

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Chrystopher Pinter Oliveira Lacerda

**Aplicação de IA Generativa para Verificação de
Conformidade de Planos de Ensino**

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Chrystopher Pinter Oliveira Lacerda

**Aplicação de IA Generativa para Verificação de
Conformidade de Planos de Ensino**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, como parte dos requi-
sitos exigidos para a obtenção título de Ba-
charel em Ciência da Computação.

Orientador: Victor Sobreira

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2025

Chrystopher Pinter Oliveira Lacerda

Aplicação de IA Generativa para Verificação de Conformidade de Planos de Ensino

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 12 de maio de 2025:

Victor Sobreira
Orientador

Claudiney Ramos Tinoco

Ilmério Reis da Silva

Uberlândia, Brasil
2025

Resumo

Este trabalho tem como objetivo avaliar a eficácia da utilização de modelos de linguagem baseados em inteligência artificial, especificamente o LLaMA 3.1 8B, na verificação de conformidade de planos de ensino com base nas diretrizes institucionais. Para isso, foi desenvolvida uma aplicação em Python capaz de extrair o conteúdo textual de documentos em formato PDF e submetê-lo ao modelo para análise, utilizando um *prompt* estruturado com base nas instruções de elaboração do plano e na ficha da disciplina. Os resultados foram comparados com análises manuais previamente realizadas por professores da instituição, evidenciando um alto grau de concordância entre as avaliações humanas e automatizadas, especialmente em seções padronizadas como identificação, ementa, objetivos, programa e bibliografia. Divergências pontuais foram observadas em aspectos que exigem maior interpretação textual, como justificativa e metodologia. Conclui-se que o modelo LLaMA demonstrou desempenho satisfatório na tarefa de verificação automatizada, com potencial para auxiliar processos de revisão pedagógica, desde que aliado à supervisão humana para lidar com subjetividades e interpretações contextuais.

Palavras-chave: Inteligência Artificial Generativa, Análise de Conformidade, RAG, LLM

Lista de ilustrações

Figura 1 – Diagrama de Pacotes do Sistema.	18
Figura 2 – Fase 1 — Entrada e Extração do Conteúdo do Plano.	19
Figura 3 – Fase 2 — Estruturação dos Prompts e Preparação dos Dados.	20
Figura 4 – Fase 3 — Análise Final e Geração do Relatório de Conformidade.	21
Figura 5 – Parâmetros de inferência modelo Llama.	26
Figura 6 – Parâmetros de carregamento modelo Llama.	27
Figura 7 – Servidor LM Studio.	28
Figura 8 – Interface da aplicação antes do recebimento dos arquivos.	29
Figura 9 – Interface da aplicação depois do processamento dos arquivos.	30
Figura 10 – Diagrama de atividades - Verificação de Conformidade.	32
Figura 11 – Requisição para a API do LM Studio.	34

Lista de tabelas

Tabela 1 – Comparativo entre os trabalhos relacionados	17
Tabela 2 – Especificações da máquina utilizada para execução local do modelo LLaMA.	23
Tabela 3 – Comparação entre análise manual e resposta do LLaMA.	42
Tabela 4 – Comparação entre análise manual e resposta do LLaMA.	45
Tabela 5 – Comparação entre análise manual e resposta do LLaMA.	46
Tabela 6 – Comparação entre análise manual e resposta do LLaMA.	48
Tabela 7 – Distribuição de convergência entre sistema e análise manual	50

Lista de abreviaturas e siglas

RAG	<i>Retrieval-Augmented Generation</i>
LLM	<i>Large Language Model</i>
IA	Inteligência Artificial
GPT	<i>Generative Pre-trained Transformer</i>
LLaMA	Large Language Model Meta AI
GDPR	<i>General Data Protection Regulation</i>
GenAI	Generative Artificial Intelligence
PLN	Processamento de Linguagem Natural
TA	Tecnologia Assistiva

Sumário

1	INTRODUÇÃO	9
1.1	Proposta	10
1.2	Estrutura do Texto	10
2	FUNDAMENTAÇÃO TEÓRICA	11
2.1	IA Generativa	11
2.2	Verificação de Conformidade	11
2.3	RAG (<i>Retrieval-Augmented Generation</i>)	12
2.4	Engenharia de <i>Prompt</i>	12
2.5	Parâmetros de Modelos de Linguagem (LLMs)	13
2.6	LM Studio	14
2.7	Hugging Face	14
3	TRABALHOS RELACIONADOS	16
4	DESENVOLVIMENTO	18
4.1	Coleta de Dados	21
4.2	Escolha do modelo de linguagem	22
4.3	Configuração do Ambiente	23
4.4	Recebimento e Processamento dos Arquivos	28
4.5	Criação do <i>Prompt</i>	31
4.6	Chamada a API do Llama	33
4.7	Repositório, Instalação e Uso do Sistema	34
4.7.1	Pré-requisitos	34
4.7.2	Instalação	35
4.7.3	Uso	35
4.7.4	Como Contribuir	36
4.8	Considerações Finais	36
5	RESULTADOS	37
5.1	Metodologia de Avaliação	37
5.1.1	Processo de Revisão Manual	37
5.1.2	Processo de Avaliação do Modelo	38
5.1.3	Comparação entre Modelo e Revisão Manual	38
5.2	Análise Comparativa entre Modelo e Revisão Manual	38
5.2.1	Primeira Análise	40

5.2.2	Segunda Análise	43
5.2.3	Terceira Análise	46
5.2.4	Análise de documento em branco	47
5.3	Análise Geral das Comparações	49
5.3.1	Grupo A – Planos Não Estruturados	49
5.3.2	Grupo B – Planos Estruturados (17 planos)	50
5.3.3	Considerações Finais das Comparações	51
5.4	Considerações Finais	51
6	CONCLUSÃO	53
	REFERÊNCIAS	55

1 Introdução

A análise de conformidade de documentos é uma atividade indispensável em diferentes contextos, como o acadêmico, jurídico e empresarial, pois assegura que documentos atendam a normas, regulamentos e requisitos específicos. Tradicionalmente realizada de forma manual, essa prática demanda tempo e atenção detalhada para evitar erros que podem comprometer processos essenciais. Nos últimos anos, avanços em tecnologias de Inteligência Artificial (IA), especialmente os *Large Language Models* (LLMs), têm transformado essa dinâmica, oferecendo ferramentas capazes de automatizar essas verificações com maior eficiência e precisão (MACEY-DARE, 2023).

