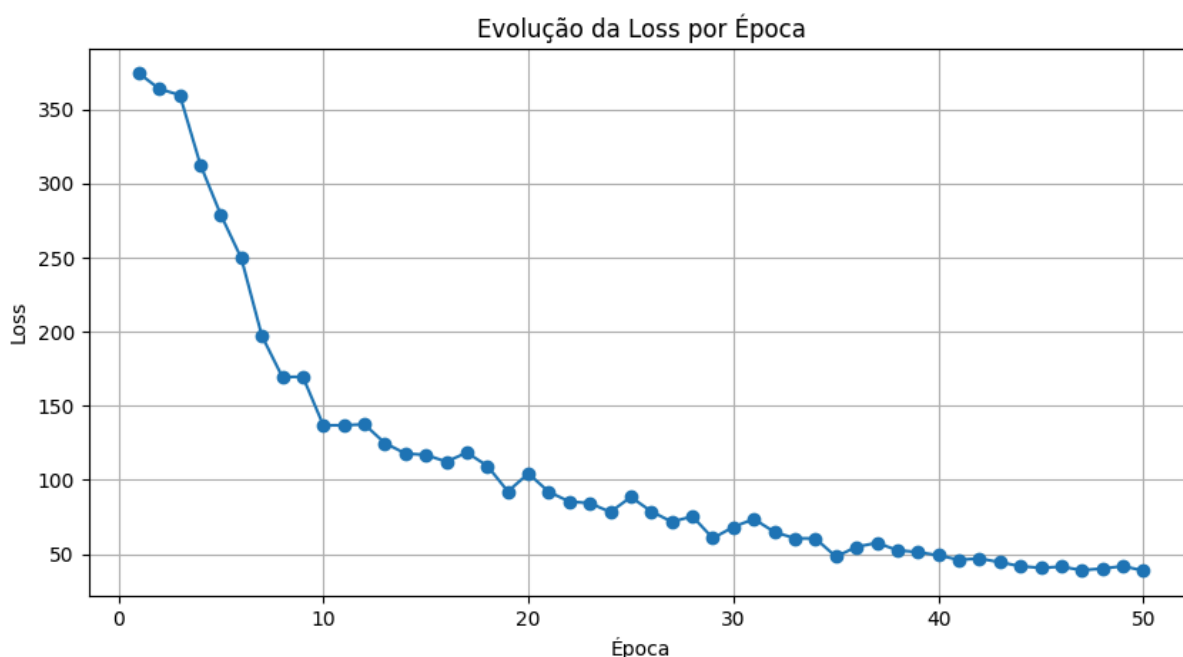
As principais métricas calculadas foram:

- **MSE:** Mede o erro quadrático médio entre as predições e os valores reais, penalizando mais fortemente grandes discrepâncias (HUANG et al., 2022). A perda final do modelo foi de 38,91, indicando boa estabilidade no aprendizado;

- **Mean Absolute Error (MAE):** Representa o erro absoluto médio entre o ângulo previsto e o ângulo verdadeiro (KOTU; DESHPANDE, 2019). É uma métrica intuitiva para interpretar o desempenho em graus. O modelo obteve um MAE de 5,65°, o que significa que, em média, o desvio entre o valor previsto e o valor real foi de apenas 5,65 graus.

A combinação desses resultados sugere que o modelo foi capaz de generalizar adequadamente sobre o conjunto de teste, fornecendo previsões coerentes com os valores reais de inclinação das vértebras. A Figura 28 ilustra a curva de perda ao longo das épocas, demonstrando a convergência do modelo.

Figura 28 - Evolução da Loss por Época



Fonte: Autores, 2025.

O estudo de GSTOETTNER et al. (2007) relata que as medições do ângulo de Cobb sofrem variações de aproximadamente 4° a 8° entre observadores diferentes ou mesmo repetida por um mesmo observador, o que torna o erro médio de 5,65° considerado aceitável no contexto clínico e comparável à variabilidade humana.

