

# Liv.IA: Um laboratório para linguagem, visualização, inferência e análise integrada com inteligência artificial

Alessandro de Oliveira Borges<sup>1</sup>

**Resumo:** Este trabalho apresenta o desenvolvimento de um *middleware* especializado denominado Liv.IA (Laboratório para Linguagem, Visualização, Inferência e Análise Integrada com IA), projetado para abstrair a complexidade de integração com múltiplos provedores e modelos de inteligência artificial em organizações públicas. O projeto encontra-se em desenvolvimento com aproximadamente 80% de conclusão e busca solucionar os desafios estruturais enfrentados na implementação de soluções inovadoras no setor público, incluindo limitações de recursos humanos, restrições infraestruturais e resistência cultural a mudanças tecnológicas. A arquitetura proposta baseia-se em princípios de separação de responsabilidades, inversão de dependência e extensibilidade, oferecendo uma API unificada para funcionalidades como *embeddings* vetoriais, processamento de linguagem natural, análise visual, classificação de texto e análise contextualizada. O sistema permite que aplicações de auditoria e controle incorporem capacidades de IA sem necessidade de reimplementar integrações complexas, concentrando-se na solução de problemas específicos ao invés de complexidades tecnológicas de integração.

**Palavras-chave:** Inteligência artificial, *middleware*, inovação no setor público, auditoria governamental, integração de sistemas, api unificada

## 1. INTRODUÇÃO

### 1.1. Contexto e Motivação

A inteligência artificial generativa passou por uma transformação fundamental nos últimos anos, evoluindo de uma tecnologia restrita a laboratórios de pesquisa para uma ferramenta amplamente acessível. Essa democratização ocorreu através de dois caminhos principais: os serviços em nuvem oferecidos por grandes corporações como OpenAI, Google e Anthropic, e os modelos locais de código aberto disponibilizados por organizações como Meta, Mistral e HuggingFace.

Embora as interfaces de chat tenham se tornado a face mais visível dessa revolução tecnológica, oferecendo uma forma intuitiva de interação, o verdadeiro potencial transformador da IA generativa manifesta-se quando transcendemos a interação humano-máquina. É na integração programática do tipo B2B (*Business-to-Business*) (Bussler, 2003), onde sistemas conversam diretamente entre si através de APIs, que encontramos as possibilidades mais promissoras para transformação organizacional.

### 1.2. Objetivos

Este trabalho tem como objetivo principal apresentar o Liv.IA como uma solução de *middleware*

1. Engenheiro Civil – UFG. Especialista em Ciência da Computação – UnB. Auditor Federal de Finanças e Controle – CGU. Analista de Infraestrutura – MPOG. Software & Application Development Manager - Fracture Technologies Ltd (Reino Unido). Professor 3º Grau (UNICESP e CEFET-Go). Analista de TI – Dataprev.

(AREND, 2022) que busca resolver um dos grandes desafios da adoção de IA em organizações públicas: a complexidade técnica. O Liv.IA foi concebido para abstrair as dificuldades inerentes à integração com múltiplos provedores de inteligência artificial, oferecendo em seu lugar uma API unificada e padronizada que simplifica o acesso a diversas capacidades de IA.

Mais do que uma solução técnica, o projeto visa democratizar o acesso a tecnologias avançadas em organizações públicas, reconhecendo que muitas equipes possuem expertise em seus domínios específicos de atuação, mas podem não ter o conhecimento técnico especializado necessário para implementar soluções de IA. Ao remover essa barreira, o Liv.IA permite que os profissionais se concentrem naquilo que realmente importa: resolver os problemas específicos de suas áreas de atuação, sejam eles relacionados a auditoria, controle, análise de dados ou qualquer outro domínio governamental.

### 1.3. Questões a serem respondidas

O desenvolvimento do Liv.IA foi guiado por questões fundamentais que emergem quando consideramos a implementação de IA no setor público. A primeira e mais premente questão é como podemos simplificar a integração de múltiplas tecnologias de IA em sistemas governamentais que frequentemente operam com recursos limitados e infraestrutura legada?

Além disso, precisamos compreender quais são as barreiras técnicas e organizacionais reais que impedem a adoção mais ampla de IA no setor público. Não se trata apenas de limitações tecnológicas, mas também de questões culturais, de capacitação e de processos organizacionais que precisam ser considerados.

Por fim, uma questão crítica que permeia todo o projeto é como garantir segurança e conformidade no uso de IA para auditoria governamental, considerando a sensibilidade dos dados tratados e as exigências regulatórias específicas do setor público.

