UNIVERSIDADE FEDERAL DO PARANÁ

EDUARDO COSTA MATOS
MARIANE ROESLER
VICTORIA KUNZ VIEIRA

ARQUITETURA DE AGENTES PARA ORQUESTRAÇÃO DE MODELOS MULTIMODAIS FUNDACIONAIS EM ONCOLOGIA DE PRECISÃO

EDUARDO COSTA MATOS
MARIANE ROESLER
VICTORIA KUNZ VIEIRA

ARQUITETURA DE AGENTES PARA ORQUESTRAÇÃO DE MODELOS MULTIMODAIS FUNDACIONAIS EM ONCOLOGIA DE PRECISÃO

Trabalho de conclusão de curso apresentado ao curso de graduação em Análise e Desenvolvimento de Sistemas, Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, como requisito parcial à obtenção do título de tecnólogo.

Orientador: Prof. Dr. Mauro Antonio Alves Castro.

Co-orientador: Victor Henrique Apolonio.

UNIVERSIDADE FEDERAL DO PARANÁ
SETOR DE EDUCAÇÃO PROFISSIONAL E TECNOLÓGICA
Rua Alcides Vieira Arcoverde 1225, - - Bairro Jardim das Américas, Curitiba/PR, CEP 81520-260
Telefone: 3360-5000 - https://ufpr.br/

Ata de Reunião

TERMO DE APROVAÇÃO

EDUARDO COSTA MATOS
MARIANE ROESLER
VICTORIA KUNZ VIEIRA

ARQUITETURA DE AGENTES PARA ORQUESTRAÇÃO DE MODELOS MULTIMODAIS FUNDACIONAIS EM ONCOLOGIA DE PRECISÃO

Monografia aprovada como requisito parcial à obtenção do título de Tecnólogo em Análise e Desenvolvimento de Sistemas, do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná.

Prof. Mauro Antonio Alves Castro
Orientador – SEPT/UFPR

Prof. Victor Henrique Apolonio
Co-orientador – Bioinformática/SEPT/UFPR

Arquitetura de Agentes para Orquestração de Modelos Multimodais Fundacionais em Oncologia de Precisão

Eduardo Costa Matos[1,*], Mariane Roesler[1,*], Victoria Kunz Vieira[1,*],
Victor Henrique Apolonio[2,†], Mauro Antonio Alves Castro[2,†]

[*]*Os três primeiros autores contribuíram igualmente para este trabalho.*
[†]*Autores correspondentes: mauro.castro@ufpr.br, victor.henrique.apolonio@gmail.com*

[1]*Setor de Educação Profissional e Tecnológica, Universidade Federal do Paraná, Curitiba, Brasil*
[2]*Bioinformatics and Systems Biology Laboratory, Federal University of Paraná, Curitiba, PR 81520-260, Brazil*

Resumo: **O crescimento exponencial do volume de dados em saúde e a complexidade inerente à patologia digital demandam ferramentas de apoio à decisão capazes de integrar informações multimodais. Este trabalho apresenta o desenvolvimento de uma arquitetura de software baseada em um agente de IA, projetada para integrar o *vision–language foundation model* MUSK (*Multimodal transformer with Unified maSKed modeling*) ao fluxo de diagnóstico histopatológico. Metodologicamente, a solução evoluiu de uma prova de conceito baseada em *Retrieval-Augmented Generation* (RAG) para uma arquitetura modular, composta por uma interface conversacional, uma camada de orquestração e um agente de inteligência artificial responsável por coordenar ferramentas de busca vetorial. Os resultados da verificação funcional indicam que a arquitetura é capaz de interpretar intenções em linguagem natural, acionar o modelo MUSK para extração de embeddings e recuperar casos clínicos semelhantes com seus respectivos metadados. Conclui-se que a solução demonstra a viabilidade técnica de integrar um modelo multimodal fundacional ao processo de apoio ao diagnóstico em patologia digital. Isso é alcançado por meio de uma arquitetura de software baseada em um agente de IA, na qual a separação entre interface, orquestração e serviços de inferência e busca vetorial favorece a modularidade, a extensibilidade e a evolução futura de sistemas interativos de apoio à decisão.**

Palavras-chave: **arquitetura de agente, histopatologia, modelos fundacionais, inteligência artificial, recuperação multimodal, VLMs.**

Abstract: **The exponential growth of big data in healthcare and the inherent complexity of digital pathology require decision support tools capable of integrating multimodal information. This work proposes an agent-oriented software architecture designed to integrate the vision-language foundation model MUSK (Multimodal transformer with Unified maSKed modeling) into the histopathological diagnostic workflow. Methodologically, the solution evolved from a Retrieval-Augmented Generation (RAG) proof of concept to a modular architecture, composed of a conversational interface, an orchestration layer, and an Artificial Intelligence agent responsible for coordinating vector search tools. Functional verification results indicate that the architecture is capable of interpreting natural language intentions, triggering the MUSK model for embedding extraction, and retrieving similar clinical cases with their respective metadata. It is concluded that the implemented approach proved effective for the proposed orchestration, exhibiting characteristics of modularity and extensibility that suggest a promising path for building interactive decision support systems in digital pathology.**

Keywords: **agent-based architecture, histopathology, foundational models, artificial intelligence, multimodal retrieval, VLMs.**

I. INTRODUÇÃO

O crescimento contínuo da capacidade de geração e armazenamento de informações tem transformado diversos domínios científicos e tecnológicos, criando novos desafios para a análise, interpretação e utilização eficiente desses dados em larga escala. Nesse cenário, o conceito de *big data* tornou-se central para descrever situações em que o volume, a variedade e a velocidade de produção de dados excedem a capacidade dos sistemas tradicionais de processamento.

Embora frequentemente associado às tecnologias contemporâneas, o termo *big data* remonta a 1997, quando Michael Cox e David Ellsworth o empregaram em uma apresentação no IEEE para caracterizar os desafios de visualização de grandes conjuntos de dados e as limitações dos sistemas computacionais convencionais [1]. Desde então, passou a designar um conjunto de práticas e arquiteturas voltadas a lidar com dados cuja escala supera os limites de bancos de dados tradicionais [1].

No contexto da saúde, esse crescimento exponencial assume proporções particularmente relevantes: estima-se que o setor de *healthcare* seja responsável por aproximadamente 30% de toda a produção de dados em nível global e que, até 2025, ocorra um incremento adicional de 36% [2], tornando progressivamente inviável a análise manual por profissionais da área [3]. Além de ultrapassar os limites humanos de síntese, a anotação manual de grandes volumes de dados, especialmente imagens médicas, é cara, demorada e sujeita a erros [4]. Em resposta a essas limitações, técnicas de computação e de inteligência artificial (IA) têm sido empregadas como ferramentas para

automatizar etapas de processamento e viabilizar a extração de conhecimento biologicamente significativo [5].

De forma geral, IA pode ser entendida como um campo da ciência da computação dedicado ao estudo e desenvolvimento de sistemas capazes de perceber informações do ambiente, processá-las e tomar decisões ou executar ações de maneira considerada inteligente ou racional [6]. No contexto da saúde, isso se traduz no uso de algoritmos e sistemas computacionais que analisam grandes conjuntos de informações para identificar padrões, classificar elementos, realizar previsões e apoiar a tomada de decisão em diferentes cenários clínicos e gerenciais [7], [8].

Nessa perspectiva, o *machine learning* (ML) configura-se como uma subárea da IA voltada ao desenvolvimento de métodos que permitem que sistemas computacionais aprimorem seu desempenho em tarefas específicas a partir de dados e exemplos, em vez de depender exclusivamente de regras explicitamente programadas [6]. Em aplicações médicas, o "aprendizado" desses sistemas decorre da exposição a dados clínicos previamente observados, como registros eletrônicos de saúde, que são utilizados para treinar modelos capazes de apoiar tarefas de classificação e predição de desfechos [7].

Dentre as abordagens de ML, os modelos generativos, especialmente os *Large Language Models* (LLMs), têm se destacado, uma vez que atuam como interfaces humano–computador avançadas capazes de processar linguagem natural e gerar respostas coerentes em tempo real [9]. Do ponto de vista conceitual, Russell [6] descreve o *Natural Language Processing* (NLP) como o conjunto de técnicas que permite a computadores interpretar e produzir sentenças em linguagem natural; os LLMs podem ser compreendidos, nesse contexto, como uma realização dessa abordagem, funcionando como modelos estatísticos treinados para prever sequências linguísticas e gerar texto contextualizado. A combinação entre essas capacidades de compreensão e geração automática de linguagem natural confere aos LLMs elevado potencial para aplicações médicas inteligentes, contribuindo para o desenvolvimento de dispositivos e soluções de saúde mais eficientes [10].