5.3 API

Esta seção apresenta as funcionalidades da API desenvolvida para o sistema, responsável por realizar o gerenciamento de pacientes e médicos e a autenticação

de usuário. A API atua como intermediária entre o *frontend* e a lógica de negócio, assegurando a troca eficiente e segura de dados.

5.3.1 [POST] /api/auth/login

Esta rota é responsável por realizar a autenticação de um usuário no sistema. O médico ou administrador deve enviar suas credenciais (usuário e senha) no corpo da requisição. A API valida os dados fornecidos junto ao banco de dados e, em caso de sucesso, retorna um código 200 (*OK*). Caso as credenciais sejam inválidas, a resposta conterá um código 401 (*Unauthorized*) com mensagem de erro apropriada.

5.3.2 [POST] /api/auth/register

Esta rota permite o registro de credenciais de acesso para um médico previamente cadastrado no sistema. O administrador deve fornecer o identificador único do médico, juntamente com o login e senha desejados através do corpo da requisição. A API então retorna o código 200 (*OK*) junto aos dados da nova autenticação do usuário.

5.3.3 [DELETE] /api/auth/{user_id}

Esta rota realiza a exclusão de uma autenticação já existente de um médico. O administrador informará o identificador único do médico como parâmetro na rota. A API realiza as validações necessárias (como verificação de existência do médico para exclusão) e, em caso de sucesso, retornará o código 200 (*OK*) junto com uma mensagem de confirmação de exclusão. Caso o médico informado não seja encontrado no sistema, é retornado o código 404 (*Not Found*).

5.3.4 [POST] /api/user

Esta rota permite o registro de novos usuários no sistema, tais como médico ou paciente. Tanto para médico quanto paciente, o sistema deve fornecer dados básicos, como nome, e-mail, documento, telefone, data de nascimento, foto e tipo do usuário (se encaixando como paciente ou médico) no corpo da requisição. A API realiza as validações necessárias (como verificação de existência prévia do usuário) e, em caso de sucesso, retornará o código 200 (*OK*) junto aos dados recém

cadastrados desse novo usuário. Caso o usuário informado já exista no sistema, é retornado o código 400 (*Bad Request*), informando que o mesmo já existe.

5.3.5 [PUT] /api/user/{user_id}

Esta rota permite a alteração dos dados de um usuário já existente no sistema. O sistema fornecerá dados como nome, e-mail, documento, telefone, data de nascimento e foto no corpo da requisição, junto ao identificador único do usuário como parâmetro na rota. A API realiza as validações necessárias (como verificação da existência do usuário) e, em caso de sucesso, retornará o código 200 (*OK*), junto aos dados recém alterados do usuário. Caso o usuário informado não exista na base de dados, é retornado o código 404 (*Not Found*), informando que o mesmo não existe.

5.3.6 [GET] /api/user/{user_id}

Esta rota realiza a pesquisa de um usuário existente no sistema a partir do seu identificador único no banco de dados. O sistema fornecerá apenas este identificador único como parâmetro na rota. A API verificará a existência do usuário e, em caso de sucesso, retornará o código 200 (*OK*) junto aos dados do usuário, caso contrário, retornará o código 404 (*Not Found*), informando que o mesmo não existe.

5.3.7 [GET] /api/user/by_role/{role}

Esta rota retorna uma lista de usuários a partir do tipo, seguindo um *enum* embutido no sistema, sendo Paciente (0), Médico (1) e Administrador (2). O sistema fornecerá esse *enum* como parâmetro na rota. A API realizará a pesquisa dos usuários e, em caso de sucesso, retornará o código 200 (*OK*), junto a uma lista contendo os usuários referentes ao tipo.

5.3.8 [GET] /api/user/patients/by_doctor/{doctor_id}

Esta rota retorna uma lista de pacientes a partir do identificador único do médico. O sistema deve fornecer apenas este identificador único como parâmetro na rota. A API realiza as validações necessárias (como verificação de existência do

usuário e se o mesmo é um médico) e, se bem-sucedida, retornará o código 200 (*OK*) junto a uma lista dos pacientes daquele médico. Caso o usuário fornecido não exista no sistema, é retornado o código 404 (*Not Found*), ou se o mesmo existe, porém, não é um médico, é retornado o código 400 (*Bad Request*), com uma mensagem explicativa conforme o tipo de erro.

