

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO GRANDE DO SUL
ESCOLA POLITÉCNICA
BACHARELADO EM SISTEMAS DE INFORMAÇÃO**

**OPENECG: DESENVOLVIMENTO
DE PLATAFORMA DE
INTERPRETAÇÃO DE
ELETROCARDIOGRAMA
ASSISTIDA POR INTELIGÊNCIA
ARTIFICIAL**

PEDRO SOARES PINTO ZART

Trabalho de Conclusão II apresentado
como requisito parcial à obtenção do
grau de Bacharel em Sistemas de
Informação na Pontifícia Universidade
Católica do Rio Grande do Sul.

Orientador: Prof. Azriel Majdenbaum

**Porto Alegre
2024**

DEDICATÓRIA

Dedico este trabalho a meus pais, por terem me criado, me apoiado, me incentivado e me amado como o fizeram, sem Ronald e Tatiana não tem Pedro. Obrigado. Aos meus professores do primeiro ao último semestre do curso de graduação pela transmissão de conhecimento e tolerância com os dias de falta, que raros não foram. A universidade Pontifícia Universidade Católica do Rio Grande do Sul pela estrutura do campus e pela abordagem com os alunos durante tempos de pandemia e enchentes. A meu orientador e professor Azriel por tanto ensinamento e parceria durante as aulas e o trabalho de conclusão de curso, sou muito grato por ter sido teu aluno. E por fim, a minha namorada Sofia que esteve comigo durante essa caminhada, sempre ao meu lado para me ouvir e consolar depois de uma nota baixa ou trabalho atrasado, obrigado meu amor.

“Men are only as good as their technical development allows them to be.”
(George Orwell)

OPENECG: DESENVOLVIMENTO DE PLATAFORMA DE INTERPRETAÇÃO DE ELETROCARDIOGRAMA ASSISTIDA POR INTELIGÊNCIA ARTIFICIAL

RESUMO

O OpenECG é uma plataforma de interpretação de exames de eletrocardiograma (ECG) assistida por inteligência artificial, desenvolvida para oferecer diagnósticos rápidos e acessíveis. Utilizando um modelo de IA treinado com base nas diretrizes da Sociedade Brasileira de Cardiologia (SBC) sobre Análise e Emissão de Laudos Eletrocardiográficos, o sistema automatiza a análise inicial de exames de ECG, identificando padrões e anomalias cardíacas, como arritmias e isquemias e fornece uma interpretação para avaliação do especialista.

Palavras-Chave: Eletrocardiograma, Inteligência Artificial, Diagnóstico Assistido, OpenECG.

OPENECG: DEVELOPMENT OF AN ARTIFICIAL INTELIGENCE ASSISTED ELETROCARDIOGRAM INTERPRETATION PLATFORM

ABSTRACT

OpenECG is a platform for interpreting electrocardiogram (ECG) exams, assisted by artificial intelligence, designed to provide fast and accessible diagnoses. Utilizing an AI model trained according to the guidelines of the Brazilian Society of Cardiology (SBC) on Electrocardiographic Analysis and Report Issuance, the system automates the initial analysis of ECG exams by identifying cardiac patterns and anomalies, such as arrhythmias and ischemias, and provides an interpretation for specialist evaluation.

Keywords: Electrocardiogram, Artificial Intelligence, Diagnostic Assistance, OpenECG.

LISTA DE FIGURAS

Figura 2.1 – Willem Einthoven e o primeiro registro eletrocardiográfico	13
Figura 2.2 – Análise do traçado eletrocardiográfico	14
Figura 2.3 – O Turco, gravura de 1789 de Joseph Friedrich.	18
Figura 2.4 – Cubo de escala dos microsserviços.	20
Figura 7.1 – Jornada do usuário Paciente.	32
Figura 7.2 – Jornada do usuário Médico.	33
Figura 7.3 – Tela de Login.	34
Figura 7.4 – Tela de Listagem de Exames do usuário Paciente.	35
Figura 7.5 – Tela de Upload de exame.	35
Figura 7.6 – Tela de Exame do Paciente.	36
Figura 7.7 – Tela de Listagem de Exames do usuário Médico.	36
Figura 7.8 – Tela de Exame do usuário Médico.	37
Figura 7.9 – Trecho de código do micro serviço que lista os exames.	38
Figura 7.10 – Trecho de código do micro serviço que solicita interpretação ao GPT.	39
Figura 7.11 – Diagrama Entidade e Relacionamento do banco de dados do projeto	40
Figura 7.12 – Trecho do código que valida e processa o arquivo recebido.	41
Figura 7.13 – Trecho do código que recupera a imagem e envia para o Assistente	42
Figura 7.14 – Tela de Exame, onde o usuário visualiza o ECG e a interpretação	43
Figura 7.15 – Trecho do código com o prompt enviado ao Assistente	44
Figura 7.16 – Trecho do código que interage com a API	46
Figura 8.1 – Primeira questão do questionário.	48
Figura 8.2 – Resposta para primeira questão do questionário.	49
Figura 8.3 – Gráfico de assertividade entre o profissional e a IA.	50
Figura 8.4 – Tabela de preços da OpenAI	52
Figura 8.5 – Tabela de preços da OpenAI Assistant	53
Figura 8.6 – Comparação respostas especialista e OpenECG	53

LISTA DE SIGLAS

ECG – Eletrocardiograma

GPT – Generative Pre-Trained Transformer

IA – Inteligência Artificial

DICOM – Digital Imaging and Communications in Medicine

PACS – Picture Archiving and Communication System

RIS – Radiology Information System

HIS – Hospital Information System

ID – Identificador

SBC – Sociedade Brasileira de Cardiologia

PLN – Processamento de Linguagem Natural

API – Application Programming Interface

HTTP – Hypertext Transfer Protocol

CD – Compact Disk

USB – Universal Serial Bus

RIS – Radiology Information System

HIS – Hospital Information System

VSC – Visual Studio Code

DOM – Document Object Model

HTML – HyperText Markup Language

SPA – Single Page Applications

SFC – Single File Component

GDPR – General Data Protection Regulation

HIPAA – Health Insurance Portability and Accountability Act

TEP – Tromboembolismo Pulmonar

SUMÁRIO

1	INTRODUÇÃO	11
2	REFERENCIAL TEÓRICO	13
2.1	ELETROCARDIOGRAMA (ECG)	13
2.2	NOÇÕES BÁSICAS DO ECG	14
2.2.1	SÍNDROMES ISQUÊMICAS	15
2.2.2	ARRITMIAS CARDÍACAS	15
2.2.3	SÍNDROMES CARDÍACAS	15
2.2.4	OUTRAS PATOLOGIAS	16
2.3	PROCESSAMENTO DE IMAGEM	16
2.4	INTELIGÊNCIA ARTIFICIAL (IA)	17
2.5	MICROSSERVIÇOS	19
3	PROBLEMA	21
4	OBJETIVO	22
4.1	OBJETIVO GERAL	22
4.2	OBJETIVOS ESPECÍFICOS	22
4.3	MOTIVAÇÃO	23
5	TRABALHOS RELACIONADOS	25
5.1	DIGITAL IMAGING AND COMMUNICATIONS IN MEDICINE (DICOM)	25
5.2	PICTURE ARCHIVING AND COMMUNICATION SYSTEMS (PACS)	25
5.3	POWER MEDICAL RADIO (PM RADIO)	26
5.4	VISION	27
6	FERRAMENTAS E TECNOLOGIAS UTILIZADAS	28
6.1	FLASK	28
6.2	DOCKER	28
6.3	MYSQL	28
6.4	GPT ASSISTANT	29
6.5	NGINX	29
6.6	POSTMAN	29

6.7	VISUAL STUDIO CODE (VSC)	30
6.8	VUE	30
6.9	PYTHON	30
7	ANÁLISE DO SISTEMA	32
7.1	FRONTEND	32
7.1.1	LOGIN	33
7.1.2	LISTAGEM DE EXAMES DO USUÁRIO PACIENTE	34
7.1.3	TELA DE ENVIO DO EXAME	35
7.1.4	TELA DE VISUALIZAÇÃO E INTERPRETAÇÃO DO EXAME DO USUÁRIO PACIENTE (/EXAM/EXAMID)	35
7.1.5	LISTAGEM DE EXAME DO MÉDICO	36
7.1.6	TELA DE VISUALIZAÇÃO E INTERPRETAÇÃO DO EXAME DO USUÁRIO MÉDICO	36
7.2	API GATEWAY (NGINX)	37
7.3	BACKEND	37
7.4	BANCO DE DADOS	39
7.5	FLUXO DOS DADOS	40
7.5.1	UPLOAD E ARMAZENAMENTO DE IMAGEM	41
7.5.2	PROCESSAMENTO E INTERPRETAÇÃO DA IMAGEM	42
7.5.3	CONSULTA E VISUALIZAÇÃO	42
7.6	ASSISTENTE GPT	43
7.6.1	LISTA DE INSTRUÇÕES	44
7.6.2	ASSISTANT API	46
8	ANÁLISE HOMEM X MÁQUINA	47
8.1	ANÁLISE	47
8.2	ASSERTIVIDADE	47
8.3	TEMPO DE INTERPRETAÇÃO	51
8.4	PRECIFICAÇÃO	51
9	TRABALHOS FUTUROS	55
9.1	APRIMORAMENTO DA PRECISÃO DA INTELIGÊNCIA ARTIFICIAL	55
9.2	EXPANSÃO DOS TIPOS DE EXAMES INTERPRETADOS	55
9.3	MELHORIA DA SEGURANÇA DA PLATAFORMA	55

10	CONSIDERAÇÕES FINAIS.....	56
	REFERÊNCIAS	58

1. INTRODUÇÃO

Nas últimas décadas, o campo da saúde vem passando por transformações significativas, impulsionadas pelo avanço das tecnologias digitais e da inteligência artificial (IA). Uma das áreas que tem se beneficiado desse desenvolvimento é a de diagnósticos médicos, onde a IA tem sido cada vez mais empregada para auxiliar na análise de exames, como o eletrocardiograma (ECG) [Viv20]. O ECG é uma ferramenta fundamental na prática clínica, utilizada para monitorar a atividade elétrica do coração e diagnosticar condições como arritmias, isquemias e outras anomalias cardíacas. Entretanto, o processo de interpretação de um ECG é detalhado e exige certo grau de especialização, o que pode limitar o acesso de pacientes a diagnósticos rápidos e precisos, especialmente em regiões com reduzida disponibilidade de profissionais de saúde qualificados [Suk15].

É nesse cenário que o projeto OpenECG surge como proposta. Trata-se de uma plataforma integrada de interpretação de exames de ECG, baseada em inteligência artificial, que visa oferecer análises rápidas e precisas desses exames, automatizando a interpretação inicial e facilitando o trabalho de médicos e demais profissionais de saúde. Com a utilização de um modelo de IA, o OpenECG é capaz de processar imagens de exames de ECG e fornecer uma interpretação fundamentada nas diretrizes da Sociedade Brasileira de Cardiologia (SBC), representando um avanço para a área de diagnóstico cardiológico e um recurso potencialmente útil para a prática clínica.

A ideia central do OpenECG é permitir que os usuários façam o upload de uma imagem de ECG na plataforma, que utiliza IA para interpretar o exame com base em parâmetros e diretrizes pré-definidos pela SBC. O sistema busca preencher uma lacuna importante no setor de saúde, oferecendo suporte diagnóstico em tempo real e com alta confiabilidade. Com a análise automatizada, o OpenECG pode ajudar médicos e técnicos a obter uma primeira impressão sobre a condição clínica do paciente, especialmente em casos que exigem decisões rápidas e em ambientes onde o suporte especializado pode não estar prontamente disponível.

O Brasil enfrenta desafios significativos em relação ao acesso e à disponibilidade de diagnósticos rápidos e precisos, especialmente nas áreas mais afastadas dos grandes centros urbanos. A falta de cardiologistas e especialistas é uma realidade em muitas regiões [FDS21], dificultando o acesso de pacientes a diagnósticos de qualidade. Esse cenário contribui para o aumento das taxas de mortalidade e morbidade de doenças cardíacas, que figuram entre as principais causas de morte no país. A Organização Mundial da Saúde (OMS) destaca que o diagnóstico precoce é um fator crítico na redução de complicações graves relacionadas ao coração, como infarto agudo do miocárdio e arritmias fatais [AdPM17]. Esse contexto reforça a importância do diagnóstico precoce na conduta médica.

A interpretação de um ECG é um processo que exige conhecimento técnico para identificar variações e padrões associados a diferentes condições cardíacas. Atualmente, muitas instituições e clínicas não possuem especialistas para interpretar esses exames de forma imediata, o que pode resultar em atrasos no diagnóstico e no tratamento. Assim, uma ferramenta que ofereça uma análise inicial, mesmo que de forma automatizada, pode representar uma solução viável para melhorar o acesso a diagnósticos e contribuir para a detecção precoce de condições cardíacas.

Pensando nisso, o OpenECG foi criado: uma plataforma assistida por IA para a interpretação do traçado gráfico do ECG. A integração da inteligência artificial padronizada envolve a inclusão de uma base de conhecimento com as diretrizes da SBC, instruções detalhadas de operação, além de exemplos de ECG e diagnósticos esperados. O sistema foi projetado para receber imagens de ECG, processá-las e gerar uma análise automatizada que pode servir como um primeiro parecer para os profissionais de saúde.

2. REFERENCIAL TEÓRICO

Este capítulo apresenta os conceitos fundamentais para a compreensão do trabalho proposto. Os fundamentos abordam o eletrocardiograma, as anormalidades cardíacas detectadas por meio dele, a inteligência artificial empregada para auxiliar na interpretação do exame pelos médicos, e os microsserviços que constituem a estrutura base do código desenvolvido.

2.1 ELETROCARDIOGRAMA (ECG)

A história do ECG começa antes mesmo da descoberta dos raios-X, em 1895. Em 1887, o fisiologista britânico Augustus D. Waller, utilizando o eletrômetro capilar de Lippman, registrou o primeiro eletrocardiograma humano [eRMT09]. Waller demonstrou que uma atividade elétrica precedia a contração do coração, empregando eletrodos colocados no peito de um voluntário.

Juntamente com Waller, o médico e fisiologista holandês Willem Einthoven dedicou-se ao desenvolvimento de um método não invasivo para estudar o comportamento do sistema cardiovascular. Em 1901, conectando fios aos pés e mãos de um voluntário e mergulhando-os em recipientes com eletrólitos (Figura 2.1), Einthoven registrou as derivações em um aparelho, obtendo o primeiro registro eletrocardiográfico fidedigno, sem necessidade de correção matemática [eRMT09].