A relevância do tema está atrelada à crescente complexidade e volume de informações que instituições acadêmicas e organizações enfrentam diariamente. A verificação de conformidade é essencial tanto no meio acadêmico, para validar projetos pedagógicos e garantir o cumprimento das diretrizes curriculares e legislações educacionais, quanto no setor corporativo, onde o não cumprimento de regulamentações pode acarretar penalidades, impactos financeiros e danos à reputação. Nesse contexto, a automação por meio de LLMs não apenas alivia a carga de trabalho, mas também viabiliza análises mais robustas e abrangentes, promovendo economia de tempo, maior eficiência na gestão e viabilidade de tarefas que exigiriam muitos recursos humanos, como a verificação completa de grandes volumes documentais.

O estudo conduzido por Ioannidis Joshua Harper (2023) apresenta evidências que destacam a importância e a viabilidade da utilização de inteligência artificial na verificação de conformidade em processos jurídicos. Com base nas pesquisas e aplicações desenvolvidas, os autores apontam que os modelos de IA generativa promoverão transformações significativas em diversas funções legais, incluindo a conformidade regulatória, ampliando as oportunidades para aprimorar a tomada de decisão no campo do direito, como demonstrado com as ferramentas desenvolvidas no estudo.

Por outro lado, a pesquisa de Gao et al. (2024) propõe soluções para abordar *Retrieval-Augmented Generation* (RAG), explorando, assim, métodos eficazes para o processamento de documentos por meio de LLMs. Ademais, destaca oportunidades de pesquisa para aumentar a robustez do RAG e sua capacidade de lidar com contextos extensos, o que amplia seu impacto prático e atrai crescente interesse dos setores acadêmico e industrial.

1.1 Proposta

Com tudo isso em vista, o objetivo deste trabalho é avaliar a capacidade das [LLMs](#), como o Llama 3.1, em analisar a conformidade de documentos, explorando casos de uso específicos na verificação de planos de ensino no âmbito acadêmico. Ao longo desse estudo, também foram investigadas as técnicas e desafios para implementar soluções de [IA](#) no problema proposto.

Por fim, as principais ferramentas utilizadas para desenvolver o sistema foram a linguagem de programação Python ([FOUNDATION, 2024](#)), o modelo de IA generativa Llama 3.1:8b ([META, 2024](#)) e métodos de [RAG](#) para lidar com a entrada e geração de resposta do programa. Além disso, foram utilizadas bibliotecas adicionais como “pdf-plumber” para extração de texto de documentos PDF, “requests” para realizar requisições HTTP, json para manipulação de dados estruturados em formato “JSON” e “gradio” para construção da interface de usuário.

1.2 Estrutura do Texto

O texto está organizado da seguinte forma: O Capítulo 2 discute a fundamentação teórica, abordando tópicos como IA generativa, verificação de conformidade, [RAG](#), engenharia de *prompt*, parâmetros de [LLMs](#), além de ferramentas como LM Studio e Hugging Face. O Capítulo 3 apresenta os trabalhos relacionados à temática. O Capítulo 4 descreve detalhadamente o desenvolvimento do sistema, desde a coleta de dados até a chamada ao modelo de linguagem. O Capítulo 5 trata da apresentação dos resultados, metodologia de avaliação e análises comparativas entre o desempenho do modelo e a revisão manual. Por fim, o Capítulo 6 apresenta as conclusões do trabalho, apontando limitações, contribuições e sugestões de melhorias.

2 Fundamentação Teórica

Este capítulo apresenta a base teórica necessária para sustentar o desenvolvimento do trabalho, reunindo conceitos, técnicas e ferramentas essenciais para a aplicação de inteligência artificial no contexto proposto.

2.1 IA Generativa

IA Generativa ou Generative Artificial Intelligence ([GenAI](#)) (abreviado do termo em inglês) é uma tecnologia que utiliza redes neurais profundas, como os modelos de linguagem baseados em Transformer, para criar novos conteúdos. Ela pode gerar textos, imagens, códigos-fonte e até composições de áudio que imitam padrões observados em grandes volumes de dados de treinamento. Modelos como *Generative Pre-trained Transformer* ([GPT](#)) revolucionaram o campo com sua capacidade de compreender e gerar linguagem natural de forma contextualizada, o que possibilitou sua aplicação em áreas como atendimento ao cliente, redação assistida e criação de conteúdo digital. A IA Generativa se destaca por seu potencial de automação e criação sintética em tarefas que antes exigiam exclusivamente intervenção humana, abrindo novas possibilidades no uso de tecnologias em contextos empresariais e criativos ([BROWN et al., 2020](#)).

Um exemplo notável de IA generativa é o Large Language Model Meta AI ([LLaMA](#)), desenvolvido pela Meta ([META, 2024](#)). Projetado como uma alternativa de código aberto aos grandes modelos proprietários, o LLaMA apresenta vantagens significativas. Além de sua eficiência e escalabilidade, destaca-se por permitir a execução local (*offline*), sem a necessidade de planos pagos ou dependência de servidores em nuvem. Essa característica é especialmente relevante sob as perspectivas de customização, redução de custos e, principalmente, de privacidade e proteção de dados, pois elimina a necessidade de compartilhar informações sensíveis com terceiros. Com essas qualidades, o LLaMA fortalece o ecossistema de IA generativa ao oferecer uma solução acessível e flexível para pesquisadores, desenvolvedores e organizações que buscam aplicar inteligência artificial de forma autônoma e segura em diversos contextos.

2.2 Verificação de Conformidade

A verificação de conformidade refere-se ao processo de análise e validação de documentos e processos em relação a normas e regulamentos preestabelecidos. Esse procedimento é essencial para setores que lidam com regulamentações rigorosas, como o financeiro, o jurídico e o governamental, onde a precisão e a aderência a padrões legais são

indispensáveis. Historicamente, a verificação de conformidade é realizada manualmente, o que tornava o processo demorado e suscetível a erros. No entanto, com o avanço da automação e do aprendizado de máquina, soluções de IA agora permitem que essa verificação seja feita de forma mais eficiente e com alta precisão, minimizando riscos e garantindo a integridade dos processos (CUTTING; CUTTING-DECELLE, 2021). Um exemplo de aplicação está presente no trabalho GDPR Compliant ChatGPT Playground (NAYAK et al., 2024), onde os autores criaram uma ferramenta para garantir a conformidade com as regras da *General Data Protection Regulation* (GDPR) utilizando o modelo de linguagem da OpenAI.