A evolução dessas abordagens generativas, combinada com a crescente disponibilidade de grandes volumes de dados biomédicos, insere-se no contexto mais amplo do surgimento dos chamados *foundation models*. Estes são modelos de aprendizagem treinados em conjuntos de dados extensos, em geral não rotulados, e capazes de executar diferentes tarefas generativas e discriminativas, além de serem adaptados para aplicações específicas [11], [12].

À medida que modelos de IA mais sofisticados, como *foundation models*, passam a ser considerados para aplicações em saúde, a discussão transcende a restrição ao seu desempenho preditivo e passa a envolver também a forma como são incorporados em sistemas de apoio à decisão [7], [8]. Em contextos clínicos reais, as decisões habitualmente dependem da combinação de diferentes fontes de dados, de consultas sucessivas e da interação contínua com profissionais de saúde. Isso torna relevante o estudo de arquiteturas de software que explicitem quando e como modelos de IA são acionados, quais informações utilizam e de que maneira suas recomendações são apresentadas aos usuários [13].

Entre as alternativas propostas para estruturar sistemas baseados em ML e LLMs, destacam-se as arquiteturas baseadas em agentes. No enquadramento clássico de Russell e Norvig, um agente é definido como uma entidade que percebe o ambiente e seleciona ações para atingir objetivos específicos [6]. Neste trabalho, em consonância com descrições recentes de arquiteturas orientadas a agente e de sistemas *agentic*, empregamos o termo "agente" para designar um componente de software que atua como um módulo de raciocínio autônomo orientado a objetivos, responsável por interpretar entradas (por exemplo, consultas em linguagem natural) e decidir quais ações ou ferramentas acionar [14], [15], [16], [17].

Nas formulações contemporâneas, essa capacidade de interpretação e tomada de decisão tem sido frequentemente sustentada por LLMs, que funcionam como o núcleo cognitivo do agente, permitindo-lhe compreender instruções em linguagem natural, manter coerência contextual e adaptar o fluxo de ações ao longo da execução [18]. Além disso, tornou-se comum que esses agentes coordenem o uso de ferramentas externas (incluindo bancos de dados, modelos de visão e módulos de recuperação) acionando-as conforme a necessidade da tarefa. Essa integração configura uma evolução natural do arcabouço clássico para abordagens que combinam, de forma coordenada, percepção, tomada de decisão e uso de recursos especializados [19].

Em aplicações médicas, agentes de IA têm sido discutidos como componentes capazes de receber dados clínicos, acionar modelos de IA apropriados e retornar recomendações, sínteses ou alertas para apoiar a tomada de decisão em um ambiente descrito como complexo e em constante transformação [13]. Inspirado por essa perspectiva, este estudo considera uma arquitetura orientada a agente, na qual um componente de software atua como intermediário entre os dados disponíveis e os modelos de IA, coordenando seu uso em um cenário delimitado de apoio ao diagnóstico.

Para que tal agente seja capaz de lidar de forma integrada com informações visuais e textuais provenientes da patologia digital, faz-se necessária a utilização de um modelo multimodal de amplo escopo. Nesse trabalho, essa função é desempenhada pelo *Multimodal transformer with Unified maSKed modeling* (MUSK), um *foundation model* pré-treinado em larga escala sobre aproximadamente 50 milhões de imagens de patologia, derivadas de dezenas de tipos tumorais, e sobre um grande volume de texto biomédico [20], [21].

Assim, o problema investigado neste trabalho consistiu em definir e avaliar uma arquitetura baseada em agentes, capaz de integrar um *vision–language foundation model* (VLFM) ao processo de apoio ao diagnóstico em patologia digital. O objetivo geral foi projetar e implementar essa arquitetura, utilizando o MUSK como componente central para a análise de lâminas histológicas, adotando como premissa técnica a viabilidade de tal integração. Para examinar essa premissa, utilizou-se o conjunto de dados público *Histological images for MSI vs. MSS classification in gastrointestinal cancer, FFPE*

samples, de Kather [22], composto por patches de imagens histológicas normalizados e derivados de lâminas FFPE de tumores colorretais e gástricos da coorte TCGA, rotulados quanto ao status de instabilidade de microssatélites (MSS ou MSIMUT).

II. REVISÃO DE LITERATURA

A. Multimodalidade na Tomada de Decisão Clínica e na Patologia Digital

A tomada de decisão clínica contemporânea é, por natureza, multimodal: combina dados clínicos estruturados, anotações textuais, imagens diagnósticas e, cada vez mais, informações ômicas. Xiang et al. afirmam que a decisão clínica é "impulsionada por dados multimodais, incluindo notas clínicas e características patológicas" e que abordagens de IA capazes de integrar essas fontes "têm um potencial significativo para avançar o cuidado clínico" (tradução livre dos autores) [20]. De forma complementar, AlSaad et al. observam que "dados multimodais são predominantes e cruciais para decisões clínicas informadas", abrangendo desde imagens médicas até texto livre e dados ômicos (tradução livre dos autores) [23]. No contexto oncológico, revisões sistemáticas indicam que a integração de imagem, genômica e outros marcadores por meio de *machine learning* favorece o desenvolvimento de biomarcadores mais precisos e uma melhor predição de prognóstico [24], [25].

Na patologia digital, essa necessidade de integração é particularmente evidente devido ao papel central das *whole-slide images* (WSIs), isto é, lâminas histológicas digitalizadas em alta resolução, e de relatórios descritivos de padrões morfológicos. Ahmad et al. mostram que WSIs figuram como modalidade primária em muitas ferramentas de *visual analytics*, embora, em determinados cenários, dados clínicos estruturados ou genômicos também assumam papel central [26], [27]. Em paralelo, métodos recentes de *image–text retrieval* baseados em alinhamento multimodal exploram explicitamente a correspondência entre imagens histológicas e relatórios associados, permitindo consultas em que descrições textuais recuperam padrões visuais específicos e vice-versa [28], [29]. Sem esse alinhamento multimodal, a busca manual por casos semelhantes em grandes acervos digitais permanece limitada: o patologista precisa inspecionar lâminas gigapixel visualmente, com anotações muitas vezes globais e pouco específicas para a região de interesse, o que torna o processo lento, pouco reprodutível e difícil de escalar [30], [31]. Ainda que métodos clássicos baseados em *handcrafted features* continuem presentes em alguns fluxos clínicos, a literatura indica que sua capacidade de escalar e capturar a variabilidade morfológica é restrita quando comparada a abordagens multimodais recentes.

Além das dificuldades práticas na recuperação manual de casos, há desafios estruturais de escalabilidade, variabilidade e padronização em patologia digital. O tamanho massivo das WSIs, "combinado com a ausência de anotações em nível de pixel, impõe desafios significativos" ao desenvolvimento de métodos eficientes de análise (tradução livre dos autores) [32], enquanto diferenças de coloração, scanners e protocolos impactam a generalização de modelos entre instituições [30]. Revisões recentes apontam ainda altos custos computacionais, validação externa limitada, barreiras regulatórias e dificuldades de integração em fluxos clínicos reais como obstáculos à adoção em larga escala [26], [25]. Em paralelo, a interoperabilidade entre sistemas e padrões heterogêneos de dados exige esforços contínuos de normalização de formatos e metadados [33]. Esse cenário abre espaço para *foundation models* multimodais, como o MUSK, e para arquiteturas que os integrem de forma sistemática, tema desenvolvido na subseção seguinte sobre *vision–language models* específicos de domínio.

B. Multimodal Foundation Models (Vision–Language Models, VLMs)

Vision–Language Models (VLMs) aprendem representações conjuntas de imagens e textos em um espaço vetorial compartilhado, permitindo alinhamento direto entre modalidades. Arquiteturas como CLIP e BLIP utilizam objetivos contrastivos para aproximar pares imagem–texto e afastar pares não relacionados no *embedding space*. Zhang et al. apontam que VLMs pré-treinados "capturam um conhecimento rico de correspondência visão–linguagem e podem realizar predições zero-shot ao associar os embeddings de quaisquer imagens e textos fornecidos" (tradução livre dos autores) [34]. Gao et al. descrevem essa abordagem como "um novo paradigma [...] para aprender diretamente o alinhamento entre imagens e textos brutos em um cenário de vocabulário aberto" (tradução livre dos autores) [35]. Em contraste com pipelines tradicionais de *feature engineering*, que dependiam de atributos projetados manualmente e classificadores específicos, VLMs aprendem representações gerais a partir de grandes volumes de dados, frequentemente com supervisão fraca ou auto-supervisão.

Conceitualmente, essa mudança desloca o foco da extração manual de atributos para embeddings genéricos que podem ser reutilizados em múltiplas tarefas. Em vez de projetar features específicas para cada problema, VLMs permitem adaptação via *zero-shot*, *few-shot* ou *fine-tuning* leve, favorecendo generalização mais robusta. Pesquisas recentes destacam que modelos multimodais combinam "múltiplas modalidades de dados" para habilitar capacidades unificadas em tarefas de compreensão, geração e recuperação de informação (tradução livre dos autores) [36], enquanto AlSaad et al. enfatizam seu papel emergente em cenários clínicos complexos [23]. Ao mesmo tempo, revisões em aplicações biomédicas reconhecem que, em alguns contextos, embeddings de VLMs continuam sendo combinados a features manuais para aumentar interpretabilidade ou robustez [27], [37].