### 1.4. Desafios enfrentados

Os principais desafios enfrentados foi a baixa disponibilidade de mão de obra e de hardware adequado para execução de modelos de IA. Isso resulta em maior tempo para entrada tempestiva de solu-

ções de IA, reduz nossas oportunidades de maior produtividade e aumenta o gap entre o que temos disponível para uso de IA e o estado da arte disponível no mercado.

Dentre os desafios de ordem técnica, destacamos a estudo e identificação da solução adequada para cada tipo de funcionalidade proposta. Em muitos casos, a solução precisou ser customizada, de sorte utilizando os recursos de IA e LLM disponibilizados pela própria aplicação. No entanto, uma vez solucionada, a solução encontrada fica disponível na API, para ser reutilizada por ela mesma e pelas aplicações cliente. Foi o caso dos Classificadores Customizados, uma ferramenta de grande utilidade na auditoria contínua.

## 2. METODOLOGIA E FERRAMENTAS UTILIZADAS

### 2.1. Abordagem metodológica

O desenvolvimento do Liv.IA adotou uma abordagem metodológica que combina princípios consagrados de engenharia de software com as necessidades específicas do contexto de inovação no setor público. A base conceitual do projeto fundamenta-se no Design Orientado a Domínio (DDD) (Fowler, 2003), reconhecendo que a modelagem efetiva de um sistema complexo deve partir de uma compreensão profunda dos domínios do problema que se busca resolver.

O processo de desenvolvimento seguiu metodologias ágeis, com iterações curtas que permitiram feedback contínuo e ajustes rápidos de direção (IANO, 2023). Essa agilidade foi fundamental para navegar as incertezas inerentes a um projeto inovador, permitindo que aprendizados fossem rapidamente incorporados ao desenvolvimento.

### 2.2. Arquitetura e componentes

A arquitetura de *middleware* desenvolvida, conforme ilustrada no Diagrama 1, posiciona o Liv.IA como camada intermediária entre sistemas clientes e provedores externos de inteligência artificial. O núcleo do *middleware* disponibiliza diversas funcionalidades, via WebService REST (Microsoft Azure, 2025), tais como: *embeddings* e análise semântica (Jurafsky, 2023) (Jurafsky, 2023), processamento de linguagem natural (NLP) (Bird, 2009), análise visual e OCR, busca web (Croft, 2015), classificação e ca-

tegorização, ferramentas de análise e assistência etc. Outras funcionalidades poderão adicionadas, de forma transparente. Esta organização permite que sistemas clientes como ALICE (Analisador de Licitações, Contratos e Editais), ATENA (Ambiente para Criação de Trilhas Especializadas), e-CGU, e outros sistemas de auditoria acessem capacidades de IA através de interface unificada, independentemente da complexidade de customização e integração com provedores externos como OpenAI, Microsoft Azure, Anthropic Claude, e Google Gemini. O *middleware* também incorpora modelos locais e bases de conhecimento especializadas, proporcionando flexibilidade entre processamento local e serviços em nuvem conforme requisitos de segurança e performance específicos.

### 2.2.1. Princípios arquiteturais

O sistema é construído sobre princípios sólidos de engenharia de software:

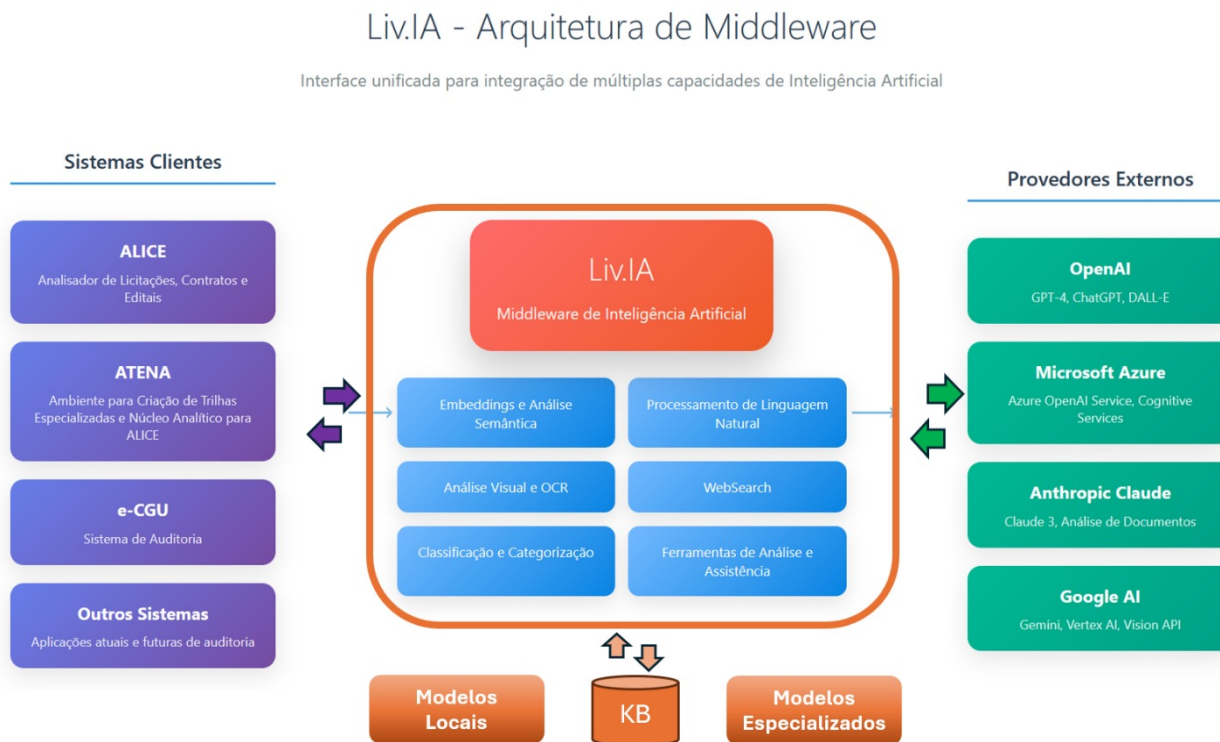
- **Separação de Responsabilidades:** Cada camada tem uma função específica e bem definida
- **Inversão de Dependência:** A lógica de negócio não depende de implementações específicas de provedores ou de acesso a recursos complexos de IA.
- **Extensibilidade:** Novos modelos e provedores podem ser adicionados sem modificar o código

existente.

- **Transparência:** O sistema pode selecionar automaticamente o melhor modelo para cada tipo de tarefa, seja para solução generativa ou funcional (NLP, NER, Reranking etc.).

A extensibilidade foi uma preocupação central desde o início do projeto. Reconhecendo a velocidade de evolução do campo de IA, o sistema foi projetado para que novos modelos e capacidades possam ser adicionados sem necessidade de modificar o código existente, seguindo o princípio Open-Closed da programação orientada a objetos.

Os componentes principais do sistema refletem essa filosofia arquitetural. A Camada de API, implementada com FastAPI (Tiangola, 2022), oferece uma interface RESTful moderna e eficiente. A Camada de Serviços encapsula toda a lógica de negócio de forma independente dos detalhes de implementação. O Registro de Modelos funciona como um catálogo centralizado que mantém metadados e configurações de todos os modelos disponíveis. O Sistema de Provedores implementa o padrão Adapter (Gamma, 1994), permitindo a conexão transparente com diferentes implementações de IA. Por fim, o Cache Inteligente otimiza tanto a performance quanto os custos operacionais, armazenando resultados de operações computacionalmente custosas.

**FIGURA 1 - VISÃO GERAL LIV.IA - MIDDLEWARE PARA INTEGRAÇÃO NOTA: KB – BASE DE CONHECIMENTO OU KNOWLEDGE BASE**

Fonte: Elaborado pelo autor, 2025.

### 2.3. Tecnologias utilizadas

A escolha das tecnologias para implementação do Liv.IA foi guiada por critérios de performance, confiabilidade e adequação ao contexto do setor público. O FastAPI emergiu como escolha natural para o framework principal, oferecendo uma combinação única de alta performance através de processamento assíncrono, documentação automática da API e suporte nativo para validação de tipos através do Pydantic. O esteio das funcionalidades de IA recaem nas tecnologias HuggingFace Transformes e Sentence Transformers (Tunstall, 2022); Scikit Learn e *llama.cpp* (Zhang, 2024), além das APIs Python OpenAI para conectividade com serviços de IA LLM compatíveis.

Para implementação de diversas funcionalidades relacionadas a classificadores de texto, operações REGEX e de buscas em bases de conhecimento, Liv.IA utiliza o SGBD PostgreSQL com extensão Pg-Vector e busca textual, em substituição a soluções restritas ou de licença limitada, como Elasticsearch e Apache SOLR.