2.3 RAG (*Retrieval-Augmented Generation*)

RAG, ou *Retrieval-Augmented Generation*, é uma técnica avançada que combina a capacidade de geração de conteúdo dos modelos de linguagem com a recuperação de informações específicas de bancos de dados ou fontes externas. Ao integrar a recuperação de dados no processo de geração, o RAG permite que o modelo produza respostas mais precisas e contextualmente relevantes ao acessar informações atualizadas ou especializadas. O RAG é, portanto, um passo importante para superar limitações de modelos de linguagem tradicionais, que tendem a “alucinar” ou gerar respostas imprecisas quando dependem apenas de seu próprio treinamento estático (GAO et al., 2024). Visto que o RAG é eficiente em tarefas que exigem uma menor taxa de alucinação, é um conjunto de técnicas comumente utilizado em análise de documentos, onde é necessária uma alta confiabilidade nos dados apresentados, como podemos ver no trabalho Garlyal, Hariharan e Singh (2024), que aplica RAG no contexto de análise de textos jurídicos.

2.4 Engenharia de *Prompt*

A engenharia de *prompt* é o processo de elaborar, refinar e otimizar entradas de texto para orientar sistemas de GenAI a produzir saídas específicas e de alta qualidade. Nesse cenário, um *prompt* é uma instrução em linguagem natural que descreve a tarefa que a IA deve realizar, podendo variar desde perguntas diretas até descrições detalhadas que fornecem contexto e orientações específicas (IBM, 2025). O conceito de engenharia de *prompt* ganhou destaque com o avanço dos modelos de linguagem natural e da IA generativa. Em 2018, pesquisadores propuseram que diversas tarefas de processamento de linguagem natural poderiam ser abordadas como problemas de perguntas e respostas sobre um contexto, treinando um modelo multitarefa capaz de responder a diferentes tipos de questões. Com o lançamento de modelos avançados, como o ChatGPT em 2022, a engenharia de *prompt* tornou-se uma habilidade importante, permitindo que usuários e

desenvolvedores orientassem modelos de IA para gerar respostas mais relevantes e precisas (MCCANN et al., 2018).

2.5 Parâmetros de Modelos de Linguagem (LLMs)

Modelos de linguagem de grande escala (LLMs) possuem diversos parâmetros ajustáveis que influenciam diretamente na qualidade, diversidade e controle das respostas geradas. Entre os mais utilizados estão a temperatura (*temperature*), o número máximo de tokens (*max tokens*), o filtro de probabilidade acumulada (*top-p* ou *nucleus sampling*), e o limite de palavras mais prováveis consideradas (*top-k*). Esses parâmetros são essenciais para equilibrar criatividade e precisão na geração de conteúdo, especialmente em tarefas críticas como verificação de conformidade, onde coerência e aderência ao conteúdo são fundamentais.

A **temperatura** controla o grau de aleatoriedade na geração de texto. Valores mais baixos (por exemplo, 0,2) resultam em saídas mais determinísticas e conservadoras, enquanto valores mais altos (como 0,8 ou 1,0) promovem maior diversidade e criatividade nas respostas, podendo gerar conteúdos mais variados, porém menos previsíveis (HOLTZMAN et al., 2020).

O parâmetro ***max tokens*** define o número máximo de *tokens* (unidades de texto, como palavras ou partes de palavras) que podem ser gerados em uma resposta. Este parâmetro é essencial para limitar a extensão da saída, evitando respostas excessivamente longas ou cortes inesperados em respostas extensas (BROWN et al., 2020).

Já o ***top-k sampling*** restringe a geração ao conjunto das k palavras com maior probabilidade em cada etapa da geração. Por exemplo, se $k = 40$, o modelo escolherá a próxima palavra apenas entre as 40 mais prováveis. Essa técnica reduz ruídos gerados por escolhas improváveis e aumenta a coerência da resposta (FAN; LEWIS; DAUPHIN, 2018).

Adicionalmente, o ***top-p sampling*** ou *nucleus sampling*, propõe selecionar as próximas palavras com base em uma probabilidade acumulada. Por exemplo, com $p = 0.9$, o modelo considera apenas as palavras mais prováveis cuja soma de probabilidades atinge 90%. Isso proporciona maior flexibilidade do que o top-k, adaptando-se dinamicamente à distribuição de probabilidade (HOLTZMAN et al., 2020).

Esses parâmetros não apenas influenciam a geração textual, mas também impactam diretamente em aplicações que requerem confiança e precisão, como sistemas baseados em RAG. Ajustar esses valores corretamente pode reduzir significativamente alucinações e melhorar a utilidade prática das respostas geradas pelos modelos de linguagem.

2.6 LM Studio

O *LM Studio* é uma ferramenta que facilita a utilização de modelos de linguagem de grande porte [LLMs](#) diretamente no ambiente local do usuário, sem necessidade de conexão com servidores externos ou dependência de APIs de terceiros. Essa abordagem proporciona maior controle sobre a execução dos modelos, além de contribuir para a preservação da privacidade e segurança dos dados processados.

A interface do LM Studio permite carregar modelos no formato GGUF, otimizados para inferência local por meio de *backends* como *llama.cpp*. O software oferece compatibilidade com modelos como LLaMA, Mistral, Falcon, entre outros, e possui integração com GPUs e aceleração por CPU (via instruções AVX2 e AVX512, quando disponíveis) ([LM Studio, 2024](#)).

Para fins de pesquisa e desenvolvimento de aplicações baseadas em [LLMs](#), como no caso da análise automatizada de planos de ensino, o LM Studio representa uma solução acessível e eficiente. Ele permite testes com diferentes modelos de código aberto, ajustes de parâmetros de geração (como *temperature*, *top-p* e número de *tokens*) e a visualização em tempo real da resposta gerada, mesmo em ambientes com recursos limitados.