A literatura descreve uma ampla família de VLMs de domínio geral, como CLIP, SigLIP, BLIP/BLIP-2, PaLI-X, Flamingo e GIT, e suas contrapartes biomédicas, incluindo modelos como EyeCLIP, adaptados a modalidades diagnósticas específicas. Shi et al. mostram que EyeCLIP "supera de forma significativa os modelos de referência em diversas tarefas, incluindo classificação *zero-shot*, recuperação multimodal e VQA" (tradução livre dos autores) [38]. Modelos de maior

escala e foco biomédico, como o MUSK para patologia digital, estendem essa linha ao integrar imagens histopatológicas e textos clínicos para tarefas de apoio ao diagnóstico, recuperação multimodal e *visual question answering* (VQA) [20]. Esses avanços consolidam VLMs como componentes centrais em aplicações biomédicas que exigem alinhamento fino entre visão e linguagem.

C. VLMs Biomédicos Especializados: o Caso do MUSK

O Multimodal transformer with Unified maSKed modeling (MUSK) é um biomedical foundation model multimodal projetado especificamente para patologia digital, com o objetivo de capturar padrões morfológicos e semânticos característicos da prática clínica. Seu treinamento ocorre em duas etapas complementares: (i) pré treinamento com unified masked modeling usando 50 milhões de patches histopatológicos e 1 bilhão de tokens textuais não pareados, o que configura uma forma de aprendizado auto-supervisionado de representações visuais e textuais; e (ii) alinhamento multimodal supervisionado com 1 milhão de pares imagem–texto oriundos do QUILT-1M, consolidando um espaço latente compartilhado entre visão e linguagem [20].

Os autores enfatizam que, ao contrário de VLMs genéricos "baseados em aprendizagem contrastiva, que dependem de dados pareados imagem–texto para o pretreinamento" (tradução livre dos autores), o MUSK explora de forma intensiva dados não pareados específicos de patologia antes da fase de alinhamento supervisionado [20]. Essa estratégia é particularmente relevante em domínios biomédicos, em que conjuntos de dados anotados de forma consistente são escassos e custosos de obter.

Em termos funcionais, o MUSK suporta um conjunto amplo de tarefas médicas, especialmente aquelas que dependem de similaridade e *cross-modal retrieval*, bem como *zero-shot image classification* e VQA, todas avaliadas empiricamente no artigo. O modelo demonstra desempenho superior em *retrieval* bidirecional (*image-to-text* e *text-to-image*), superando PLIP, CONCH e BiomedCLIP em múltiplos benchmarks [20]. Para *zero-shot* e *few-shot classification* em nível de patch, aspecto central na patologia digital, o MUSK apresenta ganhos expressivos (incluindo melhorias da ordem de 20%–27% em bases como SkinCancer e PanNuke) [20]. Essa performance decorre da maneira como o modelo representa padrões morfológicos discretizados em patches, capturando detalhes histológicos que VLMs treinados em imagens naturais tendem a não modelar adequadamente. Os autores sintetizam que o MUSK "fornece uma abordagem poderosa para aprender representações de imagem mais eficazes para classificação em patologia" (tradução livre dos autores) [20].

Embora o MUSK ofereça uma base sólida para pipelines de recuperação multimodal e classificação em nível de patch, sua capacidade de generalização para diferentes domínios de patologia depende da diversidade dos dados de treinamento e, potencialmente, de estratégias de adaptação local. Ainda assim, o modelo se destaca como um dos exemplos mais robustos de VLM biomédico especializado atualmente disponíveis, justificando sua adoção em sistemas que exigem *retrieval* multimodal eficiente e interpretável em patologia digital.

A integração de modelos multimodais como o MUSK em aplicações reais demanda arquiteturas capazes de organizar decisões, consultas e o acionamento de ferramentas, tema amplamente discutido em frameworks recentes baseados em agentes [14], [15]. A literatura descreve desde arranjos de agente único até sistemas mais amplos, denominados *agentic systems*, nos quais se incorporam capacidades de planejamento, reflexão e uso dinâmico de ferramentas [16], [17]. Em paralelo, abordagens multi-agente têm sido propostas para aumentar robustez e verificação [39], [40]. Esse panorama conceitual estabelece as bases para a subseção seguinte, dedicada ao papel do RAG e dos mecanismos de orquestração em arquiteturas orientadas a agentes.

D. Retrieval-Augmented Generation (RAG), arquiteturas de agente e MCP

Retrieval-Augmented Generation (RAG) é um paradigma em que modelos generativos são conectados a módulos externos de recuperação, baseados em embeddings e vector stores, permitindo incorporar conhecimento atualizado e relevante ao processo de geração [41], [42], [43]. Na configuração unimodal, documentos textuais são codificados em embeddings, indexados em vector stores e recuperados por meio de algoritmos de *approximate nearest neighbors* (ANN), o que viabiliza buscas eficientes em grandes volumes de dados [44], [45]. Na extensão multimodal, consultas textuais ou visuais são mapeadas para um espaço vetorial compartilhado por VLMs como CLIP ou MUSK, permitindo a recuperação simultânea de imagens e textos semanticamente relacionados [41], [42], [45]. O pipeline multimodal é usualmente descrito em termos de pré-processamento da consulta, codificação em embedding compartilhado, recuperação multimodal e geração condicionada nas evidências retornadas [41], [42].

Inicialmente, o RAG foi concebido como uma arquitetura central para integração entre recuperação e geração. Revisões recentes, porém, passam a situar essa abordagem dentro de modelos orientados a agentes e, em alguns casos, no contexto mais amplo dos chamados *agentic systems*, em que um ou mais agentes autônomos coordenam operações de busca, raciocínio, verificação e execução [46], [17]. Arquiteturas baseadas em agentes vão além de pipelines fixos de IA: em vez de fluxos rigidamente pré-definidos de chamadas a modelos e APIs, o sistema é estruturado em torno de agentes orientados a objetivos, capazes de planejar, decompor tarefas, refletir sobre sua própria execução e reconfigurar o fluxo dinamicamente [14], [15]. Em geral, esses agentes são apoiados por LLMs, que interpretam a intenção do usuário em linguagem natural, escolhem quais ferramentas acionar (por exemplo, um modelo de visão, um banco de dados, um módulo de simulação ou um componente RAG multimodal) e coordenam o uso desses módulos ao longo do tempo, ajustando a estratégia à medida que novas evidências são incorporadas [27], [17]. Nesse cenário, RAG deixa de ser a arquitetura "final" e passa a funcionar como um módulo

de busca semântica (unimodal ou multimodal) que o agente decide quando e como utilizar.

De acordo com descrições recentes, sistemas classificados como *agentic* podem envolver agentes dotados de mecanismos adicionais, como memória persistente, decomposição de tarefas e uso dinâmico de ferramentas externas [16], [17]. Nesse contexto, ferramentas correspondem a funções ou serviços externos (como módulos RAG, bancos de dados, simuladores ou APIs) acionados pelo agente conforme a necessidade da tarefa [14], [17]. O acesso a ferramentas é frequentemente mediado por protocolos padronizados, como o Model Context Protocol (MCP), que definem como agentes descobrem, acessam e utilizam recursos externos de forma segura, interoperável e auditável [47], [40], [48], [49], [50]. Protocolos como o MCP permitem que ferramentas sejam expostas como *capabilities* acessíveis sob demanda, promovendo modularidade e evolução independente dos componentes do sistema [47], [40], [48].

Sobre essa base, arquiteturas *multi-agent* especializadas em *retrieval, reasoning, verification* e *explanation* vêm sendo propostas para aumentar robustez, escalabilidade e transparência. Frameworks como MA-RAG e HM-RAG mostram que organizar RAG dentro de arquiteturas *multi-agent*, com divisão clara de capacidades e mecanismos de votação ou checagem de consistência, pode mitigar falhas de um único modelo, especialmente em tarefas de longo horizonte [39], [40], [46], [17]. Ao mesmo tempo, esses trabalhos enfatizam que tais ganhos trazem *trade-offs* relevantes, incluindo maior complexidade de coordenação, latência e custo computacional, tornando a escolha entre arquiteturas *single-agent, agentic* e *multi-agent* dependente do domínio, dos requisitos de segurança e das restrições de infraestrutura [39], [40], [17].