A segurança do sistema é garantida através de autenticação baseada em api-keys, que são chaves UUID geradas para cada usuário ou projeto, permitindo controle granular de acesso e rastreabilidade completa das operações. Para processamento local de IA, quando necessário por questões de segurança ou conformidade, o sistema integra-se perfeitamente com as tecnologias *llama.cpp*, HuggingFace Transformers e Sentence Transformers permitindo execução de modelos BERT (Devlin, 2018) e LLM (Tunstall, 2022) em infraestrutura própria.

O uso da tecnologia *llama.cpp* permitiu suporte modelos BERT e LLM quantizados GGUF em 4 a 8 bits, viabilizando o uso local de modelos BERT e LLM em servidores com recursos limitados de hardware (GPU, NPU).

### 2.4. Articulações

O desenvolvimento do Liv.IA estrutura-se em uma rede colaborativa que combina recursos internos e contribuições externas. A CGPLA fornece apoio direto através da alocação de recursos hu-

manos especializados e acompanhamento técnico durante as sprints de inovação, garantindo alinhamento estratégico do projeto.

A contribuição da comunidade open source constitui elemento fundamental, materializada através de bibliotecas consolidadas como Hugging Face, Scikit Learn, Spacy, NLTK, além das tecnologias fornecidas pela Meta, Microsoft, *llama.cpp* e Ollama. Esta base tecnológica open source democratiza o acesso a capacidades avançadas de IA e representa a espinha dorsal do projeto.

A evolução da plataforma demanda novas articulações estratégicas. A colaboração com a Diretoria de Pesquisas e Informações Estratégicas (DIE) e a Diretoria de Tecnologia da Informação (DTI) mostra-se essencial para estabelecer ambiente de produção adequado e viabilizar acesso aos recursos do CGUDATA. Simultaneamente, o engajamento com a alta administração da CGU torna-se crucial para aquisição de créditos em provedores comerciais (OpenAI, Azure, Google Gemini, Anthropic Claude) e provisionamento de infraestrutura on-premise com hardware acelerador especializado - GPUs CUDA ou NPUs - necessária para processamento seguro de documentos classificados e busca de autossuficiência tecnológica.

Por fim, a interface com a LIA, através da DTI, permitirá maximizar sinergias com recursos de LLM já disponibilizados institucionalmente, otimizando investimentos em tecnologias de inteligência artificial.

### 3. RESULTADOS E IMPACTOS ESPERADOS

#### 3.1. Funcionalidades avançadas de IA

Para construir ferramentas robustas de apoio ao controle e auditoria, além do chat LLM, é necessário um conjunto abrangente de tecnologias de IA que vão muito além da simples geração de texto. O ecossistema de funcionalidades atualmente proposto na Liv.IA inclui:

##### a) Processamento de Linguagem Natural (PLN)

- **Reconhecimento de Entidades Nomeadas (NER):** Extração automática de pessoas, organizações, datas, valores monetários e outras entidades relevantes em documentos.

- **Classificação de Texto:** Categorização automática por domínio (jurídico, contábil, técnico) ou por intenção (solicitação, reclamação, elogio) ou customizada.
- **Análise de Sentimento:** Identificação de tom emocional e contexto comportamental em comunicações.

##### b) Embeddings e Análise Semântica

- Geração de representações vetoriais de textos.
- Cálculo de similaridade semântica entre documentos.
- Reordenação de resultados por relevância contextual.

##### c) WebSearch

- Capacidade de utilizar serviços de busca Web para complementação e prospecção de informações.
- Complemento ao sistema de gestão e recuperação de conhecimento (RAG).

##### d) Visão Computacional

- **Análise de Documentos Visuais:** Processamento de notas fiscais, contratos digitalizados, gráficos e tabelas
- **OCR Inteligente:** Extração de texto com compreensão contextual do layout do documento
- **Deteção de Anomalias Visuais:** Identificação de irregularidades em imagens e documentos

##### e) Análise Semântica e Recuperação de Informação

- **Embeddings Vetoriais:** Representação matemática do significado de textos para busca semântica
- **Reranking:** Reordenação de resultados por relevância contextual
- **RAG (Retrieval-Augmented Generation):** Recuperação inteligente de informações de bases de conhecimento

##### f) Análise Estrutural e Relacional

- **Extração de Documentos Complexos:** Processamento de PDFs, planilhas e documentos estruturados (*Dockling*)
- **Geração de Grafos de Conhecimento:** Mapeamento de relações e conexões entre entidades
- **Análise Contextual:** Compreensão de documentos dentro de seu contexto organizacional

### g) Classificação e Categorização

- Criação de classificadores customizados e respectivas classes.
- Classificação automática de alto desempenho, por semelhança semântica e regras REGEX.
- Análise de sentimento e intenção.
- Detecção automática de tópicos e temas.