A execução local também elimina as limitações de custo e latência associadas a serviços na nuvem, permitindo experimentação mais livre e replicável em contextos acadêmicos ([TOUVRON et al., 2023](#); [COMMUNITY, 2023](#)).

2.7 Hugging Face

A Hugging Face é uma plataforma e empresa referência na área de inteligência artificial, especialmente no desenvolvimento e distribuição de modelos de linguagem natural. Criada em 2016, inicialmente como um aplicativo de *chatbot*, a empresa rapidamente passou a se dedicar à criação de ferramentas de código aberto, sendo a biblioteca *Transformers* seu principal produto ([WOLF et al., 2020](#)). Essa biblioteca se tornou um padrão no uso de modelos como BERT, GPT, T5 e LLaMA, facilitando a aplicação de técnicas avançadas de Processamento de Linguagem Natural ([PLN](#)) por meio de interfaces simples.

Atualmente, a Hugging Face mantém um ecossistema robusto que inclui o *Model Hub*, com milhares de modelos prontos para uso, o *Datasets Hub*, com conjuntos de dados compatíveis, e os *Spaces*, que permitem a publicação de aplicações interativas por meio de interfaces como Gradio e Streamlit. Além disso, a plataforma promove iniciativas colaborativas, como o projeto BigScience, responsável pelo desenvolvimento do modelo BLOOM, voltado à pesquisa aberta e transparente em [LLMs](#).

No contexto deste trabalho, a Hugging Face é essencial por fornecer os modelos de linguagem utilizados no LM Studio, ferramenta que permite rodar [LLMs](#) localmente ([LM](#)

[Studio, 2025](#)). Por meio dela, modelos como LLaMA 3.1 Instruct podem ser empregados para tarefas específicas, como a verificação automatizada da conformidade de planos de ensino.

3 Trabalhos Relacionados

O primeiro trabalho relacionado ao projeto propõe a abordagem DERECHA, que utiliza PLN para automatizar a verificação de conformidade entre Acordos de Processamento de Dados e o GDPR. Os autores, em colaboração com especialistas jurídicos, extraíram os “*shall requisites*” do GDPR e construíram uma tabela de glossário de termos jurídicos para garantir a precisão. A tecnologia PLN é utilizada para gerar representações semânticas das cláusulas do acordo e compará-las com representações predefinidas dos requisitos, identificando violações e lacunas (CEJAS et al., 2023).

O segundo trabalho apresenta o CompliAT, uma estrutura que utiliza Modelos de Linguagem de Grande Escala LLMs para verificar a conformidade das especificações de produtos de Tecnologia Assistiva (TA) com os padrões regulamentares. A principal inovação do CompliAT é a automatização da verificação de consistência terminológica entre as especificações de produtos e os padrões de conformidade, utilizando um modelo de geração aumentado por recuperação RAG. Este sistema não apenas facilita a verificação de conformidade, mas também melhora a qualidade e a segurança do produto, o que se traduz em maior eficiência no desenvolvimento de produtos de TA (ARORA et al., 2024).

O terceiro trabalho investiga o uso do LawGPT, uma IA conversacional avançada focada no Código Penal Indiano, para melhorar a prática jurídica por meio da geração de respostas precisas a consultas legais. Utilizando uma arquitetura de RAG com modelos como o *Dense Passage Retriever* e BART, o LawGPT permite consultas rápidas e precisas ao código jurídico, ampliando o acesso ao conhecimento e melhorando a eficiência na prática legal. O estudo sugere que a Inteligência Artificial Conversacional tem um grande potencial para transformar a pesquisa e o acesso à informação jurídica (GARLYAL; HARIHARAN; SINGH, 2024).

O quarto trabalho, desenvolvido por ALQAHTANI et al. (ALQAHTANI et al., 2023) indica que a Inteligência Artificial (IA) tem revolucionado diversos domínios, incluindo a educação superior, especialmente com os avanços em Processamento de Linguagem Natural (PLN) e Modelos de Linguagem Grandes LLMs, que têm ampliado nossa compreensão e aplicação da IA no campo educacional, oferecendo novas possibilidades para a personalização do ensino, avaliação automática e análise de conteúdo educacional. O estudo fornece uma visão abrangente sobre o papel emergente dessas tecnologias na educação superior e na pesquisa, discutindo suas vantagens, desafios e aplicações inovadoras, além de argumentar que essas ferramentas têm o potencial de transformar práticas educacionais e de pesquisa, levando a melhores resultados para estudantes, educadores e pesquisadores.

O quinto trabalho, proposto por [LIU; LI; ZHU](#), explora um método baseado em modelos [GPT](#) para verificação automática de conformidade utilizando engenharia de prompts. A pesquisa investiga como técnicas de engenharia de prompts podem otimizar a performance de [LLMs](#) na identificação de violações de conformidade em documentos regulamentares. O sistema proposto emprega representações semânticas geradas por [LLMs](#) para realizar comparações entre cláusulas textuais e requisitos legais, oferecendo uma abordagem flexível e extensível para ambientes regulatórios diversos ([LIU; LI; ZHU, 2023](#)).

Trabalho	Objetivo Principal	Tecnologia Utilizada	Contribuição Relevante
Cejas et al. (2023)	Verificação de conformidade entre acordos e GDPR .	PLN e representações semânticas.	Mapeamento automático de cláusulas para requisitos regulatórios.
Arora et al. (2024)	Verificação de conformidade de produtos de TA com padrões regulamentares.	LLM e RAG .	Consistência terminológica automatizada entre especificações e padrões.
Garlyal, Hariharan e Singh (2024)	Responder consultas legais com base no Código Penal Indiano.	RAG , <i>Dense Passage Retrieval</i> e BART .	Desenvolvimento de uma solução de IA jurídica conversacional com acesso eficiente a bases legais.
Alqahtani et al. (2023)	Exploração de LLMs na educação superior	PLN e LLM .	Panorama geral do uso de IA na personalização e análise educacional.
Liu, Li e Zhu (2023)	Verificação automática de conformidade com engenharia de <i>prompts</i> .	LLM (GPT) e engenharia de <i>prompts</i>	Demonstra como <i>prompts</i> e LLMs podem ser aplicados à verificação de conformidade em documentos complexos.