Figura 2.1 – Willem Einthoven e o primeiro registro eletrocardiográfico

Segundo Einthoven, seus estudos buscavam compreender “... o funcionamento do coração em detalhes e a causa de uma grande variedade de anormalidades cardíacas,

que permitirá aliviar o sofrimento de nossos pacientes” [Gua18]. Considerado o pai da eletrocardiografia, ele recebeu o Prêmio Nobel de Medicina em 1924 por suas contribuições. Além dele, Thomas Lewis e Frank N. Wilson também são destacados por suas contribuições históricas: Lewis, por suas pesquisas sobre arritmias, e Wilson, pelos estudos das derivações unipolares, que possibilitaram a padronização do sistema de 12 derivações. Este sistema, que oferece 12 perspectivas diferentes do coração, é utilizado até os dias de hoje [Gua18].

O ECG registra a atividade elétrica do músculo cardíaco por meio de um eletrocardiógrafo. O exame é realizado com 10 eletrodos conectados ao paciente, divididos entre periféricos e precordiais, permitindo a visualização do ritmo cardíaco por meio de derivações [eRMT09].

2.2 NOÇÕES BÁSICAS DO ECG

O ECG é um exame essencial que registra a atividade elétrica do coração em um gráfico. Para realizá-lo, eletrodos são colocados na pele do paciente, captando os sinais elétricos do coração, que são apresentados em um traçado composto por diferentes ondas. Este traçado inclui a Onda P, que reflete a ativação dos átrios (as câmaras superiores do coração); o Complexo QRS, que representa a ativação dos ventrículos, responsáveis pelo bombeamento de sangue; e a Onda T, que demonstra o processo de recuperação dos ventrículos após a contração [Suk15].

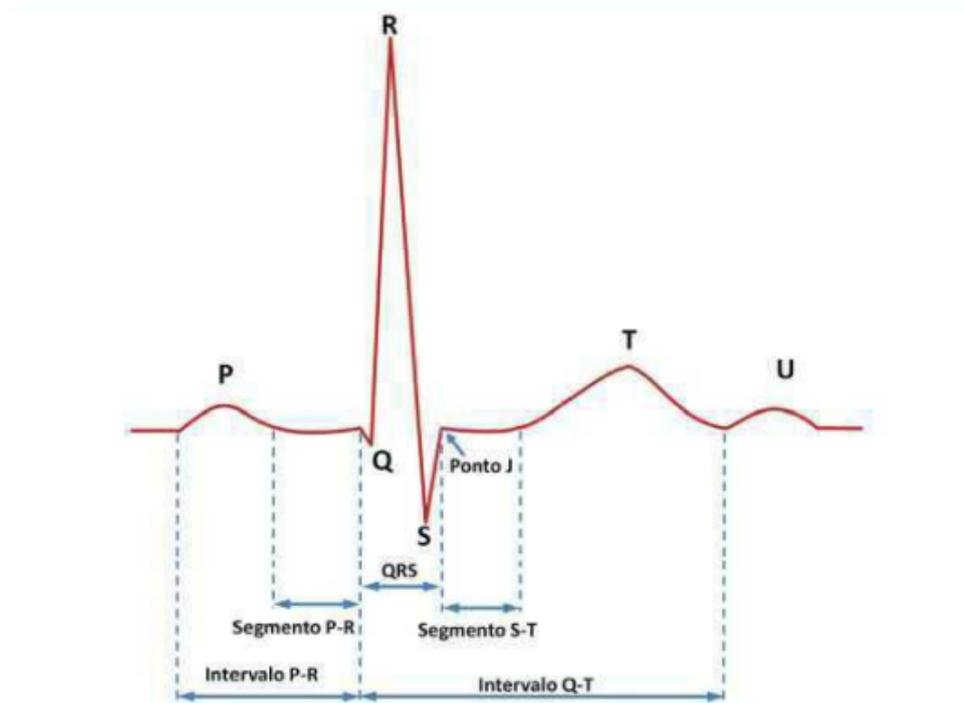


Figura 2.2 – Análise do traçado eletrocardiográfico

Cada ciclo do ECG representa uma batida do coração e é registrado em várias posições ao redor do tórax e dos braços, proporcionando uma visão detalhada da atividade elétrica cardíaca.

2.2.1 SÍNDROMES ISQUÊMICAS

As síndromes isquêmicas ocorrem quando o coração não recebe sangue suficiente, geralmente devido a obstruções nas artérias coronárias. Essas condições podem ser identificadas no ECG por manifestações específicas:

Isquemia: falta temporária de oxigênio, caracterizada por alterações na onda T. Lesão: dano ao músculo cardíaco, identificado por elevações ou depressões no segmento ST. Necrose (infarto): morte do tecido cardíaco, detectada por mudanças permanentes, especialmente no Complexo QRS. Essas características permitem diferenciar entre angina (dor temporária) e infarto (danos permanentes), possibilitando diagnósticos e tratamentos mais precisos para doenças cardíacas [Suk15].

2.2.2 ARRITMIAS CARDÍACAS

Arritmias são alterações no ritmo cardíaco, podendo ser acelerado, lento ou irregular. No ECG, essas mudanças se manifestam por variações nos padrões das ondas e nos intervalos entre elas:

Taquicardia sinusal: batimentos rápidos devido à ativação do nódulo sinusal (intervalos RR curtos). Bradicardia sinusal: batimentos lentos e regulares (intervalos RR longos). Fibrilação atrial: contrações desordenadas dos átrios, causando batimentos irregulares. Taquicardia ventricular: batimentos rápidos e desordenados nos ventrículos, com complexos QRS largos. Algumas arritmias, como a fibrilação ventricular, requerem intervenção médica imediata [Suk15].

2.2.3 SÍNDROMES CARDÍACAS

Em situações de maior demanda, o coração pode aumentar suas câmaras para suportar a carga extra. No ECG, isso aparece como:

Hipertrofia ventricular: complexos QRS mais largos. Hipertrofia atrial: alterações na onda P, indicando sobrecarga nas câmaras superiores. Problemas na condução elétrica,

como bloqueios, também são detectáveis, manifestando-se como atrasos ou ausência de certas partes do traçado [Suk15].

2.2.4 OUTRAS PATOLOGIAS

O ECG também detecta condições específicas, como:

Repolarização precoce: leve elevação no segmento ST, comum em jovens. Pericardite: inflamação no revestimento do coração, com alterações no segmento ST e na onda T. Tromboembolismo pulmonar (TEP): mudanças específicas no ECG causadas por coágulos nos pulmões. Síndrome de Wolff-Parkinson-White (WPW): via elétrica extra, resultando em batimentos rápidos. Síndrome de Brugada: condição genética associada a risco de parada cardíaca, identificada por um padrão característico no ECG. A identificação dessas condições permite tratamentos mais eficazes e direcionados [Suk15].

2.3 PROCESSAMENTO DE IMAGEM

O processamento de imagem é um campo essencial no desenvolvimento de sistemas que utilizam dados visuais para análise, interpretação e tomada de decisão, sendo amplamente aplicado em áreas como medicina, sensoriamento remoto e visão computacional. Esse processo envolve uma sequência de etapas que transformam imagens brutas em informações úteis, garantindo que os dados visuais possam ser interpretados com maior eficiência.

Inicialmente, o processamento de imagem realiza o pré-processamento, uma etapa fundamental que corrige distorções geométricas, radiométricas e topográficas. Técnicas como filtragem de ruído, correções de luminosidade e ajuste de contraste são aplicadas para melhorar a qualidade da imagem, preparando-a para as etapas subsequentes. Em seguida, ocorre a extração e seleção de características, onde variáveis relevantes, como contornos, texturas ou padrões espectrais, são identificadas para facilitar a distinção entre classes ou regiões específicas na imagem[D. 07].

Dependendo do contexto e da complexidade dos dados, diferentes métodos de classificação podem ser utilizados. Métodos tradicionais, como a classificação por pixel, baseiam-se na análise individual de cada ponto da imagem. No entanto, abordagens mais avançadas, como redes neurais e classificadores baseados em conhecimento, são empregadas para lidar com dados complexos e de alta dimensionalidade, oferecendo resultados mais robustos e precisos. Além disso, técnicas como classificação contextual consideram relações espaciais entre os pixels, enquanto a classificação subpixel trata a mistura de diferentes informações em uma única unidade de imagem[D. 07].

Dessa forma, o processamento envolve a preparação das imagens, a extração das características visuais presentes no exame e validação dos dados extraídos, facilitando a identificação de padrões relevantes para o diagnóstico. O uso de técnicas avançadas de processamento também permite tratar possíveis ruídos ou variações nas imagens, garantindo consistência nos resultados.

2.4 INTELIGÊNCIA ARTIFICIAL (IA)

A história da Inteligência Artificial (IA) remonta a períodos em que a ideia de máquinas capazes de simular o pensamento humano existia apenas no campo da imaginação e da literatura. Obras de filósofos e escritores, como René Descartes e Jules Verne, exploraram o conceito de "máquinas pensantes" muito antes de a tecnologia possibilitar sua concretização.

Foi apenas em meados do século XX que a IA começou a ganhar corpo como um campo de estudo acadêmico e científico. Em 1950, Alan Turing introduziu o conceito de máquinas inteligentes por meio do agora famoso "Teste de Turing", que buscava determinar se uma máquina poderia simular a inteligência humana a ponto de ser indistinguível de um ser humano em uma conversa. Seguindo essa linha, em 1956, o termo "Inteligência Artificial" foi oficialmente cunhado por John McCarthy durante a Conferência de Dartmouth, evento que marcou o início formal do campo [Buc06].

A IA passou por diversas fases, incluindo períodos de grande entusiasmo e investimento, seguidos por momentos de descrença e cortes de verbas – os chamados "invernos da IA". Esses ciclos ocorreram principalmente nos anos 1970 e no final dos anos 1980, quando as limitações tecnológicas e a falta de resultados práticos diminuíram o interesse e os recursos destinados ao desenvolvimento da área.

Segundo John McCarthy, um dos pioneiros na área, a Inteligência Artificial pode ser definida como "a ciência e engenharia de criar e manter máquinas inteligentes". Nas suas primeiras fases, o desenvolvimento da IA baseava-se em programar diretamente as máquinas para realizar tarefas específicas. Por exemplo, o objetivo era fazer com que a máquina seguisse instruções detalhadas para jogar xadrez ou resolver problemas matemáticos, imitando o comportamento humano por meio de uma lógica rígida e previamente definida (Figura 2.3). Esse modelo inicial dependia totalmente de regras e instruções humanas, com algoritmos que simulavam o pensamento por meio de operações lógicas estruturadas [Buc06].

Alan Turing, por sua vez, abordou a IA sob um viés filosófico e prático. Ele propôs que um computador poderia ser considerado inteligente caso fosse capaz de imitar o comportamento humano em uma situação de comunicação, a ponto de um interlocutor humano não conseguir distinguir entre uma máquina e uma pessoa. Essa ideia deu origem

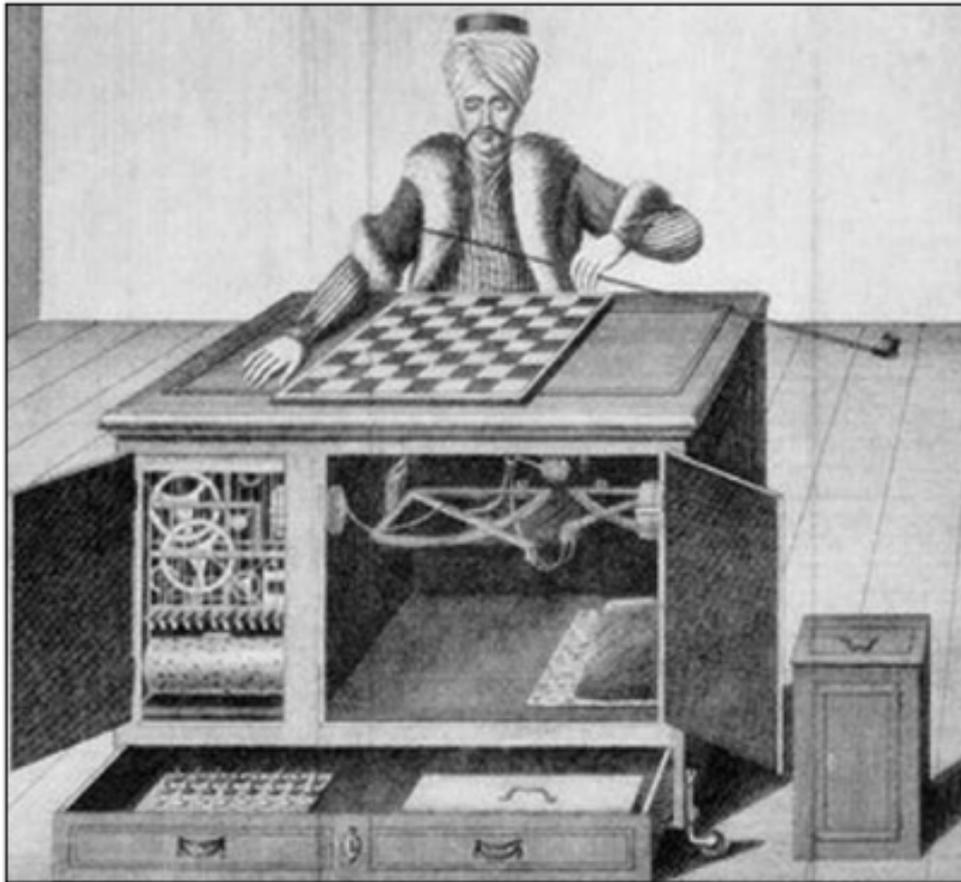


Figura 2.3 – O Turco, gravura de 1789 de Joseph Friedrich.

ao famoso "Teste de Turing", que permanece até hoje como uma referência para avaliar a capacidade de uma máquina em exibir comportamentos considerados "inteligentes".

Levando em consideração essas duas perspectivas, a IA moderna pode ser descrita como um campo multidisciplinar que busca entender e replicar aspectos do pensamento e comportamento humanos por meio de máquinas. No entanto, diferentemente das abordagens iniciais, as tecnologias atuais focam na capacidade de aprendizado autônomo, com algoritmos capazes de aprimorar suas próprias habilidades ao interagir com dados e ambientes variados. Esse conceito se materializa principalmente no aprendizado de máquina (Machine Learning), onde a IA não depende mais exclusivamente de programações rígidas, mas sim de um treinamento baseado em dados, no qual a máquina identifica padrões, formula hipóteses e ajusta seus próprios parâmetros para melhorar o desempenho.