5.3.9 [DELETE] /api/user/{user_id}

Esta rota realiza a exclusão de um usuário existente no sistema. O sistema deve fornecer o identificador único do usuário (médico ou paciente) como parâmetro na rota. A API realiza as validações necessárias (como verificação de existência do usuário para exclusão) e, em caso de sucesso, retornará o código 200 (*OK*) junto a uma mensagem de confirmação de exclusão. Caso o usuário informado não seja encontrado no sistema, é retornado o código 404 (*Not Found*), com uma mensagem detalhando o erro.

5.3.10 Considerações Finais

Esta API representa o núcleo de comunicação entre o *frontend* e a camada de serviços responsáveis pelo processamento dos dados. Sua arquitetura foi projetada para garantir escalabilidade, segurança e facilidade de manutenção. Os *endpoints* implementados permitem a gestão completa de pacientes e médicos, assim como a autenticação e o controle de acesso dos usuários do sistema. Com base nos testes realizados, a API demonstrou desempenho estável, respostas coerentes e tratamento adequado de erros, assegurando a integridade das informações trafegadas entre as camadas do sistema. Dessa forma, sua implementação cumpre o papel de orquestrar as operações da aplicação OrthoAI, garantindo confiabilidade e eficiência na comunicação entre os componentes da aplicação.

6 CONCLUSÃO

O presente projeto alcançou com sucesso o objetivo proposto, ao desenvolver um sistema automatizado para o cálculo do ângulo de Cobb, aplicando de forma integrada técnicas de inteligência artificial e visão computacional ao diagnóstico da escoliose. Por meio da utilização da rede YOLOv8 para a detecção das vértebras, combinada com uma CNN de regressão para a análise de inclinação, tornou-se possível automatizar o cálculo do ângulo de Cobb de maneira precisa, rápida e padronizada, reduzindo significativamente a dependência do julgamento humano e a variabilidade entre observadores.

A solução desenvolvida demonstrou elevado potencial de aplicação clínica, fornecendo aos profissionais de saúde uma ferramenta eficiente e acessível para o diagnóstico e acompanhamento de pacientes com escoliose. A utilização de inteligência artificial mostrou-se especialmente relevante na otimização do tempo de análise, na minimização de erros de medição e na melhoria da qualidade e rapidez do diagnóstico, evidenciando o papel da tecnologia como suporte estratégico na prática médica.

O projeto reforça a importância da integração entre computação e saúde, demonstrando como técnicas de aprendizado de máquina podem complementar e potencializar a atuação dos profissionais, oferecendo soluções inovadoras para desafios tradicionais da medicina.

Conclui-se, portanto, que o sistema desenvolvido representa um avanço significativo no contexto do diagnóstico por imagem da escoliose, combinando precisão, eficiência e inovação tecnológica. O trabalho evidencia o papel da inteligência artificial como suporte essencial ao trabalho médico, promovendo diagnósticos mais rápidos, confiáveis e contribuindo para uma medicina mais moderna, acessível e centrada no paciente.

7 TRABALHOS FUTUROS

O desenvolvimento deste trabalho permitiu identificar diversas oportunidades de aprimoramento que podem contribuir significativamente para o avanço da pesquisa e para a aplicação prática do modelo proposto.

As melhorias sugeridas a seguir visam otimizar o desempenho técnico da inteligência artificial, ampliar a qualidade e diversidade dos dados utilizados, garantir maior segurança e integridade das informações médicas e facilitar a utilização da ferramenta por profissionais da área da saúde.

7.1 Ampliação e Qualificação dos Datasets

Para o treinamento eficaz do modelo de inteligência artificial, é imprescindível dispor de uma base de dados extensa, diversificada e devidamente tratada. No presente estudo, identificou-se a necessidade de imagens de raio-X com marcações anatômicas das vértebras e cálculo do ângulo de Cobb, além de um conjunto de imagens brutas sem tratamento. A principal limitação enfrentada foi a escassez de datasets públicos com qualidade adequada e informações estruturadas.