Tabela 1 – Comparativo entre os trabalhos relacionados

4 Desenvolvimento

Este capítulo descreve o processo de desenvolvimento de um sistema que utiliza o modelo Llama 3.1 8B, executado via LM Studio, para verificar a conformidade de planos de ensino em uma universidade. A implementação envolve a extração de texto de documentos PDF, a formatação de um *prompt* adequado e a análise das respostas geradas pelo LLM. A seguir, são apresentados dois diagramas que ilustram aspectos estruturais e funcionais do sistema desenvolvido.

O diagrama de pacotes na Figura 1 apresenta a organização modular do sistema, destacando os principais componentes envolvidos, como os responsáveis pela leitura de arquivos PDF, formatação do prompt, comunicação com o modelo de linguagem e geração de relatórios.

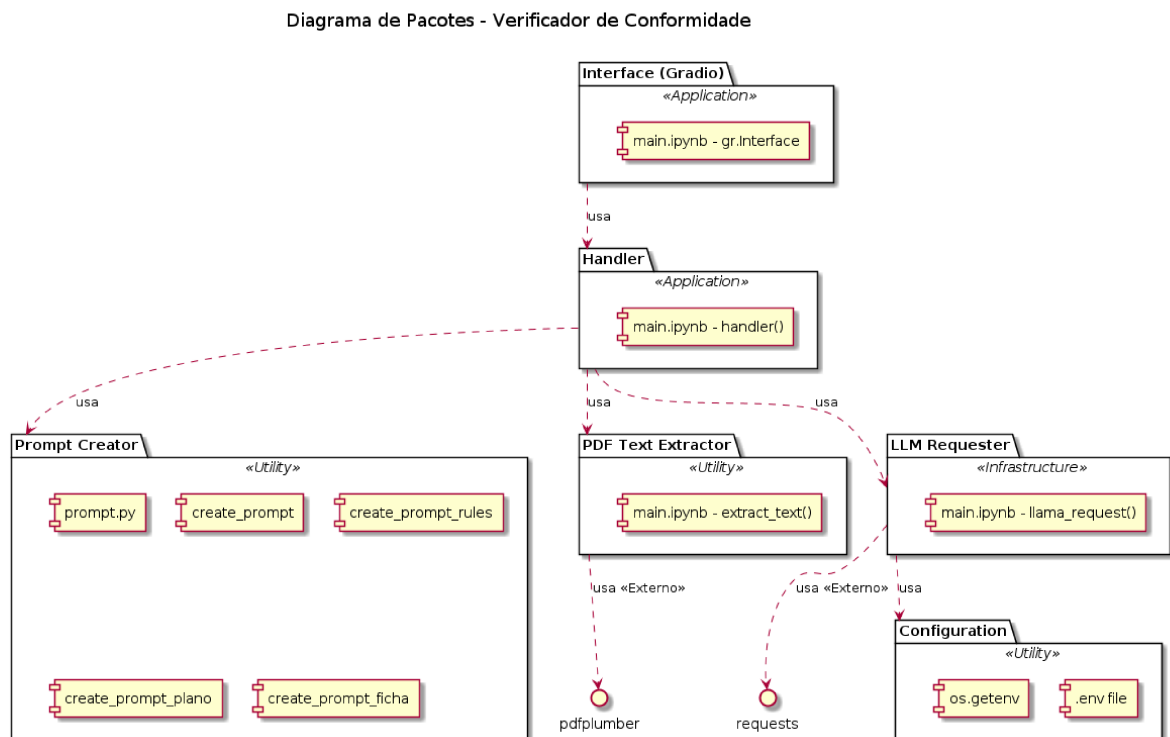


Figura 1 – Diagrama de Pacotes do Sistema.

O processo de verificação de conformidade foi representado por meio de um diagrama de atividades, dividido em três etapas principais, conforme mostrado nas Figuras 2, 3 e 4. Esse fluxo descreve, de forma sequencial, as ações realizadas desde a entrada do plano de ensino até a geração do relatório final, organizadas pelas fases de Entrada e Extração, Estruturação dos *Prompts* e Análise Final com Geração de Saída.