O desenvolvimento do aprendizado profundo (Deep Learning), por exemplo, tem sido particularmente transformador para a IA moderna, pois permite que redes neurais artificiais, inspiradas na estrutura do cérebro humano, processem grandes volumes de dados e realizem inferências complexas. Essas redes, compostas por camadas de nós interconectados, "aprendem" por meio da análise de vastas quantidades de

informações, possibilitando avanços impressionantes em áreas como visão computacional, processamento de linguagem natural e reconhecimento de voz [Jim24].

Em suma, a IA atual representa um esforço contínuo para criar sistemas que não apenas executem tarefas específicas, mas que evoluam com a experiência e adaptem-se a novos desafios, imitando cada vez mais o processo de aprendizado humano.

2.5 MICROSSERVIÇOS

A arquitetura de microsserviços é um modelo de desenvolvimento de software em que uma aplicação é dividida em pequenos serviços independentes, cada um responsável por uma função específica e executando em seu próprio processo. Essa arquitetura permite que os microsserviços se comuniquem por meio de interfaces bem definidas, geralmente utilizando protocolos leves, como o Hypertext Transfer Protocol (HTTP), o que possibilita sua integração eficiente e independente dentro de um sistema maior. Essa abordagem contrasta com a arquitetura monolítica, na qual todos os componentes de uma aplicação estão interligados em uma única estrutura, o que pode gerar limitações de escalabilidade e aumentar a complexidade de manutenção.

Entre as principais características e vantagens dos microsserviços estão a independência e a descentralização. Cada serviço pode ser desenvolvido, testado e implantado de forma isolada, permitindo que equipes trabalhem simultaneamente em diferentes serviços sem afetar o funcionamento global da aplicação. Além disso, os microsserviços oferecem escalabilidade e flexibilidade, possibilitando que cada serviço seja dimensionado de acordo com suas necessidades de recursos. Esse modelo é altamente adaptável, permitindo o uso de tecnologias e linguagens de programação diferentes em cada serviço, o que facilita a atualização e a modernização contínua do sistema. Para melhor compreensão, uma ilustração comumente utilizada para explicar o particionamento de uma aplicação monolítica em microsserviços é o modelo do cubo de escala (Figura 2.4) [Dmi14].

No eixo X (horizontal), o escalonamento ocorre pela replicação de múltiplas cópias idênticas da aplicação. Um balanceador de carga distribui as solicitações entre essas cópias, garantindo que a capacidade de processamento aumente com a adição de novos servidores. No eixo Z (sharding), a aplicação é segmentada para que cada servidor processe apenas um subconjunto dos dados, utilizando proxies ou chaves primárias para encaminhar as solicitações ao servidor correto. Essa abordagem é comum em bancos de dados e serviços que precisam particionar dados entre diferentes categorias, como usuários gratuitos e pagos. Por fim, o eixo Y (decomposição funcional) representa a fragmentação de uma aplicação monolítica em serviços menores e independentes, com cada microsserviço gerenciando funcionalidades específicas. Essa decomposição reduz a

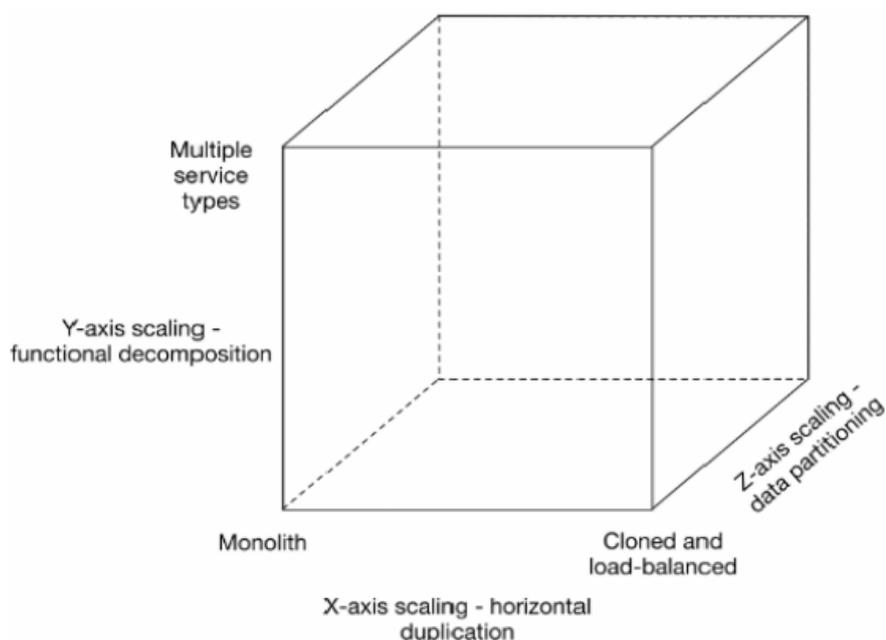


Figura 2.4 – Cubo de escala dos microsserviços.

complexidade e facilita o desenvolvimento e a manutenção do sistema. Enquanto os eixos X e Z tratam de aumentar a capacidade e a disponibilidade, o eixo Y aborda a complexidade do sistema, proporcionando modularidade e escalabilidade funcional. Assim, o modelo do cubo demonstra como essas três abordagens podem ser combinadas para criar sistemas distribuídos robustos e escaláveis [Dmi14].

A imagem descrita evidencia a complexidade dos sistemas distribuídos, que exigem coordenação e gerenciamento robusto de múltiplos serviços independentes. A comunicação entre esses serviços pode demandar transações complexas e mecanismos que garantam a consistência e a disponibilidade dos dados. Além disso, como cada serviço necessita de seu próprio ambiente de execução, o consumo de recursos pode ser maior em comparação com arquiteturas monolíticas [Dmi14].

No contexto deste projeto, a arquitetura de microsserviços permite uma flexibilidade significativa na escolha e implementação das ferramentas utilizadas para interpretar exames. Com essa arquitetura, é possível substituir ou atualizar o microsserviço responsável por uma funcionalidade específica sem afetar o restante do sistema. Por exemplo, em vez de utilizar o GPT Assistant da OpenAI para interpretar exames, o microsserviço correspondente pode ser facilmente alterado para integrar o Google AI Studio ou outra ferramenta que melhor atenda às necessidades do projeto. Essa modularidade possibilita adaptação contínua e rápida às inovações tecnológicas, mantendo a independência dos demais componentes do sistema e facilitando a atualização e o gerenciamento de cada serviço.

3. PROBLEMA

A interpretação de ECGs é um processo complexo que exige conhecimento especializado e uma análise detalhada para identificar condições cardíacas, como arritmias, síndromes isquêmicas e outras anomalias. Tradicionalmente, a interpretação de ECGs é realizada por cardiologistas ou outros profissionais da área médica, o que pode ser um desafio para pacientes que não têm acesso rápido ou fácil a esses especialistas, especialmente em regiões remotas ou com recursos limitados na área da saúde.

A falta de acesso rápido à interpretação de ECGs representa um obstáculo significativo para a detecção precoce e o tratamento adequado de condições cardíacas. Além disso, a espera por uma avaliação especializada pode aumentar o risco de progressão de condições graves, como infartos e arritmias que requerem intervenções urgentes. Nesse contexto, surge a necessidade de uma solução que permita aos pacientes e profissionais de saúde acessarem interpretações confiáveis e rápidas dos exames de ECG, reduzindo o tempo de espera e facilitando decisões médicas mais ágeis.

Considerando esses desafios, o desenvolvimento de uma plataforma digital para interpretação de exames de ECG, capaz de fornecer resultados baseados nas diretrizes da Sociedade Brasileira de Cardiologia, apresenta-se como uma solução promissora. Ao integrar uma tecnologia avançada de inteligência artificial, especificamente um assistente configurado para interpretar exames conforme normas rigorosas, a plataforma oferece uma alternativa prática para a análise de ECGs.

Assim, o problema que este trabalho busca resolver é a carência de uma solução acessível e confiável que permita a interpretação imediata de exames de ECG, fornecendo diagnósticos precisos e em tempo real, baseados nas diretrizes cardiológicas nacionais.

4. OBJETIVO

Neste capítulo, serão discutidas as motivações e os objetivos do projeto OpenECG, uma plataforma projetada para auxiliar na interpretação de exames de ECG utilizando inteligência artificial. O projeto busca atender a uma necessidade crescente por diagnósticos cardíacos rápidos e confiáveis, especialmente em regiões com carência de profissionais especializados. Os objetivos do OpenECG estão alinhados com o uso de tecnologias emergentes para promover acessibilidade e precisão em diagnósticos médicos. A seguir, são detalhados o objetivo geral e os objetivos específicos da plataforma.

4.1 Objetivo geral

O objetivo principal do projeto OpenECG é desenvolver uma plataforma de interpretação de exames de ECG baseada em inteligência artificial. Essa plataforma será capaz de analisar imagens de exames de ECG e fornecer uma interpretação inicial fundamentada nas diretrizes estabelecidas pela Sociedade Brasileira de Cardiologia. A meta é automatizar o processo de interpretação do exame, facilitando o trabalho de profissionais de saúde e auxiliando na tomada de decisões clínicas, especialmente em locais onde o acesso a cardiologistas é limitado.

4.2 Objetivos específicos

Para alcançar o objetivo geral, o projeto OpenECG define uma série de objetivos específicos, cada um contribuindo diretamente para a construção e funcionalidade da plataforma:

- **Automatizar a interpretação inicial de exames de ECG utilizando IA para analisar padrões cardíacos e identificar possíveis anomalias:**

Esse objetivo abrange o desenvolvimento e treinamento de modelos de inteligência artificial capazes de reconhecer padrões complexos em exames de ECG, como variações no ritmo cardíaco e anomalias estruturais no traçado. O uso de IA possibilita uma análise minuciosa dos dados do exame, permitindo a detecção precoce de condições como arritmias, isquemias e hipertrofias cardíacas. A automação dessa interpretação inicial fornece uma base confiável para que os profissionais de saúde realizem análises mais aprofundadas, quando necessário..

- **Oferecer interface de fácil uso para médicos, técnicos de saúde e pacientes:**

A interface da plataforma deve ser projetada com foco na usabilidade, permitindo que tanto profissionais de saúde quanto pacientes possam navegar com facilidade. O objetivo é que o upload das imagens de ECG seja rápido e intuitivo, de modo que usuários com diferentes níveis de habilidade técnica possam acessar a plataforma sem dificuldades. A experiência do usuário é um ponto central, pois uma interface amigável facilita o uso da plataforma e promove sua adoção em diversas instituições de saúde.

- **Garantir a precisão das interpretações baseando-se nas diretrizes da Sociedade Brasileira de Cardiologia:**

A interface da plataforma será projetada com foco na usabilidade, permitindo que tanto profissionais de saúde quanto pacientes naveguem de forma intuitiva. O objetivo é que o upload das imagens de ECG seja rápido e acessível, de modo que usuários com diferentes níveis de habilidade técnica possam utilizar a plataforma sem dificuldades. Uma interface amigável facilita o uso e promove a adoção da plataforma em diversas instituições de saúde.

- **Disponibilizar um recurso acessível para regiões com pouca infraestrutura médica:**

O OpenECG foi projetado para ser uma solução acessível, ampliando o alcance de diagnósticos cardiológicos para populações que enfrentam barreiras geográficas ou econômicas no acesso ao atendimento especializado. Ao oferecer uma plataforma de IA fácil de acessar, o projeto busca reduzir desigualdades no atendimento médico e fornecer uma alternativa viável para pacientes em áreas rurais ou com baixa infraestrutura médica. Este objetivo contribui para a democratização do acesso a diagnósticos de qualidade, oferecendo suporte em locais com escassez de recursos.

4.3 Motivação

O uso de inteligência artificial na área da saúde tem se mostrado um recurso promissor para otimizar processos e ampliar a acessibilidade de diagnósticos. O OpenECG surge como uma resposta às necessidades de pacientes e profissionais de saúde por uma plataforma acessível, rápida e precisa. Ao automatizar a análise de exames de ECG, o projeto pretende aprimorar a precisão dos diagnósticos cardiológicos, reduzir o custo do serviço de interpretação, tornando-o mais acessível, e diminuir o tempo de espera dos pacientes para receberem uma avaliação inicial.

Além disso, a inteligência artificial oferece aprendizado contínuo, permitindo a atualização constante do sistema de acordo com novas diretrizes e avanços médicos e tecnológicos. Esse aspecto torna o OpenECG uma plataforma dinâmica, capaz de evoluir

com as mudanças nas práticas de diagnóstico e tratamento recomendadas pela Sociedade Brasileira de Cardiologia, outras entidades internacionais e pela evolução da IA. Assim, o projeto propõe uma abordagem prática e atualizada, alinhada às necessidades modernas da medicina preventiva e de urgência.

5. TRABALHOS RELACIONADOS

5.1 Digital Imaging and Communications in Medicine (DICOM)

O DICOM é um padrão internacional amplamente utilizado para comunicação e gerenciamento de imagens médicas. Ele possibilita a interoperabilidade entre dispositivos de diferentes fabricantes, garantindo que imagens e informações associadas sejam armazenadas, transferidas e acessadas de maneira padronizada e eficiente.

Esse padrão especifica protocolos de comunicação em rede, formatos de arquivo para armazenamento e estruturas de diretórios médicos. Sua estrutura técnica é baseada em um modelo que organiza os dados em níveis hierárquicos, como Paciente, Estudo, Série e Imagem. Além disso, o DICOM utiliza o protocolo TCP/IP para transferência de dados em rede, assegurando confiabilidade e consistência.

Entre seus componentes principais, destacam-se as definições de objetos de informação, que representam entidades como imagens e relatórios, e as especificações de classes de serviço, que relacionam esses objetos a comandos, como armazenamento e consulta. O padrão também define requisitos para troca de mensagens, segurança dos dados e armazenamento em mídias físicas, como CDs e dispositivos USB.

O DICOM é amplamente utilizado em diversas especialidades médicas, incluindo radiologia, cardiologia e oncologia, além de terapias baseadas em imagens e diagnósticos remotos. Sua aplicação permite a integração entre equipamentos e sistemas, otimizando fluxos de trabalho e garantindo a acessibilidade e a proteção das informações.