Protocolos de orquestração como o Model Context Protocol (MCP) têm sido propostos para padronizar a integração entre agentes e ferramentas, provendo um espaço de contexto compartilhado para consulta, registro de estados intermediários e coordenação de decisões [40], [51], [48], [52]. Nessa abordagem, capability servers expõem dados e serviços enquanto clientes agentic consomem essas capacidades, o que favorece interoperabilidade modular e evolução independente de componentes. A literatura recente trata MCP como uma tecnologia emergente, particularmente promissora para sistemas multi-agent em ambientes complexos e regulados, ainda em processo de consolidação como padrão [40], [48], [52].

III. Metodologia

Esta seção descreve o processo de desenvolvimento da solução proposta, desde a seleção do *vision–language foundation model* até a evolução da arquitetura do sistema, passando pela prototipagem inicial e pelas limitações encontradas. O foco recai tanto na justificativa técnica para o uso do MUSK quanto nas decisões de projeto que levaram à adoção de uma arquitetura orientada a agente de IA para apoio ao diagnóstico em patologia digital.

A. Seleção do modelo MUSK

A escolha do modelo multimodal constituiu o ponto de partida da metodologia. Dentre as alternativas disponíveis, optou-se pelo MUSK (*Multimodal transformer with Unified maSKed modeling*) devido à sua concepção específica para patologia digital e ao desempenho robusto em tarefas diretamente relacionadas a este trabalho, como *zero-shot/few-shot classification* em nível de patch e *bidirectional retrieval*. Esses resultados demonstram a capacidade do modelo de capturar padrões morfológicos relevantes e de sustentar consultas imagem–texto de maneira consistente, atributos essenciais para o protótipo proposto.

Outro fator determinante foi a validação científica do modelo. O MUSK foi apresentado e avaliado em um artigo publicado na *Nature*, o que reforça sua maturidade como *biomedical vision–language foundation model* e atesta a confiabilidade de suas representações multimodais. Essa validação independente, aliada ao desempenho superior frente a *VLMs* genéricos treinados em imagens naturais, justifica sua adoção como base técnica da solução desenvolvida. [20]

B. Configuração inicial e limitações encontradas

Apesar da atratividade do modelo, seu uso prático apresentou limitações. A distribuição do MUSK e do código de referência ainda é recente, e a comunidade de usuários é restrita, o que dificulta o processo de *troubleshooting*. Parte da metodologia envolveu, portanto, a etapa de engenharia necessária para tornar o modelo utilizável em um ambiente experimental próprio.

Um exemplo concreto dessas dificuldades foi a configuração do tokenizer. O MUSK depende do tokenizer XLM-RoBERTa [53], que não é automaticamente incluído em todos os diretórios do modelo quando o repositório é clonado. Essa ausência resultava em falhas recorrentes na inferência sempre que o pipeline multimodal exigia a etapa textual. Isso inviabilizava a geração de embeddings de texto, qualquer tarefa de alinhamento texto–imagem e, por consequência, as capacidades *zero-shot* e *few-shot* do modelo. Sem o tokenizer, apenas operações puramente visuais (como correspondência imagem–imagem) funcionavam corretamente. A solução exigiu a identificação manual do tokenizer apropriado e sua inserção explícita na estrutura de diretórios do modelo, restabelecendo o funcionamento completo das funcionalidades multimodais. Essa etapa ilustra que parte relevante da metodologia não se limitou ao desenho conceitual da solução, mas também à superação de barreiras de integração e execução do modelo em ambiente local.

C. Prototipagem inicial: arquitetura monolítica RAG multimodal

Uma vez estabelecido o funcionamento básico do MUSK, foi desenvolvida uma primeira prova de conceito em torno de uma arquitetura monolítica de *Retrieval-Augmented Generation* (RAG) multimodal. Nessa versão inicial, o sistema foi construído como uma aplicação única em que todas as responsabilidades estavam concentradas em um mesmo serviço.

O frontend foi implementado em Streamlit [54], permitindo ao usuário enviar uma imagem histopatológica por meio de um componente de upload. A imagem recebida era processada pelo MUSK para geração de um embedding vetorial e, em seguida, consultava-se uma vector store implementada em ChromaDB, acessada por meio do framework LangChain. A recuperação de casos semelhantes baseava-se na busca por vizinhos mais próximos no espaço vetorial, retornando as imagens mais similares e seus respectivos metadados. Filtros adicionais sobre metadados (por exemplo, restrições relacionadas ao tipo de tecido ou a características clínicas) eram aplicados diretamente na camada de interface.

Essa arquitetura monolítica mostrou-se adequada para a validação inicial do pipeline técnico (desde o envio da imagem até a obtenção de casos semelhantes), mas apresentou limitações importantes do ponto de vista de escalabilidade e manutenção. Em um cenário real, com múltiplos usuários acessando o sistema de forma simultânea, a concentração de responsabilidades em um único serviço aumentaria o risco de falhas e dificultaria a evolução do sistema para contemplar novos fluxos de interação ou integrações com outros sistemas.

Além disso, a organização do frontend também se mostrava um ponto crítico de rigidez, pois suas funcionalidades estavam diretamente acopladas ao comportamento do modelo e ao fluxo interno da aplicação. Dessa forma, qualquer adição de capacidades, como incorporar novas ferramentas, alterar o fluxo de consulta ou integrar fontes adicionais de dados, exigiria refatorações sucessivas na interface e na estrutura das chamadas, tornando a evolução do sistema trabalhosa e pouco sustentável. Mesmo uma eventual migração para uma arquitetura baseada exclusivamente em APIs não eliminaria esse acoplamento, já que o frontend continuaria dependente de ajustes manuais a cada modificação no backend.

D. Evolução para uma arquitetura orientada a agente

Diante das limitações observadas na prova de conceito monolítica, o foco metodológico do trabalho foi redirecionado da construção de uma ferramenta de consulta para a análise de arquiteturas adequadas à integração do MUSK em um sistema mais flexível e extensível. Em particular, buscou-se uma abordagem que favorecesse a incorporação de novas fontes de dados, a integração de ferramentas complementares e a substituição ou adição de modelos multimodais distintos, preservando a estabilidade da arquitetura existente.

Nesse contexto, arquiteturas orientadas a agentes de IA foram consideradas uma alternativa promissora. Partindo do enquadramento adotado na introdução, em que um agente é entendido como um sistema que percebe o ambiente e escolhe ações em função de objetivos estabelecidos, o paradigma de agentes passa a oferecer uma forma unificadora de descrever sistemas de IA em termos de interação agente–ambiente. Essa perspectiva permite organizar soluções complexas como a composição de componentes com responsabilidades bem definidas (percepção, decisão, ação), o que é particularmente útil quando se busca integrar diferentes modelos e fontes de dados em um mesmo fluxo de apoio ao diagnóstico [6].

No domínio da saúde, esse enquadramento permite conceber um agente de IA responsável por coordenar o fluxo de processamento, acionando o MUSK para a extração de embeddings, consultando a vector store quando necessário, além de organizar os resultados obtidos de modo estruturado. Embora o sistema atual não mantenha um estado clínico completo nem explore de forma integral ciclos complexos de interação, a arquitetura orientada a agente fornece uma estrutura clara para incorporar tais capacidades no futuro.

Do ponto de vista da literatura recente sobre *agentic systems*, a solução implementada neste trabalho corresponde a um arranjo *single-agent* de baixa complexidade, no qual um único agente baseado em LLM coordena ferramentas de recuperação multimodal. Assim, utilizamos o termo "arquitetura orientada a agente" para descrever a camada de orquestração proposta, inspirada pelos avanços em sistemas *agentic*, mas sem abranger, no estágio atual, características como coordenação multi-agente, memória de longo prazo ou protocolos padronizados de orquestração (por exemplo, MCP).

Na fase seguinte do trabalho, a arquitetura foi reestruturada de modo a separar de forma mais clara as responsabilidades entre interface, orquestração e serviços de inferência, utilizando uma interface de chat generativa no frontend, um serviço intermediário para pré-processamento de mensagens multimodais e um backend responsável tanto pela interação com o MUSK quanto pela gestão do banco vetorial de imagens. Os detalhes dessa arquitetura e de seu funcionamento em tempo de execução são apresentados na seção de resultados, enquanto aqui se registra a trajetória metodológica que levou da prova de conceito monolítica à solução orientada a agente.

As principais tecnologias empregadas durante o desenvolvimento do protótipo e da arquitetura resultante encontram-se resumidas na Tabela 1.

E. Uso do conjunto de dados para teste da arquitetura

Para exercitar a arquitetura em um cenário próximo ao uso real, adotou-se o conjunto de Kather [22], baseado em lâminas FFPE de tumores gastrointestinais da coorte TCGA, pré-processadas e convertidas em patches normalizados. A partir desse repositório, patches de tecido tumoral, acompanhados dos metadados clínicos e anatomopatológicos disponibilizados na coorte, são utilizados para compor os dados de referência da aplicação.