Assim, propõe-se como melhoria futura o investimento na busca, curadoria e padronização de bases de dados mais amplas, contemplando tanto imagens tratadas quanto não tratadas, de modo a permitir um treinamento mais robusto, confiável e representativo do modelo de IA.

7.2 Exploração de Novos Modelos de Inteligência Artificial

Verificou-se também a escassez de trabalhos científicos dedicados especificamente à detecção e análise de escoliose por meio de técnicas de aprendizado de máquina, o que limitou o embasamento teórico do projeto. Diante disso, recomenda-se, em estudos futuros, a implementação e comparação de diferentes arquiteturas de IA, como redes neurais recorrentes ou modelos probabilísticos, com o objetivo de diversificar as abordagens, ampliar o acervo científico e aprimorar a qualidade das análises desenvolvidas na área.

7.3 Previsão da Evolução da Escoliose

Uma oportunidade relevante de aprimoramento consiste em expandir as funcionalidades preditivas do modelo, permitindo estimar a progressão da escoliose com base em dados longitudinais. Essa abordagem pode incorporar variáveis clínicas e demográficas, como idade, gênero e histórico familiar, viabilizando a aplicação de técnicas de medicina de precisão. Dessa forma, o modelo passaria a oferecer suporte clínico mais assertivo, auxiliando o médico na tomada de decisões e no acompanhamento personalizado dos pacientes.

7.4 Integração Multimodal de Exames

Sugere-se, ainda, a integração de diferentes modalidades de imagem médica, combinando o raio-X com ressonância magnética, ultrassonografia e captura de movimento. Essa integração multimodal permitiria uma avaliação tridimensional da escoliose, aumentando a precisão na detecção das curvaturas e reduzindo a dependência exclusiva de exames radiográficos, resultando em diagnósticos mais completos e precisos.

7.5 Aprimoramento para Casos de Curvatura em “S”

Os casos de escoliose com curvatura em “S” apresentam maior complexidade na identificação e mensuração dos ângulos de Cobb. Para superar essa limitação, recomenda-se o treinamento do modelo com conjuntos de dados mais abrangentes e o desenvolvimento de técnicas mais avançadas de segmentação automática das vértebras, de modo a aprimorar a identificação dos pontos anatômicos de referência e aumentar a precisão nos cálculos angulares.

7.6 Implementação de Login Seguro e Controle de Acesso

Considerando o caráter sensível das informações médicas processadas, recomenda-se a implementação de um sistema de autenticação segura, com login individual e níveis de permissão diferenciados para usuários administrativos, técnicos e médicos. O uso de autenticação em dois fatores, criptografia de senhas e

controle de sessão deve ser priorizado, de modo a garantir a integridade, confidencialidade e rastreabilidade dos acessos ao sistema.

7.7 Fortalecimento da Segurança e Proteção de Dados

Além da autenticação, é fundamental investir em melhorias estruturais de segurança da informação, incluindo:

- Criptografia de ponta a ponta para armazenamento e transmissão de dados sensíveis;
- Monitoramento contínuo de vulnerabilidades e auditorias de segurança periódicas;
- Políticas de conformidade com a Lei Geral de Proteção de Dados, assegurando que todas as informações médicas e pessoais sejam tratadas de forma ética e segura;
- Mecanismos de anonimização de dados, especialmente em datasets utilizados para treinamento da IA.

Essas medidas fortalecem a confiabilidade e a governança digital da ferramenta, assegurando seu uso responsável em contextos clínicos e acadêmicos.

7.8 Geração e Edição de Relatórios Automatizados

Propõe-se a implementação de um módulo de emissão automática de relatórios, que apresente os resultados da análise do modelo de IA de forma estruturada, interpretável e visualmente clara.

Como aprimoramento adicional, sugere-se que esses relatórios sejam editáveis, permitindo ao profissional responsável inserir observações clínicas, ajustes manuais ou anotações complementares antes da emissão final. Essa funcionalidade visa tornar a ferramenta mais flexível e adaptada à prática médica, favorecendo a comunicação entre a IA e o especialista.