Dessa forma, o DICOM estabelece-se como uma tecnologia essencial para o gerenciamento de dados médicos, promovendo eficiência, interoperabilidade e segurança em ambientes clínicos [DIC24].

5.2 Picture Archiving and Communication Systems (PACS)

O PACS é um sistema informatizado desenvolvido para a gestão de imagens médicas digitais. Ele substituiu gradualmente os métodos tradicionais baseados em filmes analógicos, centralizando o armazenamento de imagens e facilitando sua distribuição entre diferentes profissionais e sistemas hospitalares.

O PACS integra hardware e software para aquisição, armazenamento, exibição e manipulação de imagens médicas. Ele utiliza o protocolo DICOM como base para o armazenamento e a troca de dados, permitindo a interoperabilidade com sistemas como o Radiology Information System (RIS) e o Hospital Information System (HIS). Essa integração

viabiliza uma gestão mais eficiente dos fluxos de trabalho e a consulta simultânea de imagens por profissionais de diferentes localidades.

Entre suas principais funcionalidades estão a aquisição digital de exames, como radiografias, tomografias e ressonâncias magnéticas, além da manipulação avançada das imagens, como ajustes de contraste, brilho e zoom. O PACS também permite o armazenamento em servidores dedicados, com redundância e sistemas de backup, garantindo segurança e acessibilidade a longo prazo.

O sistema é amplamente utilizado em instituições de saúde devido aos seus benefícios, como a redução de custos com armazenamento físico e impressão, otimização do tempo de entrega de laudos e suporte à colaboração entre médicos por meio de acessos remotos. No entanto, sua implantação exige um investimento inicial significativo, adequação da infraestrutura tecnológica e treinamento dos profissionais para garantir o uso eficaz do sistema.

Assim, o PACS consolida-se como uma ferramenta indispensável para a modernização dos serviços de diagnóstico por imagem, promovendo eficiência, acessibilidade e segurança na gestão de dados médicos [Mac12].

5.3 Power Medical Cardio (PM Cardio)

O PMcardio é uma ferramenta inovadora alimentada por inteligência artificial, projetada para auxiliar no diagnóstico de doenças cardiovasculares diretamente de um smartphone. Este dispositivo médico permite interpretar eletrocardiogramas (ECGs) de 12 derivações com rapidez e precisão, fornecendo diagnósticos em segundos para 39 condições cardíacas, incluindo fibrilação atrial, bloqueio de ramo direito e infarto do miocárdio com oclusão (OMI).

A funcionalidade do PMcardio destaca-se pela praticidade e acessibilidade. O aplicativo aceita ECGs de qualquer idade ou qualidade, permitindo até mesmo capturar imagens diretamente da tela de um computador. O processo de utilização é simples: após selecionar o ECG, o usuário fotografa o exame utilizando a tecnologia avançada de reconhecimento de imagem do aplicativo. Em seguida, o PMcardio digitaliza e interpreta o ECG, fornecendo não apenas o diagnóstico, mas também um plano de tratamento personalizado.

Entre seus principais benefícios, o PMcardio oferece uma interpretação avançada de ECG, capaz de identificar indicadores sutis e melhorar a precisão diagnóstica, mesmo em casos clínicos complexos. A ferramenta funciona como uma fonte de segunda opinião imediata, auxiliando na tomada de decisões clínicas informadas sem a necessidade de consultar outros especialistas. Além disso, o PMcardio é compatível com todas as máquinas

de ECG de 12 derivações e está disponível como um aplicativo para smartphones, garantindo ampla acessibilidade.

Outro aspecto relevante é sua capacidade de revitalizar ECGs comprometidos, transformando exames danificados ou de baixa qualidade em relatórios digitais legíveis, que podem ser facilmente compartilhados ou arquivados. Esse recurso facilita a gestão de dados médicos e aprimora a continuidade do cuidado ao paciente.

Com a integração de inteligência artificial e facilidade de uso, o PMcardio representa uma evolução significativa na prática clínica, democratizando o acesso a diagnósticos cardíacos avançados e contribuindo para decisões rápidas e precisas em diferentes cenários de saúde [PMc24].

5.4 Vision

O Vision, da OpenAI, é uma funcionalidade avançada de inteligência artificial projetada para processamento de imagens e vídeos. Essa tecnologia permite que modelos compreendam e gerem conteúdo visual, expandindo suas aplicações em áreas como saúde, segurança, educação e entretenimento.

Entre suas principais funcionalidades estão a análise de imagens, permitindo a identificação de objetos, cenas e contextos em conteúdos visuais; a geração de imagens personalizadas com base em descrições textuais; e o processamento de vídeos, que inclui a interpretação de cenas e atividades, bem como a geração de vídeos a partir de textos.

As aplicações práticas do Vision incluem assistência a pessoas com deficiência visual, criação de conteúdo para marketing e educação e análises de segurança, como a identificação de atividades suspeitas por meio de câmeras. Além disso, sua capacidade de integrar tecnologias visuais com instruções éticas e guias técnicos detalhados facilita sua adoção em diversas indústrias.

Apesar de suas capacidades robustas, esta ferramenta não foi utilizada no presente trabalho devido a restrições descritas na documentação oficial da OpenAI. Segundo as diretrizes, o Vision não é recomendado para interpretações médicas, como a análise de imagens diagnósticas ou gráficos clínicos [Ope24].

6. FERRAMENTAS E TECNOLOGIAS UTILIZADAS

Para o desenvolvimento essas foram as tecnologias e ferramentas que possibilitaram a construção da plataforma OpenECG. A seguir, são descritas as principais ferramentas utilizadas, destacando suas funções específicas e contribuições para o projeto.

6.1 Flask

O Flask é um microframework em Python utilizado para o desenvolvimento de aplicações web e APIs. No projeto OpenECG, o Flask é a base para os micros serviços que compõem a plataforma, incluindo o auth-service (serviço de autenticação), o image-upload-service (serviço de upload de imagens) e o gpt-service (serviço de interpretação de exames). O Flask foi escolhido por sua simplicidade, flexibilidade e facilidade de integração com outras bibliotecas e frameworks, permitindo que os micro serviços fossem desenvolvidos de forma modular e escalável. Além disso, a estrutura leve do Flask facilita a implementação de endpoints RESTful, essenciais para o fluxo de dados entre o frontend e o backend da plataforma [Fla24].

6.2 Docker

Docker é uma tecnologia de containerização que permite empacotar e executar aplicações de forma isolada e consistente em qualquer ambiente. Na plataforma OpenECG, o Docker é utilizado para facilitar a implantação e a gestão da arquitetura de microservices, permitindo que cada componente da aplicação (como os serviços de autenticação, upload de imagem e análise GPT) seja executado em contêineres independentes. Com o Docker, a plataforma pode ser facilmente escalada, uma vez que novos contêineres (funcionalidades) podem ser adicionados conforme necessidade. A uniformidade proporcionada pelo Docker também simplifica o processo de desenvolvimento e teste, garantindo que a aplicação funcione de maneira consistente em diferentes ambientes [Doc24].

6.3 MySQL

O MySQL é um sistema de gerenciamento de banco de dados relacional amplamente utilizado para armazenamento persistente de dados [MyS24]. No OpenECG, o MySQL é empregado para armazenar as imagens dos exames de ECG, bem como os

resultados das interpretações realizadas pela inteligência artificial. A escolha pelo MySQL se deu pela sua facilidade de uso, além de oferecer um bom desempenho por consulta. O uso de um banco de dados relacional permite que as informações dos exames e dos usuários sejam organizadas de maneira estruturada, facilitando o acesso e a recuperação de dados conforme necessário.

6.4 GPT Assistant

O modelo de linguagem da OpenAI, especificamente um assistente personalizado do GPT, é a principal ferramenta de inteligência artificial utilizada no projeto. Esse modelo foi configurado para interpretar as imagens de ECG de acordo com as diretrizes da Sociedade Brasileira de Cardiologia, permitindo que as respostas sejam geradas em linguagem natural. O GPT Assistant é capaz de analisar os padrões encontrados nos exames e fornecer uma interpretação inicial, que pode ser compreendida facilmente por profissionais de saúde.

6.5 NGINX

O Nginx é um servidor web de alto desempenho que atua como proxy reverso e balanceador de carga para a plataforma [Ngy24]. Na arquitetura do OpenECG, o Nginx é utilizado para direcionar o tráfego entre o frontend e os microserviços backend, melhorando a performance da aplicação e garantindo uma experiência de usuário fluida. Como proxy reverso, o Nginx gerencia as requisições de forma eficiente, além de fornecer uma camada adicional de segurança, protegendo o backend da plataforma contra acessos não autorizados e distribuindo as requisições de maneira uniforme.

6.6 Postman

No OpenECG, o Postman foi essencial durante a fase de desenvolvimento para testar os endpoints da API, verificar a funcionalidade dos microserviços e garantir que o fluxo de dados estivesse funcionando corretamente. Através do Postman, foram realizados testes detalhados de cada endpoint, incluindo o upload de imagens, a autenticação de usuários e a comunicação com o GPT Assistant, assegurando que cada componente da plataforma interagisse de forma correta e eficiente.

6.7 Visual Studio Code (VSC)

O VSC é um editor de código fonte popular e repleto de funcionalidades para o desenvolvimento de aplicações. Durante o desenvolvimento do OpenECG, o VSC foi utilizado como ambiente principal de codificação, permitindo organização do código e integração com ferramentas de controle de versão, como o Git. Através de extensões e funcionalidades integradas, o VS Code oferece suporte a linguagens e frameworks, como Python e Flask, utilizados no projeto [Cod24].

6.8 Vue

O Vue.js é um framework JavaScript de código aberto utilizado para a construção de interfaces de usuário modernas, oferecendo uma abordagem declarativa e baseada em componentes que simplifica o desenvolvimento. Sua reatividade permite acompanhar mudanças no estado do JavaScript e atualizar o DOM de forma eficiente, enquanto os Componentes de Arquivo Único (SFC) organizam lógica, template e estilos em um único arquivo, promovendo a reutilização de código. Além disso, sua flexibilidade permite adoção incremental, desde melhorias em HTML estático até a criação de Aplicações de Página Única (SPA) [Vue24].

No OpenECG, o Vue.js foi empregado para implementar uma interface de usuário dinâmica e responsiva, garantindo uma experiência intuitiva e facilitando a interação com os serviços da plataforma.

6.9 Python

Python é uma linguagem de programação de alto nível, orientada a objetos, amplamente reconhecida por sua simplicidade e versatilidade. Sua sintaxe clara facilita o desenvolvimento de soluções em diversas áreas, como desenvolvimento web, automação de tarefas, análise de dados e inteligência artificial. Além disso, Python possui uma biblioteca padrão abrangente, que inclui módulos para manipulação de arquivos, conexão a servidores e processamento de dados, o que reduz a necessidade de dependências externas. Gratuito e de código aberto, Python conta com uma comunidade ativa que contribui para sua constante evolução [Pyt24].

No projeto OpenECG, Python foi escolhido como linguagem principal por sua eficiência na implementação de APIs com Flask, integração com a API do GPT Assistant e

suporte a ferramentas para análise e manipulação de dados, tornando-se uma peça central para o desenvolvimento de microserviços escaláveis e robustos.

Essas ferramentas e tecnologias foram escolhidas e integradas para atender às necessidades do OpenECG, contribuindo para a criação de uma plataforma acessível e escalável. Cada uma desempenha um papel essencial no funcionamento da aplicação, desde o processamento de dados até a experiência do usuário final.

7. ANÁLISE DO SISTEMA

A análise da arquitetura do sistema OpenECG detalha as camadas fundamentais que compõem a plataforma, descrevendo como cada uma delas contribui para uma operação modular e escalável. A seguir, são discutidos os aspectos técnicos e funcionais das camadas principais.

7.1 Frontend

A camada de interface, ou frontend, foi implementada utilizando o framework Vue.js devido à sua reatividade e modularidade, que trazem responsividade à aplicação. A interface do OpenECG permite ao usuário “Paciente” fazer login, realizar o upload de exames de ECG, visualizar os resultados das interpretações, visualizar a lista de exames enviados para interpretação, enviar um novo exame e consultar informações detalhadas sobre exames já enviados. Ao usuário “Médico”, a interface permite fazer login, visualizar a lista de exames enviados pelos pacientes para interpretação, visualizar o exame selecionado, interpretar o exame selecionado e avaliar a interpretação fornecida pela IA. As Figuras 7.1 e 7.2 ilustram as jornadas de cada tipo de usuário.

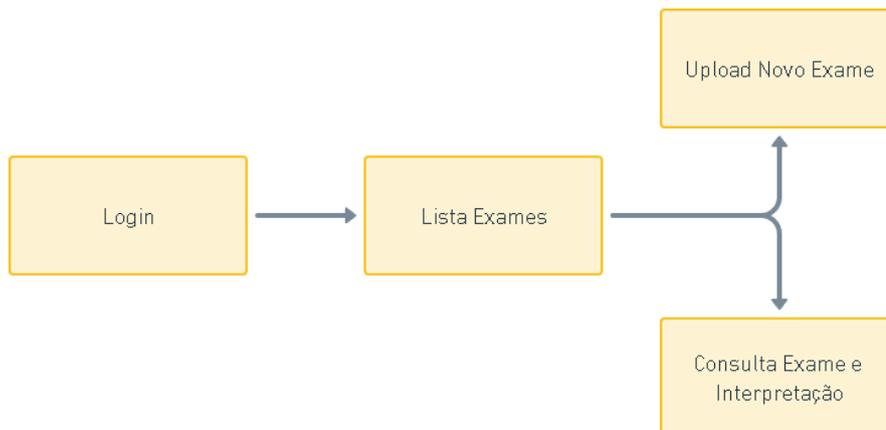


Figura 7.1 – Jornada do usuário Paciente.