Os patches foram submetidos ao MUSK para geração dos embeddings e armazenados em um vector store ChromaDB [60]). No cenário de uso considerado, o sistema é acionado tanto com consultas por imagem (uma nova biópsia enviada ao sistema) quanto com consultas textuais (descrições clínicas ou anatomopatológicas), e o agente retorna casos semelhantes do banco, apresentando as imagens recuperadas e seus metadados associados.

Nessa etapa, o objetivo não foi realizar uma avaliação quantitativa do desempenho do sistema, mas verificar, de

TABELA 1 - Tecnologias utilizadas.

Tecnologia	Utilização
Python	Linguagem de programação de propósito geral [55]
PyTorch	Biblioteca de *deep learning* em Python [56]
FastAPI	framework em Python para construção de *web APIs* [57]
MUSK	*Vision–language foundation model* para oncologia de precisão [20]
Google Gemini	Modelo multimodal de linguagem para texto e imagens [58]
Google AI Agent Development Kit (ADK)	Kit para desenvolvimento de agentes de *AI* com Gemini [59]
ChromaDB	Banco de dados vetorial (vector store) [60]
LangChain	framework para aplicações com modelos de linguagem e *RAG* [61]
Streamlit	Biblioteca para criação de interfaces *web* em Python [62]
React	Biblioteca JavaScript para interfaces de usuário [63]
Next.js	framework *full-stack* em React para aplicações *web* [64]
CopilotKit	Conjunto de componentes para integrar agentes de *AI* em aplicações *web* [65]
TypeScript	Superconjunto tipado de JavaScript [66]
Pillow (PIL)	Biblioteca para processamento de imagens em Python [67]
Git	Sistema de controle de versão distribuído [68]

FONTE: **Os autores (2025).**

forma qualitativa, se a arquitetura implementada seria capaz de percorrer o fluxo completo de processamento, desde a recepção da imagem até a recuperação de casos similares no conjunto de dados, utilizando o MUSK como componente central de extração de características.

IV. RESULTADOS

Esta seção apresenta a arquitetura implementada, os fluxos de busca multimodal e a interface desenvolvida para interação com o sistema. O objetivo é descrever, de forma objetiva, o comportamento operacional da arquitetura e os componentes que a compõem.

A. Arquitetura Implementada em Camadas

A arquitetura do sistema foi organizada em camadas lógicas, como mostra a Figura 1, de modo a separar responsabilidades de interface, processamento e recuperação de conhecimento. A camada de interface é composta por uma aplicação web em Next.js, responsável por receber comandos em linguagem natural, arquivos de imagem e interações de navegação nos resultados. Essa interface de chat permite ao patologista enviar queries textuais, anexar patches de lâminas histopatológicas e inspecionar os casos retornados.

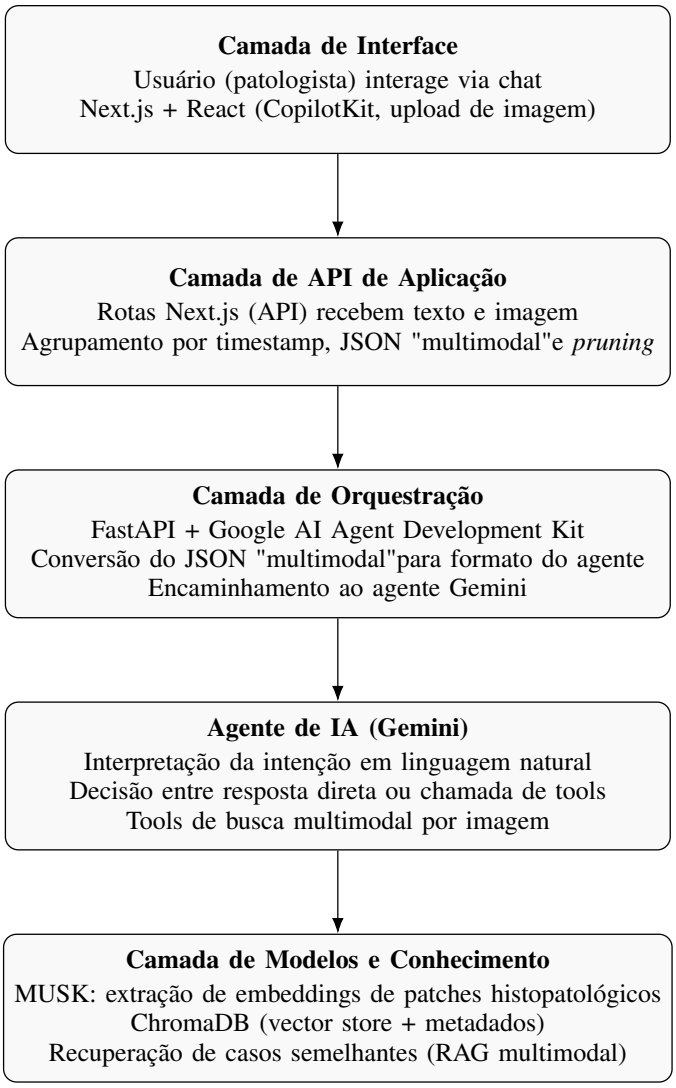

Figura 1: Arquitetura em camadas do sistema proposto. O usuário interage por meio de uma interface de chat web; a API da aplicação consolida mensagens multimodais e as envia ao servidor de orquestração, que repassa o conteúdo ao agente de IA. O agente decide quando acionar a ferramenta de busca multimodal baseada no MUSK e na ChromaDB, retornando casos semelhantes para apoio ao diagnóstico. **Fonte: Os autores (2025)**

Complementarmente, a camada de *API* de aplicação, im-

plementada nas próprias rotas do Next.js, realiza o pré-processamento das requisições. Nessa etapa, texto e imagem são agregados em estruturas JSON multimodais, com agrupamento por timestamp e remoção de partes desnecessárias, produzindo uma representação compacta adequada para envio ao servidor de orquestração.

A camada de orquestração é implementada em um serviço FastAPI integrado ao Google AI Agent Development Kit (ADK), que expõe um agente especializado em histopatologia. Esse serviço converte o JSON "multimodal"recebido da aplicação web para o formato esperado pelo agente e encaminha as requisições ao modelo Gemini, utilizado como LLM de controle.

Em sequência, no Agente de IA são administradas duas ferramentas principais: uma para busca imagem–imagem e outra para busca texto–imagem. Ambas encapsulam o pipeline de recuperação, recebendo a consulta (imagem ou descrição textual), extraindo o embedding da query correspondente e consultando o banco vetorial. Dessa forma, as tools traduzem decisões de alto nível do agente em chamadas concretas ao modelo de visão e ao repositório de conhecimento.

Finalmente, a camada de modelos e conhecimento integra o modelo MUSK, responsável pela extração de embeddings de lâminas histopatológicas, e um vector store baseado em ChromaDB, no qual as imagens do conjunto de dados são indexadas juntamente com seus metadados clínico-patológicos. A combinação MUSK + ChromaDB calcula as similaridades no espaço vetorial e retorna as top-n correspondências enriquecidas com os metadados.

Em conjunto, essas camadas permitem que o usuário interaja apenas com a interface web, enquanto API, servidor de orquestração, agente, tools e vector store permanecem encapsulados.

B. Fluxos de Busca Multimodal

O sistema admite dois fluxos principais de consulta: imagem–imagem e texto–imagem. Em ambos os casos, o patologista emite uma query que é encaminhada ao agente, o qual seleciona a tool adequada e aciona o pipeline de recuperação ilustrado na Figura 2. A região superior da figura resume essa interação entre usuário, agente e tools; a região inferior detalha o processamento da consulta no espaço vetorial.

A recuperação propriamente dita é realizada pelas tools de busca multimodal, que acionam o modelo MUSK (`musk_large_patch16_384`) e o banco vetorial construído com ChromaDB. No modo imagem–imagem, um patch de WSI enviado na requisição é convertido para RGB, redimensionado, normalizado e codificado pelo encoder visual do MUSK em um vetor de dimensão 768. No modo texto–imagem, a descrição em linguagem natural é tokenizada com XLM-RoBERTa e codificada pelo encoder de linguagem do MUSK no mesmo espaço vetorial. Em ambos os fluxos, a tool apenas orquestra esse processamento, repassando a entrada ao MUSK e utilizando o embedding resultante como vetor de consulta.

O vetor de consulta produzido pelo MUSK é então comparado aos embeddings previamente indexados no ChromaDB, obtidos a partir de patches do conjunto de dados de referência. A busca retorna as top-n correspondências, acompanhadas da distância no espaço vetorial e dos metadados clínico-patológicos associados a cada lâmina.

Quando a consulta textual inclui restrições clínicas, a mensagem em linguagem natural é analisada pelo agente, que, com apoio do modelo de linguagem, infere atributos como gênero, faixa etária, localização primária do tumor, tipo de tecido, tipo de amostra e estadiamento patológico (incluindo AJCC T, N e M) e os repassa como argumentos para a tool de busca texto–imagem. Dentro da tool, esses parâmetros são normalizados e mapeados para os campos de metadados armazenados, sendo utilizados na filtragem e ordenação dos resultados, de modo a priorizar casos que atendam às restrições impostas. O sistema também admite consultas nas quais o usuário solicita explicitamente a desativação dos filtros, permitindo buscas puramente textuais baseadas apenas na proximidade, no espaço de embeddings do MUSK, entre a descrição fornecida e as imagens indexadas.