7.9 Aprimoramento dos Relacionamentos do Modelo de Dados

Por fim, o modelo de entidade e relacionamento atualmente implementado no sistema adota uma estrutura que prioriza a simplicidade operacional, utilizando

relações diretas entre médicos e pacientes, nas quais cada paciente é vinculado a um único médico responsável. Embora essa abordagem seja funcional para o protótipo proposto, existem oportunidades significativas de evolução para tornar o sistema mais flexível, escalável e aderente a cenários reais da prática clínica.

Como melhoria futura, propõe-se o aprimoramento dos relacionamentos entre as tabelas, permitindo a implementação de vínculos do tipo N:N entre médicos e pacientes. Esse avanço possibilitaria situações em que um paciente é acompanhado por uma equipe multiprofissional ou em que mais de um médico acessa e avalia os mesmos exames. Essa flexibilização também permitiria incluir outros perfis profissionais, como fisioterapeutas e especialistas em coluna, ampliando o escopo de uso do sistema.

Além disso, sugere-se a criação de tabelas intermediárias para controle granular de permissões, auditoria de acesso e histórico de visualizações, garantindo maior segurança e rastreabilidade. Esse aprimoramento tornaria o banco de dados mais robusto, coerente com práticas avançadas de governança de dados, e prepararia a aplicação para cenários de uso em ambientes clínicos colaborativos e de maior complexidade.

REFERÊNCIAS

AANS. **Scoliosis**. Disponível em: <<https://www.aans.org/patients/conditions-treatments/scoliosis/>>. Acesso em: 8 abr. 2025.

AHUJA, A. S. The impact of artificial intelligence in medicine on the future role of the physician. **PeerJ**, v. 7, n. 7702, p. e7702, 4 out. 2019.

ALZUBAIDI, L. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. **Journal of Big Data**, v. 8, n. 1, p. 1–74, 31 mar. 2021.

ANNALA, L. et al. PRACTICAL APPROACH FOR HYPERSPECTRAL IMAGE PROCESSING IN PYTHON. **The international archives of the photogrammetry, remote sensing and spatial information sciences/International archives of the photogrammetry, remote sensing and spatial information sciences**, v. XLII-3, p. 45–52, 30 abr. 2018.

ASHER, A. **Why the Cobb Angle Is Used to Diagnose Scoliosis**. Disponível em: <<https://www.verywellhealth.com/cobbs-angle-and-scoliosis-296583>>. Acesso em: 8 ago. 2025.

AXIOS. **Getting Started | Axios Docs**. Disponível em: <<https://axios-http.com/docs/intro>>. Acesso em: 18 ago. 2025.

BELL, D. **Explore the UML sequence diagram**. Disponível em: <<https://developer.ibm.com/articles/the-sequence-diagram/>>. Acesso em: 18 abr. 2025.

BELL, D. **The class diagram**. Disponível em: <<https://developer.ibm.com/articles/the-class-diagram/>>. Acesso em: 19 set. 2025.

BOCHKOVSKIY, A.; WANG, C.-Y.; LIAO, H.-Y. M. YOLOv4: Optimal Speed and Accuracy of Object Detection. **arXiv**, v. 1, 22 abr. 2020.

BROWNLEE, J. **Deep Learning for Computer Vision**. [s.l.] Machine Learning Mastery, 2019.

CHEN, P. P. S. **The entity-relationship model : toward a unified view of data**. Cambridge, Mass.: M.I.T. Center for Information Systems Research, 1977.

CLARK, J. A. **Pillow — Pillow (PIL Fork) 8.0.1 documentation**. Disponível em: <<https://pillow.readthedocs.io/en/stable>>. Acesso em: 18 set. 2025.

COBB, J. Outline for the study of scoliosis. **Journal of Bone and Joint Surgery**, v. 5, p. 261–275, 1948.

COSTA, L. **IA entra na equação: como lidar com a novidade nas escolas**. Disponível em: <https://www.correiobraziliense.com.br/euestudante/educacao-basica/2025/09/7240121-ia-entra-na-equacao-como-lidar-com-a-novidade-nas-escolas.html?utm_source>. Acesso em: 3 out. 2025.