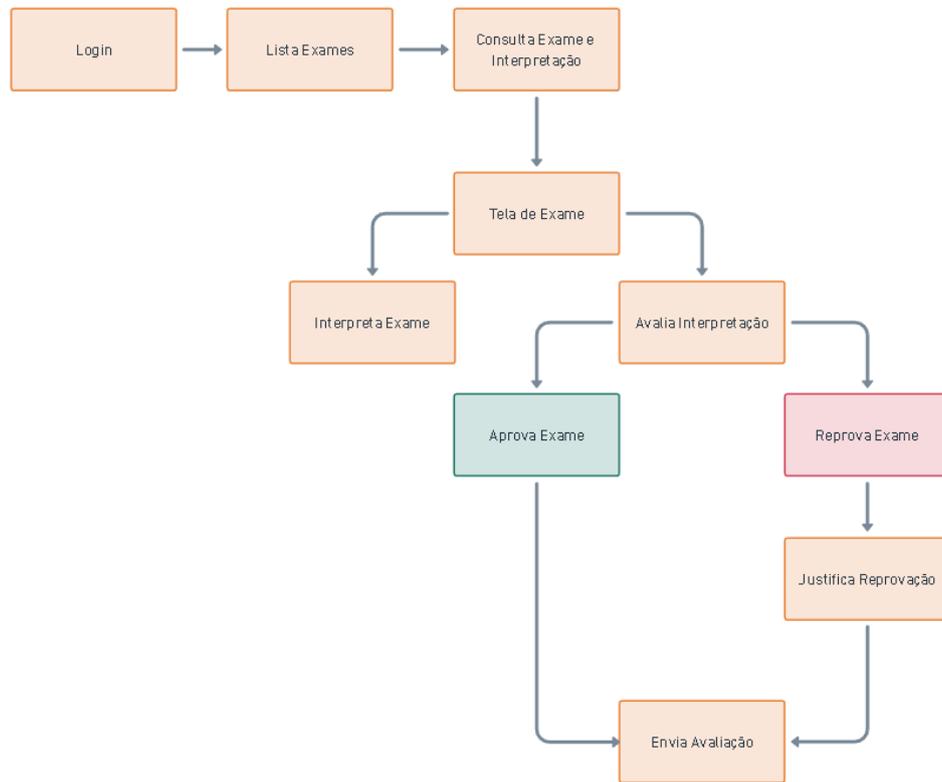


Figura 7.2 – Jornada do usuário Médico.

O frontend comunica-se com a camada de API Gateway enviando e recebendo dados dos micro serviços do backend. Com Vue.js, a arquitetura é composta por componentes reutilizáveis, permitindo que as atualizações no frontend sejam implementadas sem impactos significativos na estrutura global do sistema.

7.1.1 Login

A tela apresenta duas seções principais: "Login", onde o usuário pode inserir nome de usuário/e-mail e senha para autenticação por meio do botão azul "Login", e Registro de Novo Usuário, que permite cadastrar uma nova conta selecionando uma função no dropdown entre "médico" ou "paciente", além de preencher nome de usuário e senha, finalizando o processo com o botão verde "Registrar"(Figura 7.3).

The image shows a user interface with two main sections. The top section is titled "Login" and contains two input fields labeled "Usuário:" and "Senha:". Below these fields is a blue button labeled "Login". The bottom section is titled "Registrar Novo Usuário" and contains a dropdown menu labeled "Função:", followed by three input fields labeled "Usuário:", "Senha:", and "Senha:". Below these fields is a green button labeled "Registrar".

Figura 7.3 – Tela de Login.

7.1.2 Listagem de Exames do usuário Paciente

A tela apresenta uma listagem de exames enviados, com as seguintes funcionalidades: exibir o total de exames registrados em uma tabela contendo colunas para número identificador, nome do arquivo, data de envio, status e ações disponíveis. O botão verde "Enviar Novo Exame" permite o envio de novos arquivos, enquanto o botão azul "Ver Detalhes" na coluna "Ações" possibilita visualizar informações específicas de um exame. Há também um botão "Sair", no canto inferior direito, para encerrar a sessão do usuário (Figura 7.4).



Figura 7.4 – Tela de Listagem de Exames do usuário Paciente.

7.1.3 Tela de envio do Exame

A tela "Enviar Exame de ECG" permite o upload de imagens para análise. Possui um campo "Escolha o arquivo", que abre um seletor de arquivos para que o usuário selecione a imagem desejada. Essa tela também possui dois botões: o botão azul "Enviar" para realizar o upload do arquivo selecionado e o botão "Voltar", que redireciona o usuário para a tela anterior. A interface é simples e focada na funcionalidade de envio do exame (Figura 7.5).

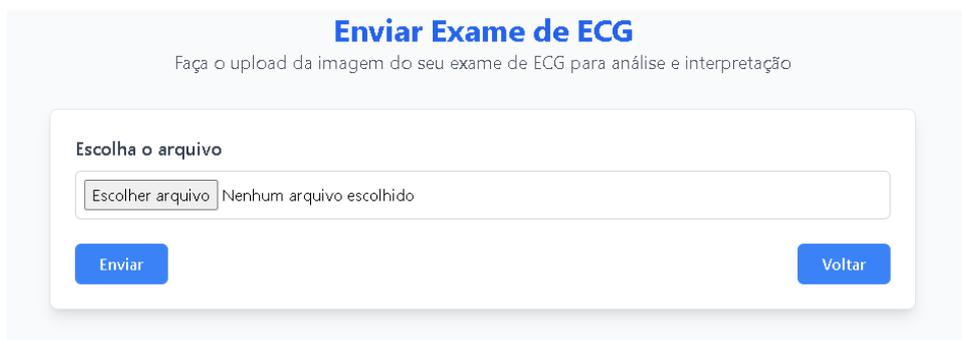


Figura 7.5 – Tela de Upload de exame.

7.1.4 Tela de visualização e interpretação do exame do usuário Paciente (/exam/examId)

A tela "Exame - Eletrocardiograma (ECG)" apresenta detalhes do exame, incluindo a data de upload, o status da análise (neste caso, "Interpretação Pendente") e a visualização da imagem do ECG enviado. Possui também um botão azul "Voltar", que permite retornar à tela anterior. Essa é a interface focada na exibição das informações do exame (Figura 7.6).

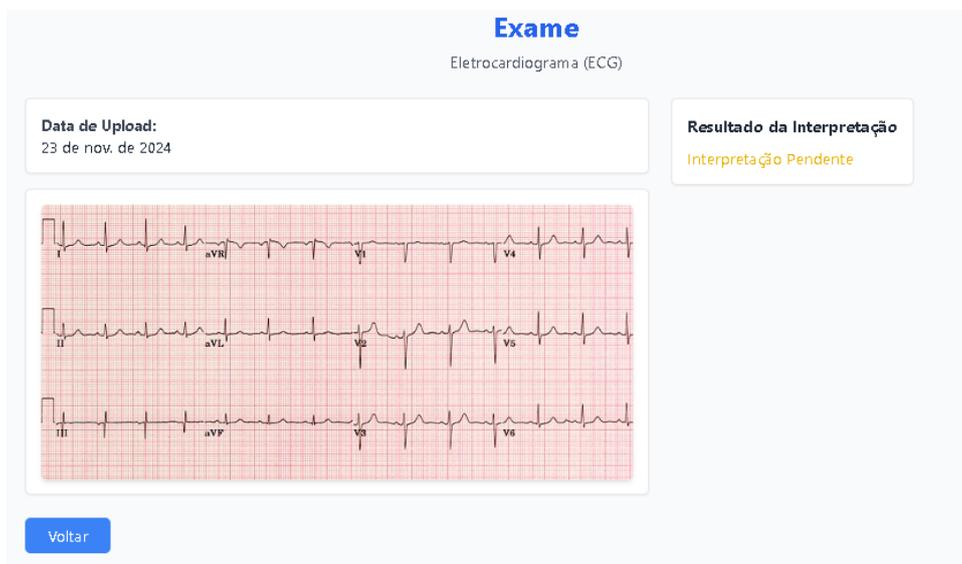


Figura 7.6 – Tela de Exame do Paciente.

7.1.5 Listagem de Exame do Médico.

A diferença entre a tela do Paciente é que essa apresenta uma listagem de exames enviados por todos os pacientes e a mesma tabela contendo as colunas para número do identificador, nome do arquivo, data de envio, status e ações disponíveis mas não possui o botão "Enviar Novo Exame" para envio de novos arquivos (Figura 7.7).

#	NOME DO ARQUIVO	DATA DE ENVIO	STATUS	AÇÕES
1	ECG_2.jpg	23 de nov. de 2024	Pendente	Ver Detalhes
2	ECG_3.jpg	23 de nov. de 2024	Pendente	Ver Detalhes
3	ECG1.jpg	23 de nov. de 2024	Pendente	Ver Detalhes
4	ECG1.jpg	23 de nov. de 2024	Pendente	Ver Detalhes

Figura 7.7 – Tela de Listagem de Exames do usuário Médico.

7.1.6 Tela de visualização e interpretação do exame do usuário Médico

Diferente da tela de Exames do Paciente, a tela atual possui funcionalidades adicionais: um botão "Solicitar Interpretação" para enviar uma requisição para o Assistente interpretar o ECG e uma seção de Avaliação, onde o especialista deve definir um status

para a interpretação (via dropdown) e enviá-lo usando o botão "Enviar Avaliação". Em caso de reprovação da interpretação fornecida pelo GPT, o Médico precisará fornecer uma justificativa ao status escolhido (Figura 7.8).

The screenshot displays a web interface for a medical exam. At the top, it says 'Exame' and 'Eletrocardiograma (ECG)'. Below this, there's a box for 'Data de Upload:' showing '23 de nov. de 2024'. The main area contains a 12-lead ECG waveform. To the right, there's a 'Resultado da Interpretação' section with 'Interpretação Pendente' and a 'Solicitar Interpretação' button. Below that is an 'Avaliação' section with a dropdown menu for 'Status do Exame' and an 'Enviar Avaliação' button. A 'Voltar' button is located at the bottom left of the interface.

Figura 7.8 – Tela de Exame do usuário Médico.

7.2 API Gateway (NGINX)

O Nginx opera como o “porteiro” do projeto, direcionando as requisições entre os “moradores”, nessa analogia aos microsserviços. O Nginx distribui as requisições do usuário de maneira uniforme entre os recursos do sistema, otimizando o tempo de resposta e evitando sobrecarga em determinados serviços. Funcionando como proxy reverso para gerenciar o tráfego de rede entre o frontend e os microsserviços do backend, desempenha um papel importante no balanceamento de carga da plataforma. Essa camada gerencia as conexões de maneira eficiente para que a experiência de usuário se mantenha responsiva.

7.3 Backend

A camada de backend do projeto OpenECG é composta por uma série de microsserviços independentes, cada um desempenhando uma função específica para garantir a modularidade, escalabilidade e facilidade de manutenção do sistema. Esses microsserviços são executados e isolados em containers Docker, permitindo que cada serviço seja atualizado ou substituído de forma independente. Abaixo estão os principais componentes da camada de backend:

- **auth-service:** Responsável pela autenticação e autorização dos usuários, o auth-service utiliza “roles” para gerenciar o acesso às rotas da aplicação. Este serviço garante que o usuário “paciente” visualize apenas os seus exames e suas interpretações e que o usuário “médico” visualize todos os exames e possa interpretar cada um deles, indiferente do usuário.
- **image-upload-service:** Este serviço gerencia o upload das imagens dos exames de ECG, além de seu armazenamento e organização. Ao receber uma imagem, o image-upload-service a armazena no banco de dados, gerando um identificador único (imagemId) para cada arquivo. Esse identificador facilita a recuperação da imagem quando necessário e permite que o gpt-service acesse as imagens para realizar a análise. Além disso, o image-upload-service pode ser configurado para utilizar armazenamento externo, caso seja necessário (Figura 7.9).

```
@app.route('/getExames/<int:user_id>', methods=['GET'])
def get_Imagens(user_id):
    role = request.args.get('role')

    imgs = Image.query

    if role == 'paciente':
        imgs = imgs.filter_by(user_id=user_id)

    imgs = imgs.all()

    images = [img.as_dict() for img in imgs]

    return jsonify({"exams": images}), 200
```

Figura 7.9 – Trecho de código do micro serviço que lista os exames.

- **gpt-service:** O gpt-service é o microsserviço central para a interpretação dos exames de ECG, realizando a integração com a API do GPT Assistant para processar as imagens. Este serviço recebe o identificador da imagem (imagemId) e acessa o banco de dados para recuperá-la, enviando a imagem ao GPT Assistant para análise. O processo de interpretação envolve a análise automatizada pelo GPT Assistant, que gera uma resposta interpretativa e a armazena no banco de dados para posterior consulta pelo usuário. Esse design modular permite que o serviço de interpretação seja facilmente atualizado ou substituído por outras ferramentas de IA, caso necessário, sem impactar o funcionamento dos demais componentes do sistema (Figura 7.10).

```

@app.route('/interpret/<image_id>', methods=['POST'])
def interpret_ecg(image_id):

    # Obtém o registro da imagem enviada
    image_record = Image.query.filter_by(id=image_id).first()

    if not image_record:
        return jsonify({'message': 'Imagem não encontrada'}), 404

    # Faz upload do arquivo para a OpenAI
    file = client.files.create(
        file=open(
            path.join(getcwd(), image_record.as_dict()['image_path']), "rb"),
        purpose="assistants"
    )

    # Cria uma nova thread e envia a mensagem inicial com a imagem anexada
    thread = client.beta.threads.create(
        messages=[
            {
                "role": "user",
                "content": [
                    {
                        "type": "text",
                        "text": "Você poderia interpretar esse ECG para mim?"
                    },
                    {
                        "type": "image_file",
                        "image_file": {"file_id": file.id}
                    }
                ]
            }
        ]
    )

```

Figura 7.10 – Trecho de código do micro serviço que solicita interpretação ao GPT.

7.4 Banco de Dados

A camada de banco de dados utiliza o MySQL para o armazenamento e gerenciamento dos dados da plataforma. Esse banco de dados relacional é responsável pela persistência das imagens dos exames de ECG, das interpretações geradas pelo GPT Assistant e dos dados de usuários. A estrutura relacional do MySQL garante consistência transacional e integridade referencial, o que é essencial para a confiabilidade dos dados em um sistema médico. Os microsserviços "image-upload-service" e "gpt-service" interagem diretamente com o banco de dados para armazenar e recuperar imagens e interpretações, permitindo a disponibilidade contínua de informações para futuras consultas e análises (Figura 7.11).