### h) Ferramentas de Assistência e Análise

- Recursos de Assistência na síntese contextualizada de textos.
- Recursos de análise e sumarização de texto.
- Geração de código de programação a partir de especificações.
- Completamento inteligente de código ou texto.
- Documentação e explicação de texto ou código existente.

## 3.2. Benefícios esperados

Para desenvolvedores de soluções de TI, o *middleware* Liv.IA oferece uma API unificada com funcionalidades de IA prontas para uso, permitindo que as equipes do ALICE e ATENA foquem no desenvolvimento de recursos específicos de auditoria. A abstração da complexidade tecnológica é especialmente valiosa considerando a rápida evolução da IA - novos modelos são automaticamente disponibilizados sem necessidade de modificações no código dos sistemas clientes.

Para a Organização: A abordagem *middleware* proporciona redução de custos através da execução de modelos locais e cache inteligente que evita chamadas redundantes a APIs comerciais. Simultaneamente, garante maior segurança ao processar documentos sensíveis localmente e facilita a governança organizacional sobre o uso de IA.

Para Auditores: Aumento significativo de produtividade através de ferramentas especializadas que executam em minutos análises que tradicionalmente demandam horas, com maior abrangência e consistência nos resultados.

## 3.3. Lições aprendidas

O desenvolvimento do Liv.IA, mesmo ainda incompleto, já proporciona aprendizado valioso sobre a implementação de soluções de IA no contexto governamental. Entre os acertos, a decisão de focar inicialmente no suporte ao ALICE e ATENA, ao invés

de tentar criar uma solução genérica desde o início, permitiu um desenvolvimento mais direcionado e validação contínua com casos de uso reais. A arquitetura modular tem facilitado o desenvolvimento incremental, permitindo que novas funcionalidades sejam adicionadas sem comprometer o que já está funcionando.

O balanceamento entre generalização e especialização continua sendo um desafio. Enquanto queremos que o Liv.IA seja flexível o suficiente para suportar diversos casos de uso, as necessidades específicas do ALICE e ATENA podem demandar otimizações que podem não ser úteis para outros contextos. Encontrar esse equilíbrio tem sido um exercício contínuo de arquitetura e design.

## 3.4. Desdobramentos e próximos passos

Com 80% do desenvolvimento concluído, os próximos passos focam tanto na finalização das funcionalidades planejadas quanto na preparação para expansão futura. A implementação das funcionalidades restantes inclui a finalização do sistema de WebSearch integrado, essencial para que o ATENA possa complementar análises com informações públicas disponíveis na internet. O desenvolvimento de classificadores customizáveis mais sofisticados permitirá que cada órgão adapte o sistema às suas categorias e necessidades específicas.

A melhoria das ferramentas de assistência e análise, incluindo capacidades avançadas de síntese contextualizada e sumarização, está em desenvolvimento ativo. Essas funcionalidades são particularmente importantes para o ALICE, que precisa processar grandes volumes de documentos e apresentar insights acionáveis para os auditores. A expansão das capacidades de geração e análise de código também está planejada, reconhecendo que muitas análises de auditoria moderna envolvem processamento de dados e scripts customizados.

Os testes extensivos com dados reais de licitações e contratos são prioritários para validar a eficácia das funcionalidades implementadas. Esses testes não apenas verificam a correção técnica, mas também a utilidade prática dos resultados gerados. Feedback contínuo dos usuários do ALICE e ATENA está sendo incorporado para ajustar e refinar as funcionalidades.

Olhando além da conclusão do desenvolvimento inicial, vislumbramos a expansão do Liv.IA para suportar outros sistemas de auditoria e controle além do ALICE e ATENA. A criação de um ecossistema onde diferentes ferramentas de auditoria possam compartilhar capacidades de IA através do *middleware* representa uma oportunidade significativa de amplificar o impacto do projeto.