CTC TECH. **Inteligência artificial na saúde**. Disponível em: <<https://ctctech.com.br/blog/inteligencia-artificial-na-saude/>>. Acesso em: 3 set. 2025.

DAVENPORT, T.; KALAKOTA, R. The Potential for Artificial Intelligence in Healthcare. **Future Healthcare Journal**, v. 6, n. 2, p. 94–98, jun. 2019.

EKMEKCI, B.; MCANANY, C. E.; MURA, C. An Introduction to Programming for Bioscientists: A Python-Based Primer. **PLOS Computational Biology**, v. 12, n. 6, p. e1004867, 7 jun. 2016.

ELECTRON. **Introduction | Electron**. Disponível em: <<https://www.electronjs.org/docs/latest>>. Acesso em: 15 out. 2025.

ESTEVA, A. et al. Dermatologist-level Classification of Skin Cancer with Deep Neural

Networks. **Nature**, v. 542, n. 7639, p. 115–118, 25 jan. 2017.

ESTEVA, A. et al. A Guide to Deep Learning in Healthcare. **Nature Medicine**, v. 25, n. 1, p. 24–29, jan. 2019.

FOUNDER, M. **Generating PDFs from HTML with jsPDF and javascript**. Disponível em: <<https://pdfforge.com/blog/generating-pdfs-from-html-with-jspdf>>. Acesso em: 22 out. 2025.

FRAIWAN, M.; AUDAT, Z.; MANASREH, T. A dataset of scoliosis, spondylolisthesis, and normal vertebrae X-ray images. **data.mendeley.com**, v. 1, 17 jan. 2022.

FREITAS, E. L. S. X. et al. Inteligência Artificial para Educação: Um Caminho para um Campo mais Inclusivo. **Revista Brasileira de Informática na Educação**, v. 31, p. 307–322, 25 jun. 2023.

FU, X. et al. The prevalence of scoliosis among adolescents in China: a systematic review and meta-analysis. **Journal of Orthopaedic Surgery and Research**, v. 19, n. 1, 28 set. 2024.

GEEKSFORGEEKS. **NodeJS Introduction**. Disponível em: <<https://www.geeksforgeeks.org/node-js/node-js-introduction/>>. Acesso em: 22 set. 2025.

GONZALEZ, R. C.; WOODS, R. E. **Digital image processing**. Uttar Pradesh, India: Pearson, 2018.

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

GOOGLE. **Colaboratory – Google**. Disponível em: <<https://research.google.com/colaboratory/faq.html>>. Acesso em: 8 set. 2025.

GSTOETTNER, M. et al. Inter- and intraobserver reliability assessment of the Cobb angle: manual versus digital measurement tools. **European Spine Journal**, v. 16, n. 10, p. 1587–1592, 5 jun. 2007.

HICKS, S. A. et al. On evaluation metrics for medical applications of artificial intelligence. **Scientific Reports**, v. 12, n. 1, p. 5979, 8 abr. 2022.

HUANG, S. et al. Deep learning for image colorization: Current and future prospects. **Engineering Applications of Artificial Intelligence**, v. 114, n. 1, p. 105006, set. 2022.

JONES, J.; THUAIMER, A. **Cobb angle**. Disponível em: <<https://radiopaedia.org/articles/cobb-angle>>. Acesso em: 24 ago. 2025.

JSON.ORG. **JSON**. Disponível em: <<https://www.json.org/json-en.html>>. Acesso em: 19 ago. 2025.

KHALIFA, M.; ALBADAWY, M. AI in diagnostic imaging: Revolutionising accuracy and efficiency. **Computer Methods and Programs in Biomedicine Update**, v. 5, n. 100146, p. 100146–100146, 2024.

KOTU, V.; DESHPANDE, B. Chapter 12 - Time Series Forecasting. In: KOTU, V.; DESHPANDE, B. (Eds.). **Data Science**. Burlington, Massachusetts: Morgan Kaufmann, 2019. p. 395–445.