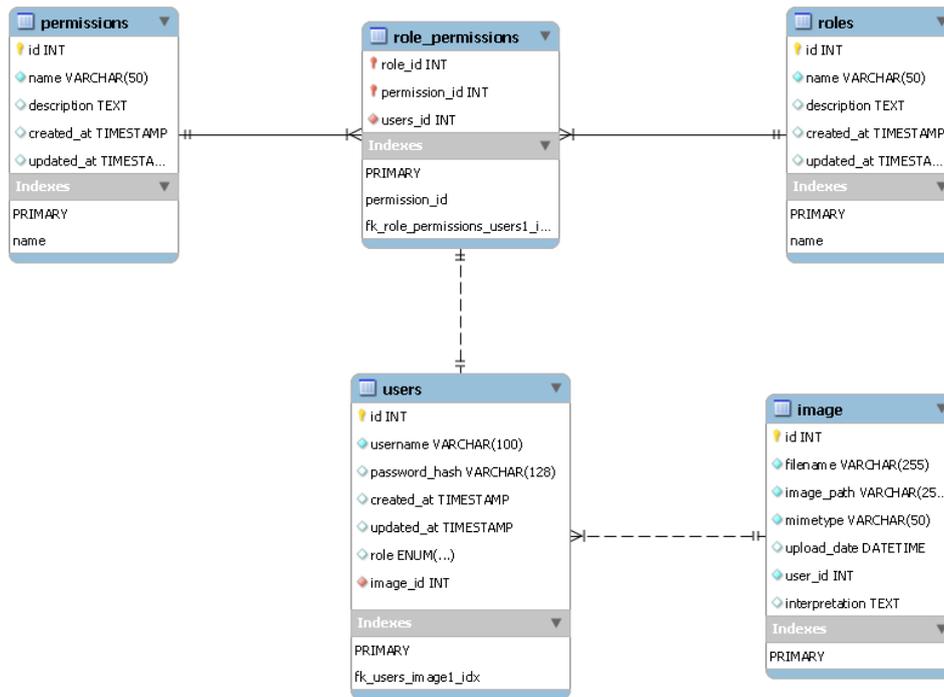


Figura 7.11 – Diagrama Entidade e Relacionamento do banco de dados do projeto

O modelo de banco de dados apresentado possui cinco tabelas principais interconectadas: users, roles, permissions, role_permissions, e image. A tabela "users" armazena informações dos usuários, como username, password_hash e referências à tabela roles (campo role) e à tabela image (campo image_id). A tabela roles define os papéis dos usuários, enquanto a tabela permissions detalha permissões específicas. A tabela intermediária role_permissions estabelece a relação entre papéis e permissões, vinculando as tabelas roles, permissions e users. Por fim, a tabela image armazena dados relacionados a imagens de ECG, como filename, image_path, e o resultado da interpretação (interpretation), associando essas informações a um usuário específico via user_id. As relações são estruturadas com chaves primárias e estrangeiras, garantindo consistência e integridade entre as tabelas.

7.5 Fluxo dos Dados

O fluxo de dados no OpenECG foi projetado para garantir uma interação segura, eficiente e contínua entre os usuários e o sistema, desde o upload do exame até a visualização das interpretações geradas. Abaixo, estão as etapas detalhadas do fluxo, divididas em três principais fases: Upload e Armazenamento de Imagem, Processamento e Interpretação da Imagem e Consulta e Visualização.

7.5.1 Upload e Armazenamento de Imagem

1. Um usuário autenticado acessa a interface de usuário (frontend) e faz o upload de uma imagem de ECG.
2. A imagem é então enviada ao image-upload-service, que valida e processa o arquivo recebido (Figura 7.12).

```
@app.route('/upload', methods=['POST'])
def upload_file():
    # Validação
    user_id = request.form.get('user_id')
    if not user_id:
        return jsonify({'error': 'User ID is required'}), 400

    if 'file' not in request.files:
        return jsonify({'error': 'No file part'}), 400

    file = request.files['file']
    if file.filename == '':
        return jsonify({'error': 'No selected file'}), 400

    if file and allowed_file(file.filename):
        # Processamento
        try:
            # Gerar um nome único para o arquivo
            unique_filename = f"{uuid.uuid4().hex}_{secure_filename(file.filename)}"
            filepath = os.path.join(app.config['UPLOAD_FOLDER'], unique_filename)

            # Salvar o arquivo no disco
            file.save(filepath)

            # Salvar o caminho no banco de dados
            img = Image(
                filename=secure_filename(file.filename),
                mimetype=file.mimetype,
                user_id=user_id,
                image_path=filepath # Salva o caminho no banco
            )
            db.session.add(img)
            db.session.commit()

            return jsonify({'message': 'Arquivo enviado com sucesso', 'file_path': filepath}), 201
        except Exception as e:
            return jsonify({'error': str(e)}), 500
    else:
        return jsonify({'error': 'Tipo de arquivo não permitido'}), 400
```

Figura 7.12 – Trecho do código que valida e processa o arquivo recebido.

3. O image-upload-service armazena a imagem no banco de dados MySQL e gera um identificador único, o `imageId`, associado a essa imagem. Esse identificador é retornado ao frontend, que o utiliza para futuras operações de consulta e interpretação.

7.5.2 Processamento e Interpretação da Imagem

1. Passando como parâmetro o `image_id`, o frontend solicita a interpretação da imagem ao `gpt-service`.
2. O `gpt-service` recupera a imagem correspondente do banco de dados usando o `image_id` e, em seguida, envia essa imagem ao GPT Assistant para que a interpretação seja realizada (Figura 7.13).

```
@app.route('/interpret/<image_id>', methods=['POST'])
def interpret_ecg(image_id):
    print("Início", ASSISTANT_ID)

    # Obtém o registro da imagem enviada
    image_record = Image.query.filter_by(id=image_id).first()

    if not image_record:
        return jsonify({'message': 'Imagem não encontrada'}), 404

    # Faz upload do arquivo para a OpenAI
    file = client.files.create(
        file=open(
            path.join(getcwd(), image_record.as_dict()['image_path']), "rb"),
        purpose="assistants"
    )
```

Figura 7.13 – Trecho do código que recupera a imagem e envia para o Assistente

3. O GPT Assistant analisa a imagem de ECG e gera uma interpretação em formato textual, contendo informações detalhadas sobre o exame e possíveis diagnósticos.
4. A interpretação gerada é então armazenada no banco de dados para que possa ser acessada posteriormente, tanto para consultas pelo usuário quanto para uso em futuras análises ou comparações.

7.5.3 Consulta e Visualização

1. Usuários autenticados podem acessar o frontend a qualquer momento para visualizar as interpretações dos exames de ECG que fizeram upload.
2. O frontend realiza uma consulta ao `gpt-service` e ao banco de dados, recuperando a interpretação associada ao `image_id` do usuário.
3. A interpretação é então exibida diretamente no frontend, permitindo que o usuário visualize o resultado da análise de forma clara e acessível (Figura 7.14).

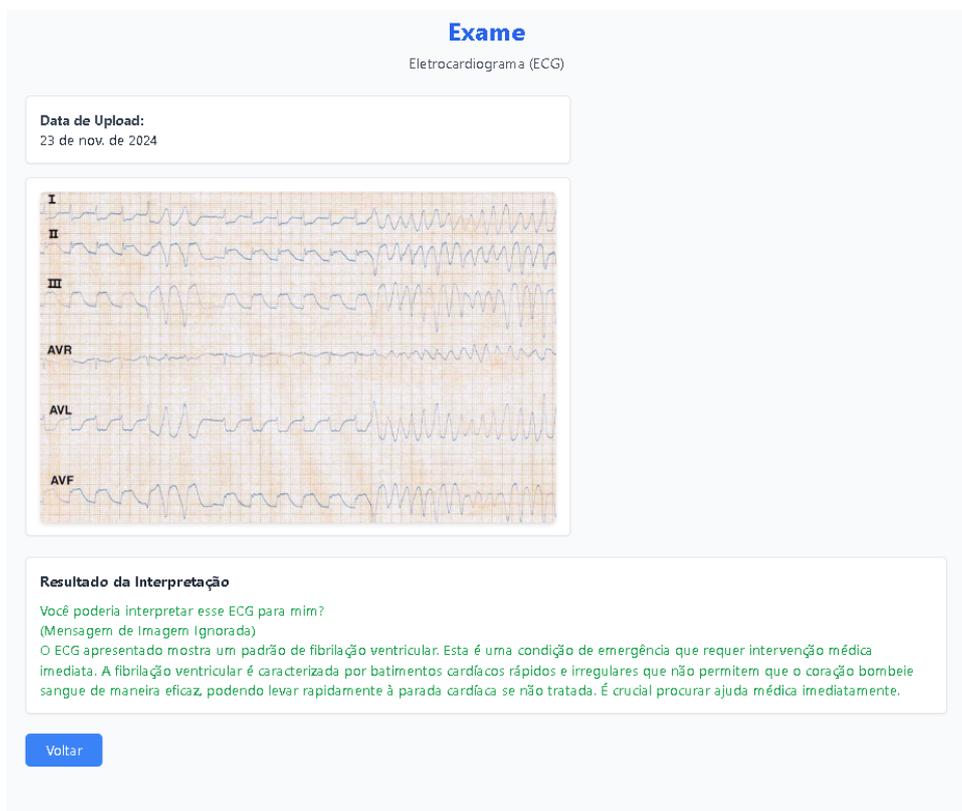


Figura 7.14 – Tela de Exame, onde o usuário visualiza o ECG e a interpretação

Esse fluxo de dados assegura que o OpenECG ofereça uma experiência contínua para os usuários, mantendo o histórico de exames e interpretações acessível e organizado. Além disso, a modularidade dos micro serviços permite que cada etapa do fluxo seja otimizada e atualizada de maneira independente, promovendo a escalabilidade e a manutenção eficiente do sistema.

7.6 Assistente GPT

O principal componente do projeto é o Assistente GPT, responsável por interpretar o exame de ECG a partir de uma solicitação em texto e da imagem enviada pelo usuário. No exemplo apresentado, o prompt inicial "Você poderia interpretar esse ECG para mim?" é utilizado como entrada textual, enquanto o exame é anexado como um arquivo de imagem identificado pelo file ID (Figura 7.15)

A imagem representa a integração entre o sistema e a API do Assistente GPT, destacando a parte do código responsável por configurar a interação. O sistema cria uma nova thread que armazena o histórico da comunicação e inclui tanto o conteúdo textual quanto a imagem recebida. Essa estrutura inicial é essencial para fornecer o contexto necessário para a interpretação do ECG.

```

# Cria uma nova thread e envia a mensagem inicial com a imagem anexada
thread = client.beta.threads.create(
  messages=[
    {
      "role": "user",
      "content": [
        {
          "type": "text",
          "text": "Você poderia interpretar esse ECG para mim?"
        },
        {
          "type": "image_file",
          "image_file": {"file_id": file.id}
        }
      ]
    }
  ]
)

# Armazena o id da thread criada
thread_id = thread.id

# Executa o Assistant no thread
run = client.beta.threads.runs.create(
  thread_id=thread_id,
  assistant_id=ASSISTANT_ID,
  instructions="Retorne APENAS o diagnóstico e os achados do ECG, use o mínimo de palavras possíveis para os achados e seja muito assertivo",
  tools=[{"type": "file_search"}]
)

```

Figura 7.15 – Trecho do código com o prompt enviado ao Assistente

Além disso, o código apresenta a configuração do prompt inicial e das instruções adicionais que personalizam o comportamento do Assistente GPT. Nesse caso, as instruções "Retorne APENAS o diagnóstico e os achados do ECG, use o mínimo de palavras possíveis para os achados e seja muito assertivo" são utilizadas para restringir o formato da resposta. Esse comando sobrescreve as instruções padrão do modelo, garantindo que a saída gerada seja mais direta e focada nos achados clínicos mais relevantes.

Por fim, o código também define o uso da ferramenta "file_search", sugerindo que o assistente consulte a base de conhecimento para fortalecer a precisão da interpretação. Esse tipo de configuração demonstra a flexibilidade da API em adaptar a resposta gerada de acordo com as necessidades do projeto, fornecendo diagnósticos rápidos e assertivos com base nas instruções criadas.

As Diretrizes Sobre Análise e Emissão de Laudos Eletrocardiográficos servirão como base de conhecimento para esse modelo e ficam armazenadas em forma de vetor no próprio serviço da OpenAI, as instruções por sua vez definem o comportamento esperado do modelo, junto com exemplos de entradas possíveis e o padrão de saída de texto esperado. O assistente então recebe a requisição do frontend para interpretar a imagem com aquele ID e o prompt padrão para interpretação, interpreta a imagem identificando os padrões do traçado eletrocardiográfico, consulta as diretrizes para basear sua decisão e retorna um diagnóstico inspirado nos exemplos passados nas instruções.

7.6.1 Lista de Instruções

- **Instruções gerais:** "Você é um interpretador de eletrocardiogramas treinado para analisar e interpretar traçados de ECG com base nas diretrizes da Sociedade

Brasileira de Cardiologia para Análise e Emissão de Laudos Eletrocardiográficos (disponível em um PDF fornecido). Sua tarefa é emitir um ****diagnóstico breve e objetivo**** para cada exame."

- **Tom e Estilo:**

- Tom: Clínico, objetivo e autoritário.
- Estilo: Direto ao ponto, focado apenas nas informações essenciais.

- **Formato da Resposta:**

- Diagnóstico: Uma frase descrevendo o traçado, focando no diagnóstico clínico principal.
- Confiança: Indicar o nível de confiança da análise com base na aderência às diretrizes.
- Qualidade da Imagem: Caso a qualidade do exame impeça uma interpretação precisa, emitir uma mensagem recomendando a repetição do exame.

- **Exemplos de Respostas:**

- "Traçado normal. Confiança: Alta."
- "Bradicardia sinusal. Confiança: Alta."
- "Bloqueio de ramo esquerdo. Confiança: Moderada."
- "Traçado de baixa qualidade. Recomenda-se repetir o exame."

- **Casos Especiais:**

- Resultados Ambíguos: Sinalizar resultados que exigem confirmação com outras ferramentas diagnósticas ou revisão por especialistas. Exemplo: "Alterações inespecíficas. Imagem de Baixa Qualidade: Se a qualidade da imagem não permite a interpretação, responder: "Qualidade insuficiente para análise. Recomenda-se repetir o exame."

- **Testes e Avaliação:**

- Testes de Diagnóstico: Avalie a performance do modelo com um conjunto de ECGs previamente analisados por especialistas.
- Revisão por Especialistas: Compare as respostas geradas pelo modelo com laudos emitidos por cardiologistas.
- Ajustes Pós-Implantação: Monitore os resultados iniciais para refinar o modelo e corrigir desvios, se necessário.