O caráter de “laboratório” do Liv.IA continuará sendo fundamental, permitindo experimentação contínua com novas tecnologias e abordagens. À medida que o campo de IA evolui, o sistema deve permanecer na vanguarda, incorporando inovações que possam beneficiar os processos de auditoria e controle governamental. A implementação de métricas mais sofisticadas para avaliar o impacto real nas atividades de auditoria, incluindo indicadores como aumento na detecção de irregularidades e redução no tempo de análise, será essencial para demonstrar valor e guiar desenvolvimentos futuros.

### 3.4.1. Exemplos de casos de uso em controle e auditoria

#### a) Análise Automática de Documentos

O Liv.IA pode processar automaticamente relatórios, atas, contratos e outros documentos, extraindo informações relevantes, identificando inconsistências e classificando conteúdo por risco ou relevância.

#### b) Análise de Conformidade

Comparação automática de documentos e processos com regulamentações e normativas, identificando possíveis não-conformidades que requerem ação corretiva.

### 3.4.2. Exemplos de caso de uso de laboratórios

#### a) Experimentação Controlada

O sistema permite que diferentes modelos ou soluções sejam testados, possibilitando:

- **Teste de Novas Funcionalidades:** Integração experimental de novos modelos e recursos sem impactar sistemas em produção
- **Comparação de Performance:** Identificação dos melhores modelos ou soluções para cada tipo de tarefa.
- **Avaliação de Custos:** Análise do custo-benefício entre diferentes soluções.

#### b) Adaptação às Necessidades Específicas

Como *middleware*, o Liv.IA pode ser customizado para atender necessidades específicas de diferentes sistemas clientes e contextos, mantendo a padronização na interface, mas permitindo especialização onde for necessário.

## 4. CONSIDERAÇÕES FINAIS E PRINCIPAIS CONTRIBUIÇÕES

O *middleware* Liv.IA representa uma abordagem sistemática para democratizar o acesso a capacidades avançadas de inteligência artificial, removendo barreiras técnicas e permitindo que equipes se concentrem na solução de problemas específicos ao invés de lidar com complexidades de integração.

Em um contexto de controle e auditoria governamental, onde a precisão, transparência e eficiência são fundamentais, ter uma plataforma unificada que combine as melhores capacidades disponíveis no mercado - sejam locais ou em nuvem - representa um diferencial estratégico significativo.

O caráter experimental e evolutivo do projeto garante que a ferramenta permanecerá relevante e eficaz conforme novas tecnologias emergem, proporcionando uma base sólida para a modernização de processos de controle através da inteligência artificial.

## REFERÊNCIAS

- AREND, D. et al. 2022. MLPro – An integrative *middleware* framework for standardized machine-learning tasks in Python. [ed.] Elsevier B.V. s.l.: Software Impacts, 2022.
- Bird, S et al. 2009. *Natural Language Processing with Python: Analyzing Text with the Natural Language Toolkit*. s.l.: O'Reilly, 2009.
- Bussler, C. 2003. *B2B Integration: Concept and Architecture*. s.l.: Springer, 2003.
- Croft, W et al. 2015. *Search Engines: Information Retrieval in Practice*. s.l.: Pearson, 2015.
- Devlin, J. et al. 2018. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. s.l.: ArXiv, 2018. 1810.04805.
- Fowler, M. 2003. *Domain-driven design*. [ed.] Addison-Wesley. Upper Saddle River: MIT Press, 2003.
- Gamma, E et al. 1994. *Design Patterns: Elements of Reusable Object-Oriented Software*. s.l.: Addison-Wesley, 1994.
- IANO, Youssef. 2023. *Engenharia de Software Moderna: Uma Abordagem para o Desenvolvimento Ágil de Software de Alta Qualidade*. s.l.: Elsevier, 2023.
- Jurafsky, D et ali. 2023. Chapter 6: Vector Semantics and Embeddings . *Speech and Language Processing: An Introduction to Natural Language Processing, Computational linguistics and Speech Recognition with Language Models*. Stanford: Stanford University, 2023.
- Microsoft Azure. 2025. *Create client for model deployed as web service*. 31/03/2025.
- Tiangola, Sebastián R. 2022. *FastAPI: The Complete Guide — Build Modern Web APIs with Python*. 2022.
- Tunstall, L. 2022. *Natural Language Processing with Transformers: Building Language Applications with Hugging Face*. s.l.: O'Reilly, 2022.
- Zhang, X. et al. 2024. *LLM in Production: Efficient Inference with llama.cpp and GGUF*. s.l.: ArXiv, 2024. 2401.09250.