C. Interface Conversacional e Visualização dos Resultados

A interface foi implementada com CopilotKit e integra-se ao estado do agente para refletir resultados em tempo real. Os casos recuperados são apresentados em cartões contendo a imagem do patch, sua posição no ranking de proximidade e os respectivos metadados demográficos e clínico-patológicos, conforme ilustrado na Figura 3. A partir da galeria dos resultados, é possível ter uma visualização detalhada de cada caso, que apresenta metadados estruturados por categoria.

O painel conversacional permanece ativo durante todo o uso, permitindo consultas sucessivas, ajustes de filtros e exploração dos resultados. A sincronização entre o painel e a galeria ocorre via estado compartilhado, garantindo atualização imediata após cada consulta.

D. Síntese dos Resultados

Os testes realizados demonstraram operação estável do protótipo em consultas por texto, imagem ou combinações multimodais. A separação entre unificação multimodal na aplicação web, orquestração pelo agente e extração vetorial pelo MUSK resultou em uma arquitetura modular e transparente. O agente forneceu um mecanismo explícito para controle dos critérios de recuperação e aplicação de filtros clínicos, contribuindo para a interpretabilidade do fluxo.

O sistema recuperou casos histológicos semelhantes com base em embeddings gerados pelo MUSK, apresentando um ranking de proximidade acompanhado de metadados clínicos relevantes. Os resultados obtidos confirmam a viabilidade da arquitetura proposta como ferramenta interativa para recuperação de casos em patologia digital e indicam potencial para expansão futura em cenários de apoio à decisão diagnóstica.

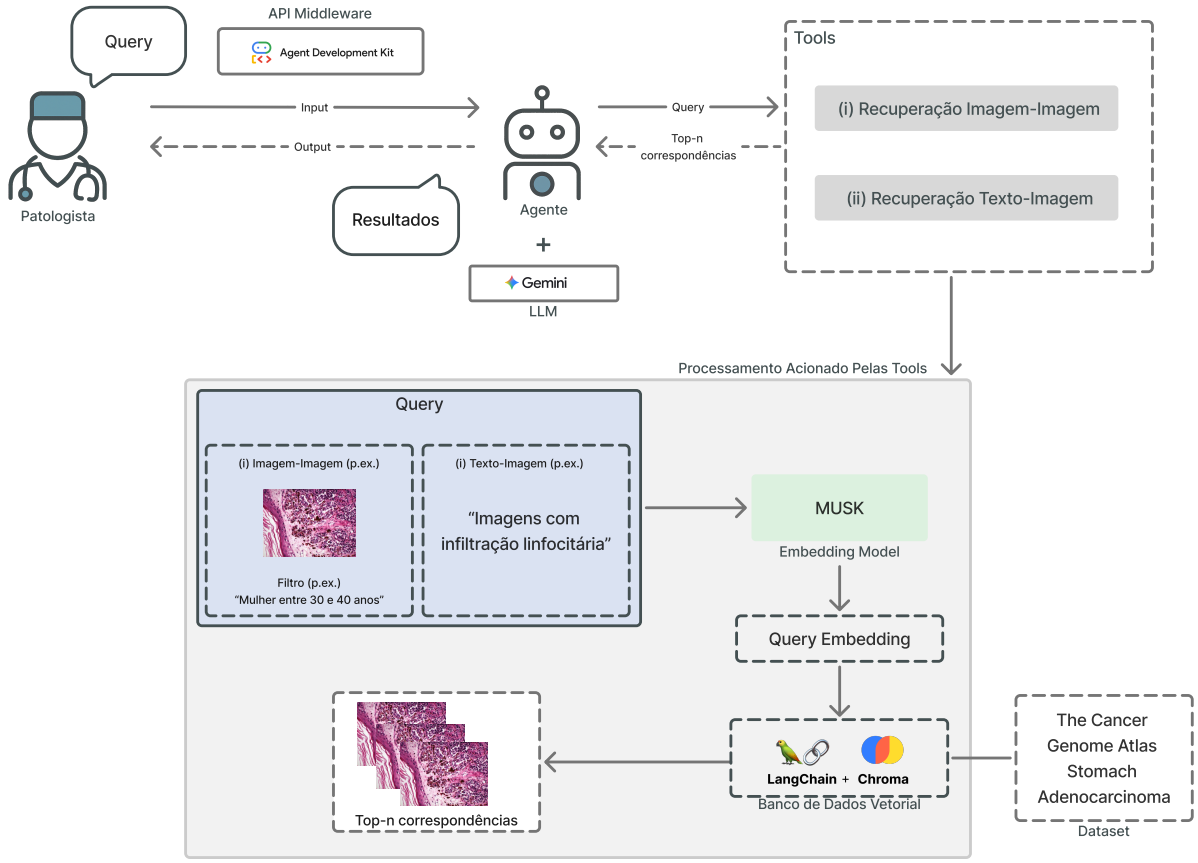

Figura 2: Fluxo operacional das buscas multimodais. A query emitida pelo patologista (imagem ou texto, possivelmente acompanhada de filtros clínicos) é encaminhada ao agente, que aciona as tools de recuperação imagem–imagem ou texto–imagem. Essas ferramentas disparam o processamento no modelo MUSK, que gera o embedding da consulta, utilizado para buscar no banco vetorial construído com Chroma as top-n correspondências e seus metadados associados. **Fonte: Os autores (2025).**

V. CONSIDERAÇÕES FINAIS

Este trabalho abordou o problema de integrar, de forma estruturada e transparente, um *vision–language foundation model* ao fluxo de apoio ao diagnóstico em patologia digital oncológica. Para isso, foram revisados conceitos de *big data*, inteligência artificial, *foundation models* e arquiteturas orientadas a agentes, com ênfase em abordagens multimodais aplicadas à patologia digital. A partir dessa base conceitual, o modelo MUSK foi selecionado como núcleo de extração multimodal e operacionalizado em ambiente local, inicialmente em uma prova de conceito monolítica e, posteriormente, em uma arquitetura orientada a agente.

Os resultados obtidos demonstram que a arquitetura proposta é tecnicamente viável e capaz de integrar o MUSK a um fluxo de recuperação de casos e organização de resultados. O agente desenvolvido recebe mensagens multimodais, aciona o VLM para geração de embeddings, consulta um vector store de imagens histológicas e organiza a resposta em um formato adequado à interação com o usuário. A separação entre interface, orquestração e serviços especializados de inferência e busca vetorial favorece a modularidade e permite a incorporação de novas ferramentas sem reestruturação da arquitetura principal. Esse arranjo também facilita a introdução de modelos especializados além do MUSK e a comparação entre diferentes pipelines de RAG.

Do ponto de vista conceitual, o estudo articula *foundation models* multimodais, RAG e arquiteturas orientadas a agentes em um cenário concreto de patologia digital, explicitando o encadeamento entre percepção (recepção de dados), decisão (seleção de ferramentas e consultas ao banco vetorial) e ação (apresentação estruturada de casos semelhantes). Do ponto de vista prático, o protótipo apresentado demonstra um caminho