Diagrama de Atividades - Fase 1: Entrada e Extração de Texto

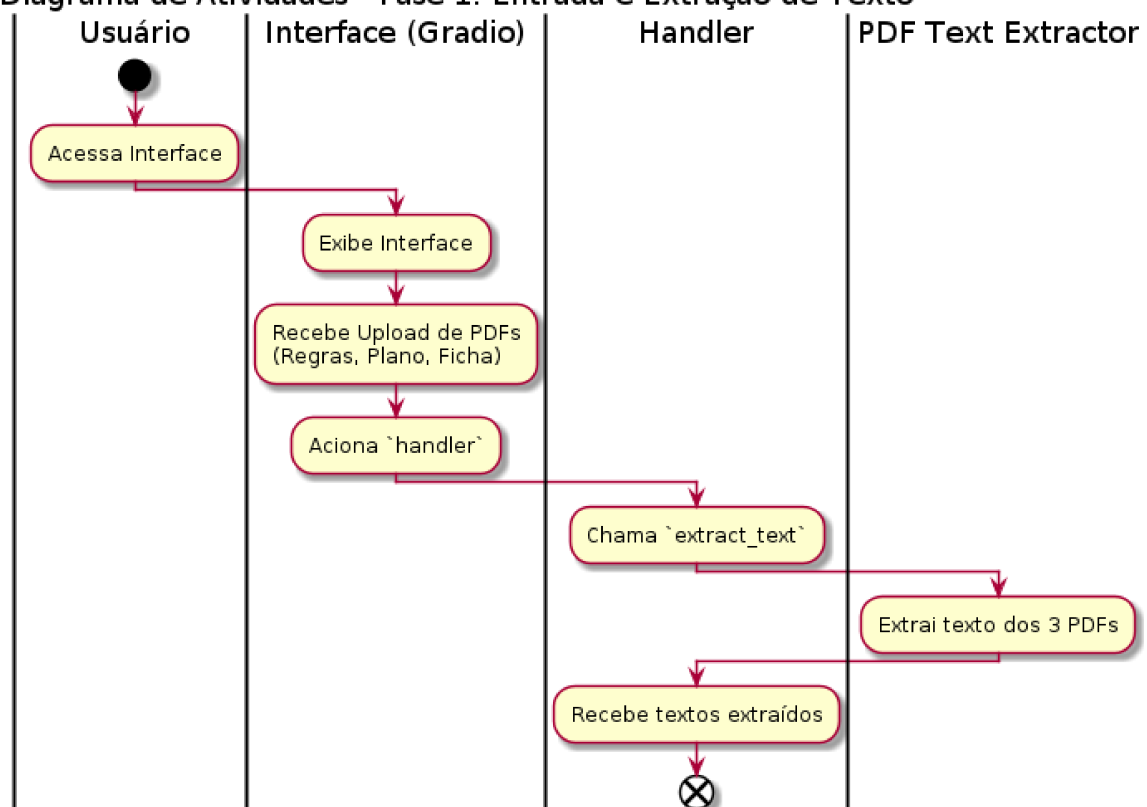


Figura 2 – Fase 1 — Entrada e Extração do Conteúdo do Plano.

O primeiro diagrama (Figura 2) ilustra o início do fluxo de interação do usuário com o sistema. O processo começa com o usuário acessando a interface web fornecida pelo Gradio. Em seguida, o usuário realiza o upload dos três documentos PDF necessários: o arquivo contendo as regras de elaboração, o plano de ensino a ser avaliado e a ficha da disciplina correspondente. A interface Gradio recebe esses arquivos e aciona a função principal ('handler') do sistema. Esta função, por sua vez, invoca o componente de extração de texto ('PDF Text Extractor'), que processa cada PDF para obter seu conteúdo textual. Finalmente, os textos extraídos dos três documentos são retornados ao 'handler', concluindo a fase inicial de preparação dos dados.

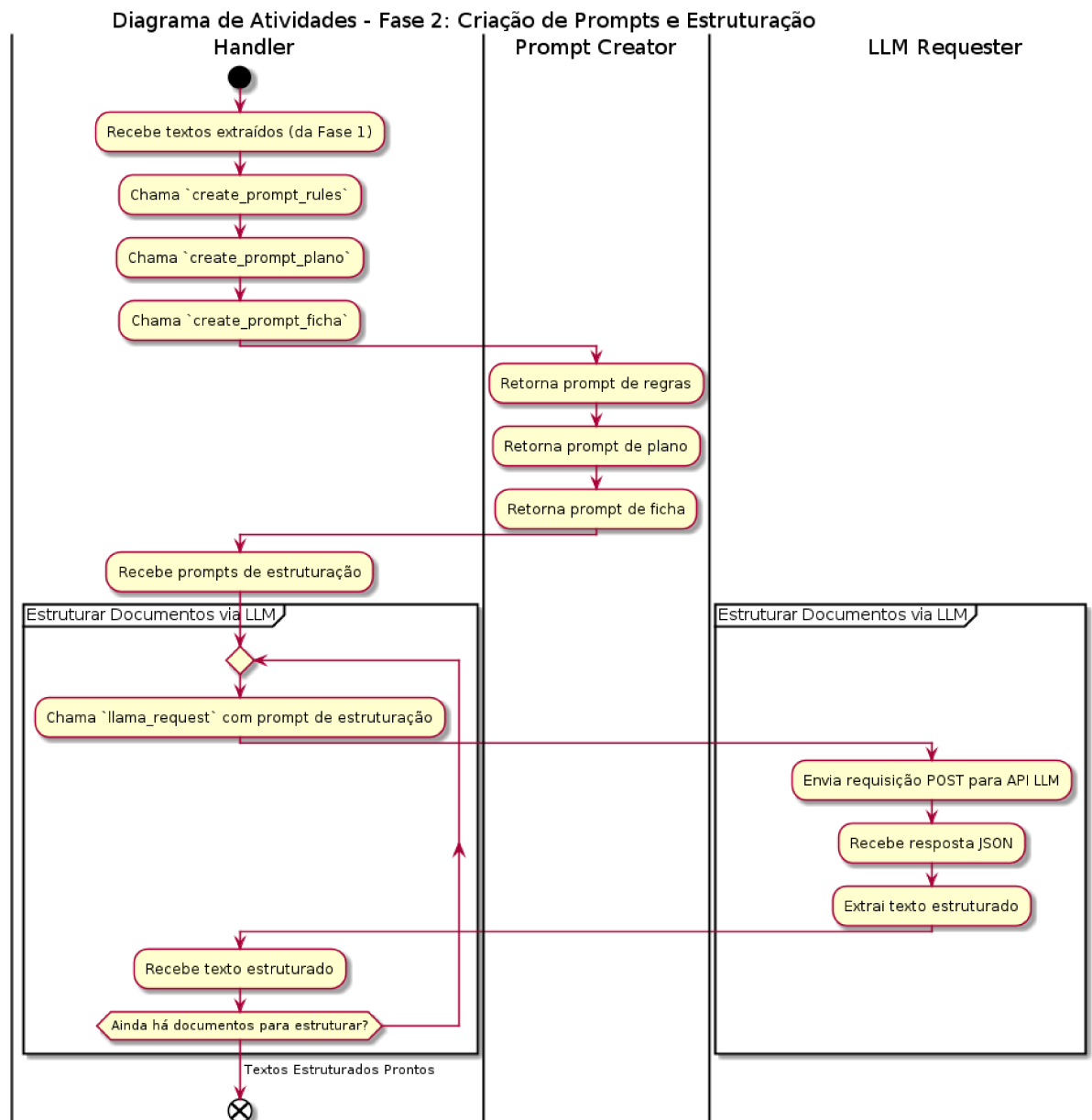


Figura 3 – Fase 2 — Estruturação dos Prompts e Preparação dos Dados.

O segundo diagrama (Figura 3) detalha a etapa intermediária de processamento, focada na estruturação dos documentos via LLM. O ‘handler’, de posse dos textos extraídos na fase anterior, chama o componente ‘Prompt Creator’ para gerar prompts específicos para cada tipo de documento (regras, plano e ficha), visando segmentá-los em seções predefinidas. Com os prompts de estruturação em mãos, o ‘handler’ inicia um processo iterativo: para cada documento, ele envia o prompt correspondente ao ‘LLM Requester’, que faz a chamada à API do modelo de linguagem. O LLM processa o texto e retorna sua versão estruturada. Esse ciclo se repete até que todos os três documentos tenham sido estruturados. Ao final desta fase, o ‘handler’ possui as versões segmentadas e organizadas

dos três documentos originais.