7.6.2 Assistant API

O código referenciado a seguir foi desenvolvido para integrar o assistente criado na plataforma da OpenAI ao micro serviço "gpt-service", e possui as seguintes definições:

1. Armazena a Id da "thread" criada e executa o assistente;
2. Aguarda a conclusão da "run" para coletar as mensagens da "thread";
3. Processa as mensagens e armazena a interpretação (Figura 7.16);

```
# Armazena o id da thread criada
thread_id = thread.id

# Executa o Assistant no thread
run = client.beta.threads.runs.create(
    thread_id=thread_id,
    assistant_id=ASSISTANT_ID,
    instructions="Retorne APENAS o diagnóstico",
    tools=[{"type": "file_search"}]
)

# Wait for run to complete
wait_for_run_completion(thread_id, run.id)

# Recupera as mensagens da thread após a conclusão
messages = client.beta.threads.messages.list(
    thread_id=thread_id
)

# Processa apenas as mensagens do assistente para armazenar a interpretação
conversation = "" # Variável para armazenar a conversa filtrada
for message in reversed(messages.data):
    # Filtrar apenas mensagens do assistente
    if message.content:
        for content_item in message.content:
            if content_item.type == "text":
                # Adiciona a mensagem do assistente ao log
                conversation += f"{content_item.text.value}\n"
            elif content_item.type == "image_file":
                conversation += "(Mensagem de Imagem Ignorada)\n"
            else:
                conversation += "(Outro tipo de conteúdo ignorado)\n"

# Atualiza a interpretação no banco de dados se houver uma resposta
if conversation.strip():
    print("Conversação completa obtida:\n", conversation)
    image_record.interpretation = conversation.strip() # Save the entire conversation
    db.session.commit()
    return jsonify({'conversation': conversation.strip()})
else:
    return jsonify({'message': 'Falha ao obter a conversa'}), 500
```

Figura 7.16 – Trecho do código que interage com a API

8. ANÁLISE HOMEM X MÁQUINA

Este capítulo explora a comparação entre o desempenho de um especialista humano altamente qualificado e um modelo de IA baseado no GPT, desenvolvido e ajustado especificamente para interpretar exames de ECG segundo as diretrizes da SBC. A análise é conduzida de forma a verificar a assertividade, tempo de interpretação e custos associados às interpretações de ECG, levando em consideração a expertise de um especialista mestre em cardiologia e doenças cardiovasculares, com mais de 30 anos de experiência, em comparação com o desempenho do GPT Assistant criado neste projeto.

8.1 Análise

Para garantir uma avaliação detalhada, a análise foi estruturada em duas etapas. Na primeira, foi realizado um quiz de ECG fornecido pela Oxford Medical School, que contém exames e laudos de ECG com as respostas corretas fornecidas pela instituição. Este quiz permitiu uma comparação direta entre a IA e o especialista, avaliando o nível de acertos de ambos frente a um gabarito oficial, o que oferece um referencial sólido para análise da precisão diagnóstica de cada um.

A segunda etapa foi baseada na comparação entre 133 diagnósticos de exames clínicos reais de uma clínica de emissão e interpretação de ECG realizados durante os meses de agosto, setembro e outubro de 2024, interpretados pelo próprio especialista, e os diagnósticos fornecidos pelo modelo GPT. É importante mencionar aqui que o formato de saída da resposta do GPT foi alterado para que se comparasse ao do especialista, diferente do que no formato da primeira etapa, aqui a resposta é mais sucinta, inspirando-se nos diagnósticos reais do cardiologista. Os resultados podem ser vistos a seguir.

8.2 Assertividade

Na primeira etapa da análise, os resultados do quiz indicaram que o especialista humano obteve 15 acertos em 22 questões (Figura 8.1), enquanto o GPT Assistant alcançou 16 acertos. A validação das respostas foi conduzida pelo especialista, que analisou e comparou os outputs da IA com o gabarito oficial para fornecer uma avaliação especializada da precisão dos acertos. Apesar de a IA ter obtido um número ligeiramente maior de respostas corretas, as questões acertadas por ela diferiram das acertadas pelo especialista, destacando diferenças nas capacidades de cada um. Por exemplo, a IA não identificou corretamente síndromes como Brugada e Wolff-Parkinson-White, que foram

diagnosticadas pelo especialista, enquanto acertou uma questão envolvendo alterações no intervalo QT, que passou despercebida pelo especialista.

Pergunta 1

Um homem de 35 anos apresenta palpitações. Ele tem bebido muito com amigos no fim de semana. Este é seu ECG. Apresente suas descobertas e dê um diagnóstico.

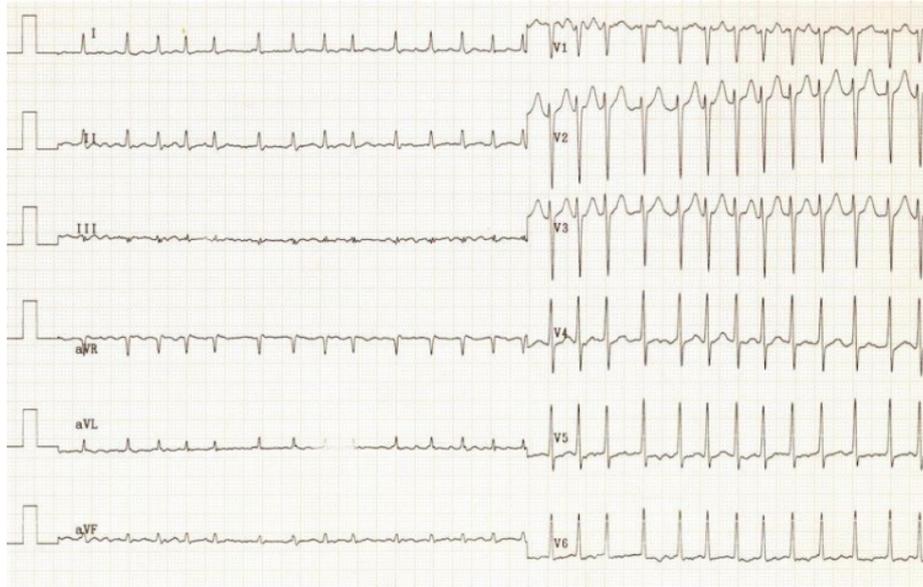


Figura 8.1 – Primeira questão do questionário.

Outro ponto relevante observado foi que a IA apresentou melhor desempenho quando o contexto clínico do paciente foi incluído como input adicional junto à imagem do ECG. Esse contexto auxiliou o GPT Assistant a interpretar o cenário clínico de maneira mais precisa, reforçando a importância de informações complementares no processo de análise (Figura 8.2).

■ Responder

Apresentação:

Avaliar	100 - 150
Ritmo	Irregularmente irregular
Eixo	Normal
Onda PR/P	Nenhuma onda p vista. Linha de base fibrilante
QRS	Estreito
Onda ST/T	Normal
QTc/outro	Normal

Diagnóstico:

Este ECG mostra fibrilação atrial (FA) com uma resposta ventricular rápida. Com essa história, o diagnóstico subjacente se encaixaria com uma síndrome de "coração de férias".

Figura 8.2 – Resposta para primeira questão do questionário.

Na segunda etapa, que envolveu a análise de 132 exames de ECG reais de pacientes, o desempenho da IA foi inferior. Das 132 interpretações, a IA apresentou correspondência exata com os laudos reais em 58 casos. Entre os 85 exames classificados como "traçados dentro dos padrões de normalidade", o GPT Assistant acertou 48, indicando maior precisão ao diferenciar traçados normais de anormais, mas com dificuldades em identificar a natureza específica de anomalias em exames alterados.

Confirmando a informação anterior, a análise revelou que a IA demonstrou uma maior taxa de acertos ao categorizar exames de forma binária – "normal" ou "com anomalia" – em vez de tentar identificar anomalias específicas. Havia 85 exames com diagnóstico igual a "Traçado dentro dos padrões de normalidade" e o OpenECG interpretou corretamente 48 deles, percentualmente, 56% dos resultados foram diagnosticados corretamente. Dos 47 exames com um ou mais tipos de alteração no traçado cardíaco 26 foram identificados com alguma alteração pela IA (Figura 8.3), 7 eram os exames com traçado indicativo de "Bradicardia Sinusal" e foram interpretados corretamente 4 vezes, confirmando o percentual de 55% de assertividade. Isso sugere que o GPT Assistant possui um potencial mais robusto para triagens iniciais, onde a prioridade é identificar exames anormais para posterior análise mais detalhada por um especialista humano. No entanto, para cenários clínicos complexos, a precisão da IA ainda apresenta limitações que devem ser consideradas.

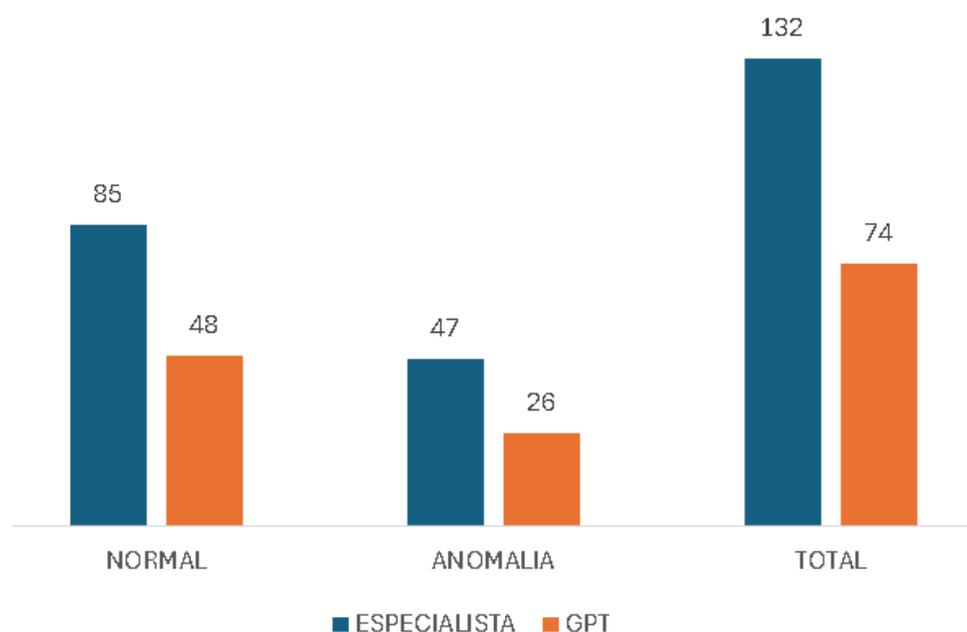


Figura 8.3 – Gráfico de assertividade entre o profissional e a IA.

A baixa precisão da ferramenta de interpretação de ECG pode ser explicada por uma combinação de fatores, esses relacionados à ausência de outras bases de conhecimento consolidadas, trazendo novas informações e validadas por diferentes especialistas. A falta dessa integração afeta diretamente a capacidade do modelo de reconhecer padrões complexos ou condições raras e para superar essa limitação, seria necessário criar uma base mais robusta que representasse um conhecimento acumulado mais extenso. O processamento das imagens também é um ponto chave. O modelo pode não estar transformando os pixels dos traçados de ECG em parâmetros claros e específicos, que são essenciais para uma análise eficiente. Essa deficiência dificulta a interpretação de características importantes, limitando a capacidade do modelo de diferenciar traçados de diferentes condições clínicas. Um aprimoramento no pré-processamento das imagens, convertendo os pixels em representações mais interpretáveis, ajudaria a ferramenta a identificar padrões com maior acurácia.

Outros fatores são a quantidade e a diversidade dos dados de treinamento. A baixa variabilidade nos diagnósticos apresentados nos dados de treinamento reduz a capacidade da ferramenta de generalizar padrões e reconhecer condições menos frequentes[LCW⁺14]. Para mitigar esse problema, é imprescindível incluir um maior número de exames no treinamento, com uma ampla gama de diagnósticos que abranjam desde casos comuns até manifestações raras, promovendo um aprendizado mais abrangente.

Por fim, melhores algoritmos e técnicas de identificação de imagens podem ser implementadas, além do uso de abordagens mais avançadas como aprendizado profundo ou redes neurais convolucionais especializadas em análise de imagens médicas [LCW⁺14], permitindo ao modelo identificar padrões específicos com maior eficácia.

8.3 Tempo de Interpretação

Na primeira análise, realizada com o quiz de ECG, o tempo médio para o especialista concluir a interpretação de cada exame e registrar sua resposta foi de 1 minuto e 20 segundos. Por outro lado, o GPT Assistant concluiu suas análises em média dentro de 15 segundos por exame, representando um tempo de interpretação cerca de cinco vezes menor do que o do especialista. Essa diferença demonstra a velocidade significativamente superior da IA no diagnóstico, sem considerar a velocidade adicional em gerar um possível laudo.

Na segunda etapa da análise, foi observado que o tempo necessário para interpretar exames considerados "normais" era substancialmente menor do que o tempo exigido para interpretar traçados com anomalias, tanto pelo especialista quanto pela IA. Para a IA, o tempo médio de interpretação foi de 7 segundos para traçados normais e 15 segundos para traçados anômalos. Para o especialista, identificar um exame normal demandava "poucos segundos", de acordo com seu relato, enquanto o tempo médio para interpretar traçados com alguma anomalia subia para aproximadamente 2 minutos.

Essa diferença reflete a complexidade envolvida na identificação de anomalias. Segundo o especialista, "fica claro assim que vejo o exame se o mesmo se trata de um exame com padrões normais ou não; identificar a anomalia, quando existente, é o maior desafio". Um comportamento semelhante foi observado na IA, que também apresentava maior rapidez em categorizar traçados normais, mas demandava mais tempo para processar e identificar a natureza das anomalias. Essa observação reforça o potencial da IA para triagens rápidas, especialmente em exames normais, embora o tempo de análise de traçados anômalos ainda permaneça inferior ao do especialista humano, a maior assertividade do mesmo perante a IA invalida o diferencial de tempo.

8.4 Precificação

A tabela de valores do OpenAI separa a precificação entre os modelos do GPT e por valor de input/output. A medida base para o cálculo do valor final da requisição são os tokens, esses que podem ser vistos como pedaços de palavras e em uma escala de 1000 palavras, 750 tokens são gerados (Figura 8.4). A proporção de input para output é de quarenta e quatro para um, ou seja, para os oitenta e oito mil tokens utilizados ao total, apenas dois mil deles são output. A medida informada aqui para análise considera o total de tokens, sem diferenciar entre output e input, portanto para a relação de U\$D 2.50 / 1 Milhão de tokens será utilizado o valor total de 88.068 tokens.