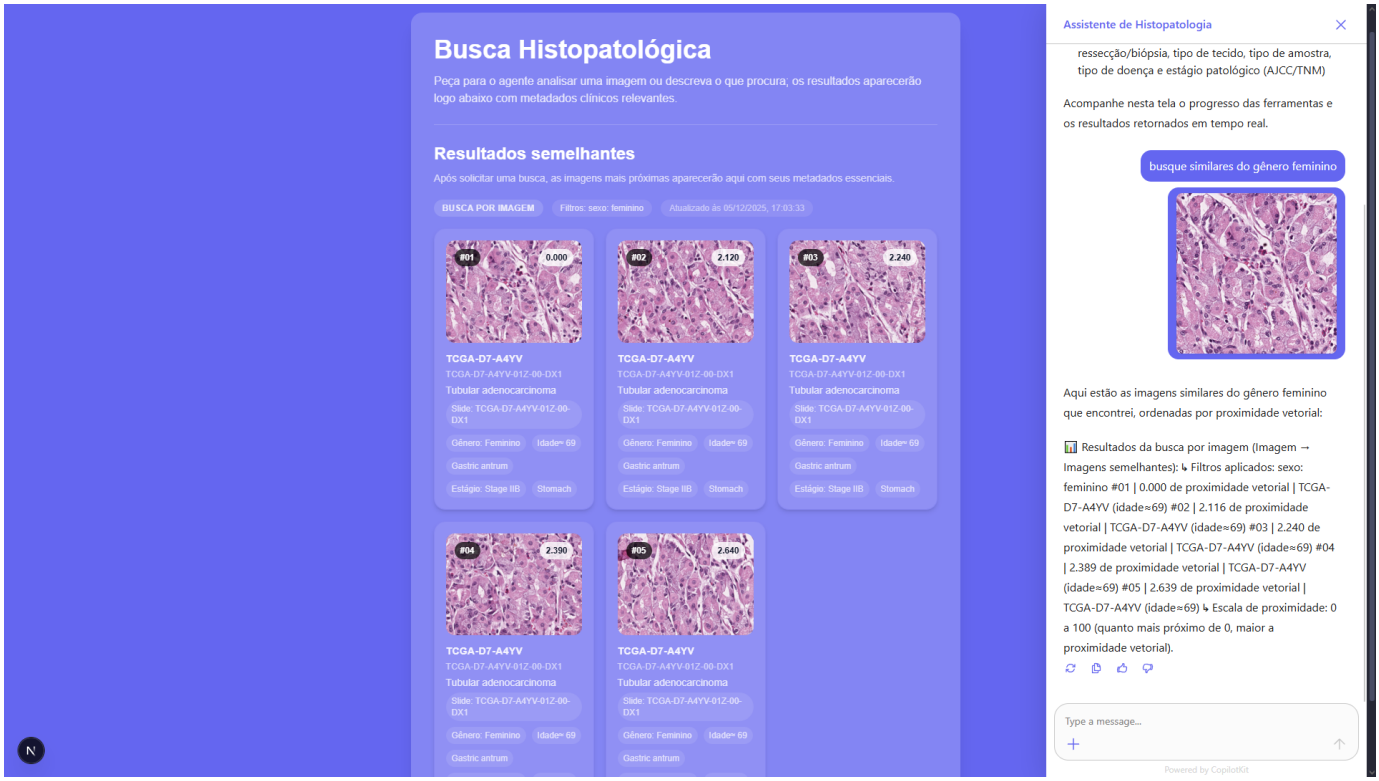

Figura 3: Interface da ferramenta com galeria de resultados. **Fonte: Os autores (2025).**

de integração de modelos no estado da arte à sistemas de apoio ao diagnóstico utilizando componentes amplamente disponíveis no ecossistema atual de IA.

O estudo apresenta, contudo, limitações importantes. A avaliação realizada teve caráter predominantemente qualitativo, concentrando-se na validação do fluxo operacional, sem análise quantitativa de desempenho diagnóstico ou comparação sistemática com modelos alternativos. O uso de um único conjunto de dados [22] restringe a generalização imediata dos resultados. A arquitetura implementada, por razões de escopo, adota um arranjo *single-agent* e não contempla mecanismos avançados de governança, auditoria ou avaliação de *usability* em ambientes clínicos reais. Além disso, parte significativa do esforço foi dedicada a desafios de engenharia decorrentes da maturidade ainda em desenvolvimento do ecossistema associado ao MUSK.

Apesar dessas limitações, o trabalho atinge seu objetivo ao demonstrar a viabilidade técnica de uma arquitetura orientada a agente para integração de um *vision–language foundation model* ao processo de apoio ao diagnóstico em patologia digital. Embora o sistema não implemente integralmente as características descritas na literatura de *agentic systems*, ele materializa um arranjo *single-agent* que antecipa parte dessas capacidades em um cenário concreto de patologia digital. A organização adotada estabelece uma base sólida para extensões futuras, incluindo arquiteturas multi-agente, incorporação de novos modelos multimodais e mecanismos padronizados de orquestração, posicionando a solução como um passo inicial em direção a sistemas de apoio à decisão mais escaláveis, transparentes e alinhados às tendências contemporâneas de IA multimodal aplicada à saúde.

A. *Recomendações para trabalhos futuros*

A partir dos resultados obtidos, diversas direções de pesquisa emergem. Em primeiro lugar, recomenda-se a realização de avaliações quantitativas sistemáticas do desempenho do sistema em tarefas de classificação e estratificação de risco, utilizando métricas consolidadas em patologia digital. A mesma infraestrutura de agente permite comparar o MUSK com outros *vision–language models* biomédicos e explorar variantes do pipeline RAG, possibilitando análises controladas sobre o impacto da substituição ou combinação de modelos.

Em segundo lugar, a arquitetura pode ser estendida para um arranjo multi-agente no qual diferentes agentes assumem funções especializadas, como recuperação multimodal, raciocínio clínico, geração de explicações e monitoramento de segurança. Protocolos de orquestração, como o Model Context Protocol (MCP), constituem um caminho promissor para padronizar a interação entre esses componentes e favorecer a evolução independente dos serviços.

Outra direção relevante envolve a exploração das capacidades avançadas do MUSK, que excedem o escopo da presente implementação. Como descrito em seu estudo original, o MUSK foi projetado como um *vision–language foundation model* capaz de desempenhar múltiplas tarefas: *visual question answering* (VQA) sobre regiões específicas de lâminas, classificação em regime *zero-shot*, recuperação imagem–texto

e texto–imagem, predição de biomarcadores moleculares, e estimativas prognósticas, incluindo risco de recidiva e resposta terapêutica em diferentes tipos de câncer. A incorporação dessas funcionalidades ao agente possibilitaria ampliar substancialmente o escopo do sistema, permitindo que ele responda perguntas clínicas de maior complexidade, realize classificações diagnósticas preliminares sem fine-tuning e produza inferências prognósticas ancoradas no modelo multimodal.

Por fim, estudos futuros podem investigar a extensão da arquitetura a outros domínios da patologia digital e imagens médicas, bem como sua integração com sistemas clínicos existentes. Avaliações envolvendo profissionais de saúde, com foco em *usability*, confiabilidade percebida e compatibilidade com fluxos de trabalho assistenciais, são essenciais para aproximar a solução de cenários reais de uso.

VI. Artefatos

Os artefatos de modelagem e código-fonte desenvolvidos estão disponíveis no acervo da Biblioteca da UFPR: `https://bibliotecas.ufpr.br/`.

Referências

[1] RAJA, R.; MUKHERJEE, I.; SARKAR, B. K. A systematic review of healthcare big data. *Scientific Programming*, v. 2020, p. 5471849, 2020.

[2] RANI, S. et al. Machine learning-powered smart healthcare systems in the era of big data: Applications, diagnostic insights, challenges, and ethical implications. *Diagnostics*, v. 15, n. 15, p. 1914, 2025.

[3] SWEENEY, S. et al. Challenges to using big data in cancer. *Cancer Research*, v. 83, p. 1175 – 1182, 2023.

[4] ZHOU, N.; CAO, J. Lr-cobras: A logic reasoning-driven interactive medical image data annotation algorithm. *Computerized Medical Imaging and Graphics*, v. 124, p. 102623, 2025. ISSN 0895-6111. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0895611125001326>.

[5] AROWOOGUN, J. O. et al. A comprehensive review of data analytics in healthcare management: Leveraging big data for decision-making. *World Journal of Advanced Research and Reviews*, v. 21, n. 2, p. 1810–1821, 2024.

[6] RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 3. ed. Upper Saddle River, NJ: Prentice Hall, 2010.

[7] RUBINGER, L. et al. Machine learning and artificial intelligence in research and healthcare. *Injury*, v. 54, n. Suppl 3, p. S69–S73, 2023.

[8] MAHESH, N. et al. Advancing healthcare: The role and impact of AI and foundation models. *American Journal of Translational Research*, v. 16, n. 6, p. 2166–2179, 2024.

[9] CHOW, J. C. L.; WONG, V.; LI, K. Generative pre-trained transformer-empowered healthcare conversations: Current trends, challenges, and future directions in large language model-enabled medical chatbots. *BioMedInformatics*, v. 4, n. 1, p. 837–852, 2024. ISSN 2673-7426. Disponível em: <https://www.mdpi.com/2673-7426/4/1/47>.

[10] SUN, Y.; ORTIZ, J. Rapid review of generative AI in smart medical applications. *ACM Transactions on Cyber-Physical Systems*, v. 1, n. 1, p. 1–13, 2024. Doi: 10.1145/nnnnnnn.nnnnnnn.

[11] HE, Y. et al. Foundation model for advancing healthcare: Challenges, opportunities and future directions. *IEEE Reviews in Biomedical Engineering*, v. 18, p. 172–191, 2025.

[12] MEDETALIBEYOGLU, A. et al. Foundational artificial intelligence models and modern medical practice. *BJR|Artificial Intelligence*, v. 2, n. 1, p. ubae018, 2025.

[13] HUANG, K. A.; CHOUDHARY, H. K.; KUO, P. C. Artificial intelligent agent architecture and clinical decision-making in the healthcare sector. *Cureus*, v. 16, n. 7, p. e64115, 2024.