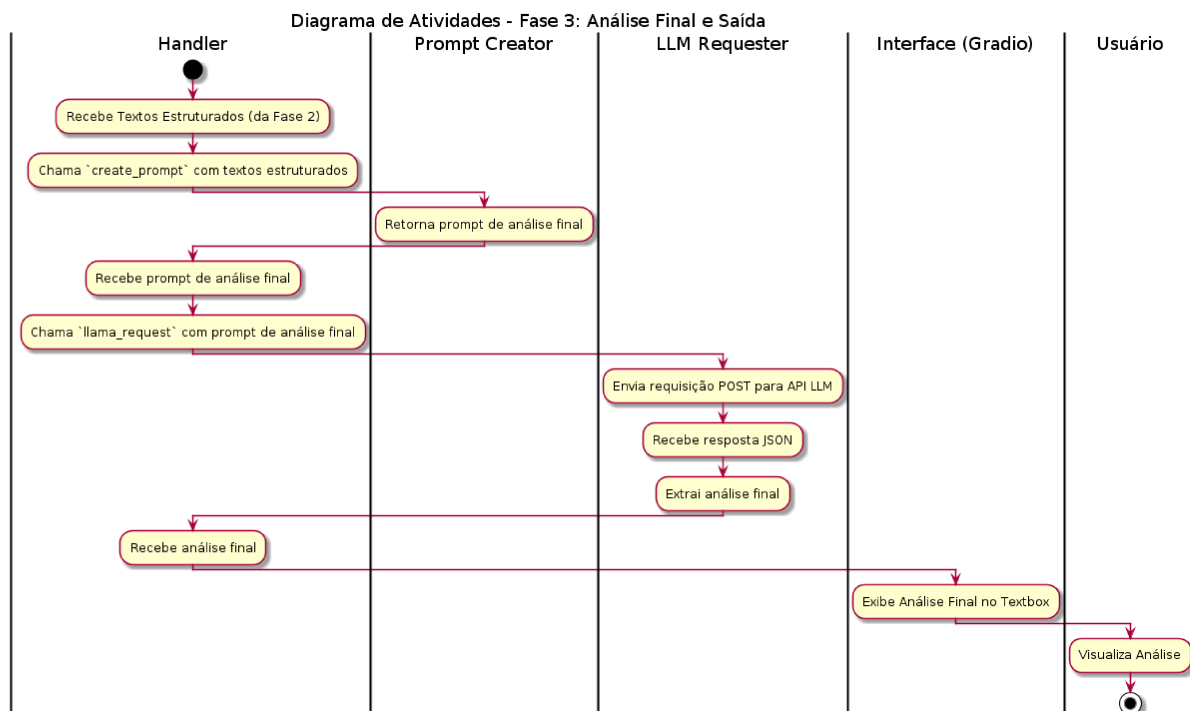


Figura 4 – Fase 3 — Análise Final e Geração do Relatório de Conformidade.

O terceiro diagrama (Figura 4) representa a etapa conclusiva do processo. O ‘handler’ utiliza os textos já estruturados (obtidos na Fase 2) e chama novamente o ‘Prompt Creator’ para gerar um prompt final, agora focado na análise comparativa entre as seções correspondentes do plano de ensino, da ficha da disciplina e das regras. Este prompt final é enviado ao ‘LLM Requester’, que realiza a última chamada à API do LLM. O modelo executa a análise comparativa e retorna o resultado final, contendo o julgamento de conformidade (De acordo, Parcialmente de acordo, Não está de acordo) e as justificativas para cada seção. O ‘handler’ recebe essa análise final e a repassa para a interface Gradio, que a exibe ao usuário em uma caixa de texto, finalizando o fluxo de verificação.

4.1 Coleta de Dados

A coleta de dados foi realizada com o objetivo de obter um conjunto de planos de ensino e suas respectivas correções para avaliar a assertividade do modelo de linguagem utilizado na análise de conformidade. Para isso, foram adotadas as seguintes estratégias:

- **Solicitação a Docentes:** Foi elaborado um formulário enviado para docentes da universidade, solicitando o envio de planos de ensino e, quando possível, versões corrigidas desses documentos. Essa abordagem permitiu coletar exemplos reais de

ajustes feitos em planos de ensino, possibilitando uma avaliação mais precisa do modelo em relação às modificações necessárias para adequação às normas institucionais;

- **Contribuições do Professor Orientador:** Além dos planos de ensino obtidos por meio dos docentes, o professor orientador forneceu documentos adicionais, incluindo planos de ensino preliminares e finais (corrigidos), instruções de preenchimento dos planos, além de planilhas utilizadas e preenchidas pelos revisores para indicar manualmente o resultado da verificação de conformidade. Esses materiais complementaram a base de dados, oferecendo mais diversidade para os testes do modelo;
- **Resgate das Fichas de Disciplina:** Para garantir a análise da conformidade dos planos de ensino em relação às diretrizes curriculares, foram coletadas as fichas de disciplina diretamente do site da Faculdade de Computação da Universidade Federal de Uberlândia. As fichas contêm informações essenciais sobre os conteúdos programáticos, objetivos e competências esperadas, servindo também como referência para a validação dos planos de ensino.

A partir desse conjunto de dados, foi possível estruturar um ambiente de testes realista para avaliar o desempenho do modelo na verificação da conformidade dos planos de ensino.

4.2 Escolha do modelo de linguagem

A escolha do modelo de linguagem foi uma das etapas iniciais do desenvolvimento deste trabalho, tendo como principais critérios a qualidade das respostas, a viabilidade de execução local e a gratuidade do modelo. Após uma análise entre diferentes modelos disponíveis no mercado, optou-se pela utilização do LLaMA 3.1 8B, desenvolvido pela Meta AI.

O principal fator determinante para essa escolha foi a disponibilidade gratuita do modelo, o que permitiu sua utilização sem custos adicionais com licenciamento. Em comparação com alternativas proprietárias, como GPT-4 ou Claude, o LLaMA 3.1 8B se mostrou uma opção economicamente viável e suficientemente robusta para os objetivos do projeto.