Model	Pricing	Pricing with Batch API*
gpt-4o	\$2.50 / 1M input tokens	\$1.25 / 1M input tokens
	\$1.25 / 1M cached** input tokens	
	\$10.00 / 1M output tokens	
gpt-4o-2024-08-06	\$2.50 / 1M input tokens	\$1.25 / 1M input tokens
	\$1.25 / 1M cached** input tokens	
	\$10.00 / 1M output tokens	
gpt-4o-audio-preview	Text	
	\$2.50 / 1M input tokens	
	\$10.00 / 1M output tokens	
	Audio***	
	\$100.00 / 1M input tokens	
	\$200.00 / 1M output tokens	
gpt-4o-audio-preview-2024-10-01	Text	
	\$2.50 / 1M input tokens	
	\$10.00 / 1M output tokens	
	Audio***	
	\$100.00 / 1M input tokens	
	\$200.00 / 1M output tokens	
gpt-4o-2024-05-13	\$5.00 / 1M input tokens	\$2.50 / 1M input tokens
	\$15.00 / 1M output tokens	

Figura 8.4 – Tabela de preços da OpenAI

Além desse custo, a OpenAI também cobra pelo uso do módulo GPT Assistant (Figura 8.5), e em caso de uso das ferramentas disponíveis a cobrança será ajustada. Nesse caso, o custo adicional por uso dessas ferramentas não se aplica, já que não utilizamos a função de interpretador de código e o custo por armazenamento de vetor (forma como nossa imagem é armazenada na thread) é a partir de 1 GB de armazenamento, e o documento importado como base de conhecimento (Diretrizes da Sociedade Brasileira de Cardiologia sobre Análise e Emissão de laudos eletrocardiográficos) possui 3.6 Mb de tamanho, o que fica dentro da 1 GB disponibilizada gratuitamente.

Assistants API

The Assistants API and its tools make it easy for developers to build AI assistants in their applications. The tokens used for the Assistant API are billed at the chosen language model's per-token input / output rates.

[Learn about Assistants API ↗](#)

Additionally, we charge the following fees for tool usage:

Tool	Input
Code Interpreter	\$0.03 / session
File Search	\$0.10 / GB of vector-storage per day (1 GB free)

GB refers to binary gigabytes (also known as gibibyte), where 1 GB is 2^{30} bytes.

Figura 8.5 – Tabela de preços da OpenAI Assistant

Sendo assim, para a análise feita comparando os resultados da IA contra o Especialista, em apenas uma thread foram enviados 23 exames para interpretação, o que retornou um total de 88.068 tokens. Convertendo os milhares de tokens em dólares temos USD 0,22 e R\$1,27 em reais como valor total da thread (dólar a US\$5.77 no momento dessa dissertação). Dividindo o valor da thread pelo total de exames interpretados chegamos no preço por exame de 5 centavos, ou R\$0.05.

Para a segunda análise, o formato de resposta esperado como saída do Assistant foi alterado de acordo com as respostas padrão do especialista, variando o custo por thread. Essa alteração na resposta da IA tornou possível analisar os resultados com maior precisão ao comparar as respostas dos participantes, já que o formato era o mesmo (Figura 8.6).

ESPECIALISTA	GPT
Bradicardia sinusal.	Bradicardia sinusal.
Traçado dentro dos padrões de normalidade.	Traçado dentro dos padrões de normalidade.

Figura 8.6 – Comparação respostas especialista e OpenECG

Interpretar os 133 exames enviados para o Assistant resultou em 820.731 tokens e usando a mesma relação dos USD 2,50 para cada milhão de tokens, temos USD 2,05 como total em dólares e R\$11,82 total em reais. Fazendo a mesma conversão, mas com tamanhos de outputs diferentes cada exame neste caso custou R\$0,08 reais ou 8 centavos.

Sabendo agora o valor médio em reais por exame, podemos comparar e avaliar os valores praticados na região da Serra Gaúcha. Em uma clínica médica localizada em Caxias do Sul que presta serviços cardiológicos, o valor cobrado para a realização do exame (interpretação do traçado de ECG com emissão de laudo, assinado por médico com Registro de Qualificação de Especialidades em Cardiologia) é de R\$50,00 para pacientes

sem convênio (particular) e R\$25,00, em média, para pacientes com plano de saúde. Em proporção, para cada ECG interpretado tradicionalmente pode-se interpretar 625 exames usando OpenECG, considerando 8 centavos por interpretação.

Concluindo a análise de preço, temos uma disparidade expressiva de uma metodologia de interpretação para outra e assim que a IA alcançar níveis de precisão maiores, o valor de mercado para interpretação de ECG certamente será impactado.

Para concluir a pesquisa, é importante ressaltar a precisão do modelo foi comparada a apenas um especialista, e a validação dos resultados também foi realizada com base no mesmo profissional. Essa abordagem pode introduzir viés nos resultados, pois eventuais erros interpretativos do especialista seriam refletidos no treinamento do modelo, fazendo com que ele reconheça interpretações equivocadas como corretas. Isso afeta diretamente a precisão geral do sistema e sua capacidade de generalização.

Para mitigar esse viés e aprimorar a ferramenta, pode-se consolidar o conhecimento de múltiplos especialistas em uma mesma base conhecimento, aumentando a robustez do modelo. Ao utilizar diagnósticos e interpretações provenientes de uma variedade de profissionais qualificados, reduz-se o risco de dependência de um único ponto de vista, permitindo que o modelo aprenda a lidar com divergências interpretativas.

9. TRABALHOS FUTUROS

Com base nos resultados apresentados e nas limitações identificadas, diversas oportunidades de aprimoramento e expansão podem ser exploradas no futuro. A seguir, são delineadas as principais direções para os próximos trabalhos:

9.1 Aprimoramento da Precisão da Inteligência Artificial

Uma das áreas mais promissoras para futuros desenvolvimentos é o aprimoramento da precisão do modelo de inteligência artificial utilizado. Isso pode ser alcançado por meio da adoção de algoritmos mais avançados ou pela criação de um modelo próprio que seja especificamente treinado para identificar padrões nas imagens médicas de forma mais precisa. Tal abordagem incluiria o uso de conjuntos de dados mais amplos e diversificados, permitindo que a IA reconheça características sutis e raras, aumentando sua eficácia diagnóstica em casos clínicos mais desafiadores.

9.2 Expansão dos Tipos de Exames Interpretados

Uma evolução natural da solução é ampliar os tipos de exames que podem ser interpretados pela plataforma. Além dos ECGs, podem ser incluídos exames gráficos e de imagem, como eletroencefalogramas, espirometrias e até exames de imagem médica, como radiografias ou ultrassonografias. Essa expansão não apenas aumentaria a aplicabilidade da ferramenta, mas também a tornaria mais versátil e abrangente no suporte ao diagnóstico clínico em diferentes especialidades médicas.

9.3 Melhoria da Segurança da Plataforma

Outro aspecto crítico a ser abordado em trabalhos futuros é o fortalecimento da segurança da plataforma. Isso inclui a implementação de métodos mais avançados para proteger o código-fonte, a infraestrutura e os dados armazenados, assegurando a confidencialidade, integridade e disponibilidade das informações médicas. Soluções como criptografia de ponta a ponta, autenticação multifator e auditorias de segurança regulares podem ser exploradas para tornar o sistema mais robusto e confiável.

10. CONSIDERAÇÕES FINAIS

As análises realizadas ao longo do desenvolvimento do OpenECG demonstraram que a plataforma funciona adequadamente, atendendo ao objetivo principal de automatizar a interpretação inicial de exames de ECG. A utilização de inteligência artificial para analisar imagens de ECG revelou-se uma ferramenta promissora, contribuindo para a democratização do acesso a diagnósticos médicos. Contudo, é necessário destacar que ainda há desafios significativos que devem ser superados antes do lançamento da plataforma.

A assertividade das interpretações geradas pela inteligência artificial é um ponto central a ser aprimorado. Apesar dos avanços obtidos, o sistema ainda apresenta limitações que comprometem a confiabilidade em determinados cenários clínicos. A precisão das análises deve ser elevada para atingir os padrões estabelecidos pela SBC, assegurando que a plataforma possa oferecer resultados consistentes e úteis na prática médica. Este aspecto é essencial para que o OpenECG possa competir com soluções de mercado e ganhar a confiança dos profissionais de saúde.

Adicionalmente, a segurança do código da plataforma precisa ser reforçada. O ambiente médico exige conformidade rigorosa com legislações de proteção de dados, como a Lei Geral de Proteção de Dados. O código atual, embora funcional, requer aprimoramentos em suas camadas de proteção contra vulnerabilidades. Sem os ajustes necessários, a plataforma pode comprometer a privacidade dos usuários e a integridade dos dados médicos armazenados.

O armazenamento local das imagens de ECG, utilizado nesta versão do OpenECG, também apresenta limitações práticas para um ambiente de produção. Em larga escala, esse método pode se tornar impraticável devido à crescente demanda por espaço físico e as dificuldades e preço de manutenção. Uma alternativa mais escalável e eficiente seria a adoção de serviços de armazenamento em nuvem, que oferecem maior flexibilidade, segurança e integração com outras tecnologias.

Apesar dessas limitações, o potencial do OpenECG é perceptível. A plataforma pode ir além da interpretação de ECGs, expandindo-se para incluir outros tipos de exames médicos e recursos de apoio diagnóstico, como integração com sistemas hospitalares (HIS, PACS e RIS) e a oferta de relatórios mais abrangentes. A modularidade da arquitetura de micro serviços empregada no projeto facilita a incorporação de novas funcionalidades e adaptações tecnológicas no futuro.

Assim, o OpenECG posiciona-se como um projeto inovador, com capacidade de impactar significativamente a área da saúde. Para atingir todo o seu potencial, é essencial investir na melhoria contínua de sua inteligência artificial, na robustez da segurança do código e na adoção de tecnologias que garantam escalabilidade. A exploração

dessas oportunidades poderá consolidar o OpenECG como uma solução referência em diagnósticos assistidos por inteligência artificial, contribuindo para um sistema de saúde mais inclusivo e eficiente.

REFERÊNCIAS BIBLIOGRÁFICAS

- [AdPM17] Antonio de Padua Mansur, D. F. “Taxas de mortalidade por doenças cardiovasculares e câncer na população brasileira com idade entre 35 e 74 anos, 1996-2017”, *Instituto do Coração do Hospital das Clínicas da Faculdade de Medicina da Universidade de São Paulo*, vol. 1–1, 2017, pp. 12.
- [Buc06] Buchanan, B. G. “A (very) brief history of artificial intelligence”, *AI Magazine Volume*, vol. 1–4, 2006, pp. 8.
- [Cod24] Code, V. S. “Tutorial: Get started with visual studio code”. Capturado em: <https://code.visualstudio.com/docs/getstarted/getting-started>, Nov 2024.
- [D. 07] D. LU, Q. W. “A survey of image classification methods and techniques for improving classification performance”, *International Journal of Remote Sensing*, vol. 870–5, 2007, pp. 12.
- [DIC24] DICOM. “Ps3.1: Dicom ps3.1 2024e - introduction and overview”, *NEMA*, vol. 1–1, 2024, pp. 34.
- [Dmi14] Dmitry Namiot, M. S.-S. “On micro-services architecture”, *International Journal of Open Information Technologies*, vol. 2–9, 2014, pp. 4.
- [Doc24] Docker. “What is docker?” Capturado em: <https://docs.docker.com/get-started/docker-overview/>, Nov 2024.
- [eRMT09] e Rosália Moraes Torres, R. T. G. “Breve história da eletrocardiografia”. Capturado em: <https://rmmg.org/artigo/detalhes/323#:~:text=O%20primeiro%20eletrocardiograma%20humano%20foi,acompanhado%20por%20uma%20oscila%C3%A7%C3%A3o%20el%C3%A9trica.>, Ago 2024.
- [FDS21] Fábio Duarte Schwalm, Enrique Falceto de Barros, I. d. O. C. S. K. K. P. L. V. T. M. M. d. A. V. L. L. P. “Medicina rural”, *NEMA*, vol. 1–1, 2021, pp. 34.
- [Fla24] Flask. “Users guide”. Capturado em: <https://flask.palletsprojects.com/en/stable/>, Nov 2024.
- [Gua18] Guarisch, A. “Artigo: Uma breve história do eletrocardiograma”, *O Globo*, vol. 1–1, 2018, pp. 1.
- [Jim24] Jim Holdsworth, M. S. “O que é deep learning?” Capturado em: <https://www.ibm.com/br-pt/topics/deep-learning>, Nov 2024.

- [LCW⁺14] Li, Q.; Cai, W.; Wang, X.; Zhou, Y.; Feng, D. D.; Chen, M. “Medical image classification with convolutional neural network”. In: 2014 13th International Conference on Control Automation Robotics Vision (ICARCV), 2014, pp. 844–848.
- [Mac12] Macêdo Firmino, S. P. e. R. V. “Pacs – sistema de comunicação e arquivamento de imagens médicas: Visão introdutória e usabilidade no sistema de saúde brasileiro”, *Instituto Nacional de Telecomunicações*, vol. 2–1, 2012, pp. 8.
- [MyS24] MySQL. “Mysqldb user’s guide”. Capturado em: https://mysqlclient.readthedocs.io/user_guide.html, Nov 2024.
- [Ngy24] Nginx. “nginx”. Capturado em: <https://nginx.org/>, Nov 2024.
- [Ope24] OpenAI. “Vision”. Capturado em: <https://platform.openai.com/docs/guides/vision>, Nov 2024.
- [PMc24] PMcardio. “Ai ecg interpretation care coordination”. Capturado em: <https://www.powerfulmedical.com/>, Nov 2024.
- [Pyt24] Python. “Beginners guide”. Capturado em: <https://wiki.python.org/moin/BeginnersGuide/Overview>, Nov 2024.
- [Suk15] Sukienik, B. “Atlas de Eletrocardiografia”. Porto Alegre, RS: Campus, 2015, 287p.
- [Viv20] Vivek Kaul, MD, F. S. E. P.-C. S. A. G. M. F. “History of artificial intelligence in medicine”, *Gastrointestinal Endoscopy*, vol. 92–4, 2020, pp. 6.
- [Vue24] Vuejs. “What is vue?” Capturado em: <https://vuejs.org/guide/introduction.html>, Nov 2024.