KUKARTSEV, V. V. et al. **Deep Learning for Object Detection in Images Development and Evaluation of the YOLOv8 Model Using Ultralytics and Roboflow Libraries**. Disponível em: <<https://www.researchgate.net/publication/385157595>>. Acesso em: 21 set. 2025.

KUMAR, S. et al. A Review of 3D Modalities Used for the Diagnosis of Scoliosis. **Tomography**, v. 10, n. 8, p. 1192–1204, 2 ago. 2024.

LEAPCELL. **FastAPI + Uvicorn = Blazing Speed: The Tech Behind the Hype.** Disponível em: <<https://leapcell.medium.com/fastapi-uvicorn-blazing-speed-the-tech-behind-the-hype-a205af1d9ee4>>. Acesso em: 21 set. 2025.

LEHNERT-SCHROTH, C.; AUNER-GRÖBL, P. **Tratamento tridimensional da escoliose: método Schroth de ortopedia respiratória.** Tradução: Renate Müller. 1. ed. [s.l.] Estação Liberdade, 2023.

LITJENS, G. et al. A Survey on Deep Learning in Medical Image Analysis. **Medical Image Analysis**, v. 42, n. 1, p. 60–88, dez. 2017.

MARQUES, G. **Redes neurais convolucionais na classificação de imagens médicas - Profissão Biotec.** **Profissão Biotec**, 4 out. 2023. Disponível em: <<https://profissaobiotec.com.br/redes-neurais-convolucionais-classificacao-imagens-medicas>>. Acesso em: 2 out. 2025

MDN. **Working with JSON - Learn web development | MDN.** Disponível em: <https://developer.mozilla.org/en-US/docs/Learn_web_development/Core/Scripting/JSON>. Acesso em: 27 set. 2025.

MORRISSY, R. T. et al. Measurement of the Cobb angle on radiographs of patients who have scoliosis. Evaluation of intrinsic error. **The Journal of Bone & Joint Surgery**, v. 72, n. 3, p. 320–327, mar. 1990.

NEGRINI, S. et al. 2016 SOSORT guidelines: orthopaedic and rehabilitation treatment of idiopathic scoliosis during growth. **Scoliosis and Spinal Disorders**, v. 13, n. 1, 10 jan. 2018.

NODE.JS. **About.** Disponível em: <<https://nodejs.org/en/about>>. Acesso em: 2 out. 2025.

NODESOURCE. **How Node.js Works: A Comprehensive Guide in 2025.** Disponível em: <<https://nodesource.com/blog/how-nodejs-works>>. Acesso em: 8 out. 2025.

OLIPHANT, T. E. **Guide to NumPy**. [s.l.] CreateSpace Independent Publishing Platform, 2015.

OLIVEIRA, J. R. L. et al. AVALIAÇÃO DE DESVIOS ESCOLIÓTICOS COM A UTILIZAÇÃO DE INTELIGÊNCIA ARTIFICIAL. **Revista CPAQV - Centro de Pesquisas Avançadas em Qualidade de Vida**, v. 14, n. v14n2, p. 1–1, 14 dez. 2022.

PENHA, P. J. et al. Prevalence of Adolescent Idiopathic Scoliosis in the State of São Paulo, Brazil. **Spine**, v. 43, n. 24, p. 1710–1718, 15 dez. 2018.

PINHO, D. **Gerando PDFs com JS - iMasters - We are Developers**. Disponível em: <<https://imasters.com.br/front-end/gerando-pdfs-com-js>>. Acesso em: 22 out. 2025.

PRESTIGIACOMO, F. G. et al. Intra- and inter-observer reliability of Cobb angle measurements in patients with adolescent idiopathic scoliosis. **Spine Deformity**, v. 10, n. 1, p. 79–86, 1 jan. 2022.

PRUIJS, J. E. H. et al. Variation in Cobb angle measurements in scoliosis. **Skeletal Radiology**, v. 23, n. 7, p. 517–520, out. 1994.

RAMÍREZ, S. **FastAPI**. Disponível em: <<https://fastapi.tiangolo.com>>. Acesso em: 8 out. 2025.

RASCHKA, S.; MIRJALILI, V. **Python machine learning : machine learning and deep learning with python, scikit-learn, and tensorflow 2**. Birmingham: Packt Publishing, Limited, 2019.