[14] MASTERMAN, T. et al. The landscape of emerging AI agent architectures for reasoning, planning, and tool calling: A survey. *arXiv preprint arXiv:2404.11584*, 2024.

[15] ACHARYA, D.; KUPPAN, K.; BHASKARACHARYA, D. Agentic ai: Autonomous intelligence for complex goals—a comprehensive survey. *IEEE Access*, v. 13, p. 18912–18936, 2025. Disponível em: <https://consensus.app/papers/agentic-ai-autonomous-intelligence-for-complex-goals

[16] GARG, V. Designing the mind: How agentic frameworks are shaping the future of AI behavior. *Journal of Computer Science and Technology Studies*, 2025.

[17] SINGH, A. et al. Agentic retrieval-augmented generation: A survey on agentic rag. *ArXiv*, abs/2501.09136, 2025.

[18] BANDI, A. et al. The rise of agentic ai: A review of definitions, frameworks, architectures, applications, evaluation metrics, and challenges. *Future Internet*, MDPI, v. 17, n. 404, 2025.

[19] NISA, U. et al. Agentic ai: The age of reasoning—a review. *Journal of Automation and Intelligence*, Elsevier, 2025.

[20] XIANG, J. et al. A vision–language foundation model for precision oncology. *Nature*, v. 638, n. 8051, p. 769–778, 2025.

[21] Li Lab, Stanford University. *MUSK: Multimodal Transformer with Unified Masked Modeling*. 2025. https://github.com/lilab-stanford/MUSK. Accessed: 30 Nov. 2025.

[22] KATHER, J. N. *Histological images for MSI vs. MSS classification in gastrointestinal cancer, FFPE samples*. Zenodo, 2019. Disponível em: <https://doi.org/10.5281/zenodo.2530835>.

[23] ALSAAD, M. et al. Multimodal data integration in clinical decision support systems: A review. *Frontiers in Public Health*, v. 12, 2024.

[24] SCHNEIDER, S. et al. Deep learning in cancer prognosis: A systematic review of multimodal integration. *npj Digital Medicine*, v. 4, n. 1, p. 1–13, 2021.

[25] LIPKOVÁ, M. et al. Multimodal deep learning for cancer prognosis: Current state and future directions. *Briefings in Bioinformatics*, v. 23, n. 2, p. 1–15, 2022.

[26] AHMAD, Z. et al. Digital pathology: Current status and future perspectives. *Frontiers in Medicine*, v. 12, 2025.

[27] GAO, Y. et al. Multimodal learning in biomedical applications: A review. *Frontiers in Physiology*, v. 15, 2024.

[28] XU, Y. et al. Cross-modal retrieval in digital pathology: Aligning histopathology images and text reports. *Medical Image Analysis*, v. 89, 2024.

[29] HU, J. et al. Multimodal retrieval for histopathology: Methods and applications. *IEEE Transactions on Medical Imaging*, v. 43, n. 2, p. 1–12, 2024.

[30] LU, L. et al. Challenges and opportunities in digital pathology: A review. *IEEE Reviews in Biomedical Engineering*, v. 13, p. 1–15, 2020.

[31] CHEN, Y. et al. Handcrafted features vs. deep learning in digital pathology: A comparative study. *Computers in Biology and Medicine*, v. 144, 2022.

[32] DANG, L. et al. Scalability challenges in whole-slide image analysis. *IEEE Journal of Biomedical and Health Informatics*, v. 29, n. 1, p. 1–10, 2025.

[33] TORAB-MIANDOAB, S. et al. Interoperability in digital pathology: Standards and challenges. *Journal of Pathology Informatics*, v. 14, 2023.

[34] ZHANG, Y. et al. Clip: Connecting vision and language for multimodal learning. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. [S.l.: s.n.], 2023. p. 1–10.

[35] GAO, Y. et al. Blip: Bootstrapped language-image pretraining for unified vision-language understanding. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2021. v. 34, p. 1–12.

[36] YIN, Y. et al. A survey on multimodal foundation models in biomedicine. *IEEE Transactions on Medical Imaging*, v. 42, n. 3, p. 1–15, 2023.

[37] SHAHIN, S. et al. Biomedical vision-language models: Applications and challenges. *Frontiers in Medicine*, v. 12, 2025.

[38] SHI, Y. et al. Eyeclip: Vision-language pretraining for ophthalmology. *Medical Image Analysis*, v. 92, 2025.

[39] LIU, P. et al. Hm-rag: Hierarchical multi-agent multimodal retrieval augmented generation. *ArXiv*, abs/2504.12330, 2025.

[40] KRISHNAN, N. Advancing multi-agent systems through model context protocol: Architecture, implementation, and applications. *ArXiv*, abs/2504.21030, 2025.

[41] ABOOTORABI, M. et al. Ask in any modality: A comprehensive survey on multimodal retrieval-augmented generation. *ArXiv*, abs/2502.08826, p. 16776–16809, 2025.

[42] MEI, L. et al. A survey of multimodal retrieval-augmented generation. *ArXiv*, abs/2504.08748, 2025.

[43] ZHAO, P. et al. Retrieval-augmented generation for ai-generated content: A survey. *ArXiv*, abs/2402.19473, 2024.
[44] KIRAN, S. Hybrid retrieval-augmented generation (rag) systems with embedding vector databases. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, 2025.
[45] MORTAHEB, M. et al. Re-ranking the context for multimodal retrieval augmented generation. *ArXiv*, abs/2501.04695, 2025.
[46] LI, Y. et al. Towards agentic rag with deep reasoning: A survey of rag-reasoning systems in llms. *ArXiv*, abs/2507.09477, 2025.
[47] LI, Q.; XIE, Y. From glue-code to protocols: A critical analysis of A2A and MCP integration for scalable agent systems. *ArXiv*, abs/2505.03864, 2025.
[48] EHTESHAM, A. et al. A survey of agent interoperability protocols: Model context protocol (mcp), agent communication protocol (acp), agent-to-agent protocol (a2a), and agent network protocol (anp). *ArXiv*, abs/2505.02279, 2025.
[49] JEONG, C. A study on the MCP x A2A framework for enhancing interoperability of LLM-based autonomous agents. *ArXiv*, abs/2506.01804, 2025.
[50] FEI, X.; ZHENG, X.; FENG, H. MCP-zero: Active tool discovery for autonomous LLM agents. *ArXiv*, abs/2506.01056, 2025.
[51] PATIL, M.; LOKHANDE, V. Model context protocol (mcp): Enabling scalable ai data integration. *International Journal For Multidisciplinary Research*, 2025.
[52] CHOPPA, N.; KNIPP, M. The future of seamless generative ai and tool integration: Exploring the model context protocol. *World Journal of Advanced Engineering Technology and Sciences*, 2025.
[53] CONNEAU, A. et al. Unsupervised cross-lingual representation learning at scale. *arXiv preprint arXiv:2001.07213*, 2020.
[54] INC., S. *Streamlit: The fastest way to build data apps*. 2025. Available at: https://streamlit.io/.
[55] Python Software Foundation. *Python Programming Language*. 2024. https://www.python.org/. Acesso em: 01 dez. 2025.
[56] PyTorch. *PyTorch*. 2019. https://pytorch.org/. Acesso em: 01 dez. 2025.
[57] FastAPI. *FastAPI – High performance, easy to learn, fast to code, ready for production*. 2024. https://fastapi.tiangolo.com/. Acesso em: 01 dez. 2025.
[58] Google DeepMind. *Gemini Models*. 2025. https://ai.google.dev/gemini-api. Acesso em: 01 dez. 2025.
[59] Google AI Studio. *Agents and the Google AI Agent Builder*. 2025. https://ai.google.dev/gemini-api/docs/agents. Acesso em: 01 dez. 2025.
[60] Chroma. *ChromaDB – The AI-native open-source embedding database*. 2024. https://www.trychroma.com/. Acesso em: 01 dez. 2025.
[61] LangChain. *LangChain*. 2024. https://www.langchain.com/. Acesso em: 01 dez. 2025.
[62] Streamlit. *Streamlit*. 2024. https://streamlit.io/. Acesso em: 01 dez. 2025.
[63] React. *React – A JavaScript library for building user interfaces*. 2024. https://react.dev/. Acesso em: 01 dez. 2025.
[64] Vercel. *Next.js – The React Framework for the Web*. 2024. https://nextjs.org/. Acesso em: 01 dez. 2025.
[65] CopilotKit. *CopilotKit – Open-source framework for AI copilots*. 2024. https://www.copilotkit.ai/. Acesso em: 01 dez. 2025.
[66] TypeScript Team. *TypeScript Language*. 2024. https://www.typescriptlang.org/. Acesso em: 01 dez. 2025.
[67] Pillow. *Pillow – Python Imaging Library*. 2024. https://python-pillow.org/. Acesso em: 01 dez. 2025.
[68] GIT. *Git SCM*. 2025. https://git-scm.com. Acesso em: 01 dez. 2025.