REACT NATIVE. **Introduction · React Native**. Disponível em: <<https://reactnative.dev/docs/getting-started>>. Acesso em: 18 out. 2025.

REACT NAVIGATION. **React Navigation: Routing and navigation for your React Native apps. 2025.** Disponível em:

<<https://reactnavigation.org/docs/hello-react-navigation>>. Acesso em: 18 out. 2025.

REDMON, J. et al. You Only Look Once: Unified, Real-Time Object Detection. **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**, p. 779–788, 2016.

SANT'ANA, H. G. F. Modelo computacional baseado em conhecimento para avaliação postural tridimensional. **Universidade Federal de Alagoas**, 22 maio 2017.

SANTOS, M. K. et al. Artificial intelligence, machine learning, computer-aided diagnosis, and radiomics: advances in imaging towards to precision medicine. **Radiologia Brasileira**, v. 52, n. 6, p. 387–396, 1 dez. 2019.

SAWARKAR, K.; ARREMSETTY, D. **Deep learning with PyTorch Lightning : build and train high-performance artificial intelligence and self-supervised models using Python**. Birmingham: Packt Publishing, 2021.

SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on Image Data Augmentation for Deep Learning. **Journal of Big Data**, v. 6, n. 1, 6 jul. 2019.

SOMMERVILLE, LAN. **Engenharia De Software**. 10. ed. London, UK: Pearson, 2019.

SOUZA, G. N. et al. Deep Learning Aplicado na Classificação de Doenças em Frutos do Cacaueiro. **SBC Open Lib**, p. 25–31, 26 set. 2024.

SPERANDIO, F. F.; TAVARES, S.; SANTOS, G. M. Reprodutibilidade Intra e Interobservadores no Ângulo de Cobb em Indivíduos Escolióticos. **Revista Brasileira de Crescimento e Desenvolvimento Humano**, v. 21, n. 1, p. 21–21, 1 abr. 2011.

STRYKER, C.; KAVLAKOGLU, E. **Inteligência artificial**. Disponível em: <<https://www.ibm.com/br-pt/think/topics/artificial-intelligence>>. Acesso em: 12 ago.

2025.

SUBRAMANIAN, V. **Deep Learning with PyTorch**. [s.l.] Packt Publishing Ltd, 2018.

TEIXEIRA, R. **Ângulo De Cobb: Diagnóstico Para Os Casos De Escoliose**. Disponível em: <<https://drricardoteixeira.com.br/angulo-de-cobb-diagnostico-para-os-casos-de-escoliose>>. Acesso em: 24 ago. 2025.

TELEMEDICINA. **Inteligência Artificial na medicina – Portal Telemedicina**. Disponível em: <<https://portaltelemedicina.com.br/inteligencia-artificial-na-medicina>>. Acesso em: 2 out. 2025.

TILKOV, S.; STEVE. Node. js: Using JavaScript to build high-performance network programs. **IEEE Internet Computing**, v. 14, n. 6, p. 80–83, 2010.

UVICORN. **Uvicorn**. Disponível em: <<https://www.uvicorn.org/>>. Acesso em: 22 out. 2025.

VARELLA, D. **Escoliose**. Disponível em: <<https://drauziovarella.uol.com.br/doencas-e-sintomas/escoliose/>>. Acesso em: 23 out. 2025.

VARNOSFADERANI, S. M.; FOROUZANFAR, M. The Role of AI in Hospitals and Clinics: Transforming Healthcare in the 21st Century. **Bioengineering**, v. 11, n. 4, p. 1–38, 2024.

WONG, J. C. et al. Validation of an artificial intelligence-based method to automate Cobb angle measurement on spinal radiographs of children with adolescent idiopathic scoliosis. **European Journal of Physical and Rehabilitation Medicine**, v. 59, n. 4, 1 set. 2023.

YUAN, X. et al. Generalized Batch Normalization: Towards Accelerating Deep Neural

Networks. **Proceedings of the AAAI Conference on Artificial Intelligence**, v. 33, p. 1682–1689, 17 jul. 2019.