O modelo LLaMA 3.1 está disponível em diferentes variantes de tamanho — como 1B, 8B e 70B —, cada uma com características distintas de desempenho e requisitos computacionais. Nesse contexto, a medida de tamanho refere-se ao número de parâmetros do modelo, onde "B" significa bilhões; por exemplo, o modelo 8B possui cerca de 8 bilhões

de parâmetros, enquanto o 70B possui aproximadamente 70 bilhões. No entanto, considerando as especificações da máquina utilizada, o modelo 3.1 8B foi a maior versão que pôde ser executada localmente, representando, assim, o melhor equilíbrio entre capacidade de processamento e limitações de hardware. Modelos maiores, como o LLaMA 3.1 70B, requerem recursos significativamente superiores, inviabilizando sua execução no ambiente utilizado neste projeto.

A configuração da máquina utilizada para execução local foi a seguinte:

Componente	Especificação
Processador	AMD Ryzen 5 8600G (6 núcleos / 12 threads, 4351 MHz)
Memória RAM	32 GB DDR5
Placa de Vídeo Dedicada	NVIDIA GeForce RTX 4060 Ti

Tabela 2 – Especificações da máquina utilizada para execução local do modelo LLaMA.

Essa infraestrutura foi suficiente para suportar o modelo de 8 bilhões de parâmetros com um bom desempenho, permitindo análises complexas de documentos sem a necessidade de serviços em nuvem ou servidores dedicados. A possibilidade de operar *offline* também foi uma vantagem, proporcionando maior controle sobre o ambiente de testes e segurança dos dados.

Em resumo, a escolha pelo LLaMA 3.1 8B foi motivada por uma combinação de fatores técnicos e práticos: gratuidade, qualidade das respostas, compatibilidade com o hardware disponível e execução local eficiente.

4.3 Configuração do Ambiente

Para garantir um ambiente adequado à análise de conformidade de planos de ensino utilizando inteligência artificial generativa, é necessário configurar corretamente o *LM Studio*, carregando o modelo Llama 3.1 8B e ativando seu servidor de API. A configuração do ambiente envolve duas etapas principais: o carregamento do modelo e a inicialização do *endpoint* da API.

O primeiro passo consiste na obtenção do modelo de linguagem. Para isso, dentro do *LM Studio*, deve-se acessar a aba de biblioteca de modelos e localizar o Llama 3.1 8B. Após a seleção da versão adequada para o hardware disponível, realiza-se o download e, posteriormente, o carregamento do modelo na aba correspondente. Com o modelo devidamente carregado, é fundamental configurar seus parâmetros de inferência para otimizar a geração de respostas.

A escolha dos parâmetros utilizados neste trabalho foi resultado de um processo experimental iterativo. Inicialmente, foram adotadas configurações padrão do *LM Studio*,

mas, ao longo da análise de conformidade dos planos de ensino, observou-se a necessidade de ajustes finos para melhorar a consistência, previsibilidade e clareza das respostas geradas. Cada parâmetro foi modificado individualmente, e seus efeitos foram analisados nas saídas do modelo, buscando sempre um equilíbrio entre precisão na avaliação e estabilidade nas respostas.

- **Temperatura:** 0,1.

A temperatura foi ajustada para 0.1 com o objetivo de reduzir a aleatoriedade das respostas geradas. Originalmente, o modelo apresentava variações na estrutura e conteúdo das respostas para a mesma entrada, o que dificultava a análise consistente de conformidade entre documentos. Com a redução da temperatura, as saídas tornaram-se mais determinísticas, proporcionando maior previsibilidade e uniformidade nas respostas — essenciais para avaliações automatizadas de documentos, como planos de ensino.

- **Top-K:** 40.

O valor de Top-K = 40 limita o modelo a escolher o próximo token entre os 40 mais prováveis. Essa restrição ajuda a conter variações linguísticas excessivas e a manter a coerência textual, sem eliminar completamente a flexibilidade do modelo. Esse valor oferece um equilíbrio entre criatividade controlada e precisão.

- **Top-P (*nucleus sampling*):** 0,9.

O parâmetro foi reduzido com o mesmo intuito do ajuste da temperatura: conter a variabilidade nas respostas. O valor escolhido permite que o modelo considere os tokens dentro de um “núcleo” de 90% de probabilidade acumulada, restringindo escolhas a palavras mais prováveis e reforçando a consistência sem tornar a geração completamente rígida.

- **Min-P *Sampling*:** 0,05

O parâmetro *Min-P Sampling* restringe a geração ao ignorar palavras com probabilidade inferior a 0,05. Essa configuração ajuda a eliminar palavras irrelevantes ou incoerentes, especialmente quando combinada com Top-P e Top-K, contribuindo para uma geração mais objetiva e direta, o que é fundamental para a tarefa de análise comparativa.

- **Max Tokens:** 20000.

O limite de 20.000 tokens foi definido após a análise da iteração do modelo para os 23 planos considerados no trabalho. Observou-se que esse valor era suficiente para cobrir a entrada e a saída completas do processo de avaliação de planos de ensino,

mesmo em casos mais extensos. Esse valor garante que todo o conteúdo necessário seja processado sem cortes, especialmente em *prompts* longos com documentos completos e instruções detalhadas.

- ***Repetition Penalty***: 1,1.

Um leve penalizador de repetição foi aplicado com valor de 1.1 para evitar que o modelo gere frases ou expressões redundantes, comuns em tarefas com estrutura repetitiva como relatórios de conformidade. Esse ajuste ajuda a manter a fluidez do texto e a reduzir repetições desnecessárias sem comprometer a clareza das justificativas.

- ***GPU Offload***: 32/32.

A configuração 32/32 indica que todas as 32 camadas do modelo foram alocadas na GPU, o que garante maior desempenho na inferência e aproveitamento máximo da capacidade gráfica disponível no sistema. Isso é especialmente importante ao usar modelos grandes como o LLaMA 8B, que exigem alta largura de banda para processar grandes volumes de texto.

- ***CPU Thread Pool Size***: 6/6.

O uso de 6 *threads* de CPU corresponde ao total de *threads* disponíveis na máquina. Essa configuração foi feita para maximizar o uso da capacidade computacional do sistema e acelerar tarefas auxiliares do modelo, como carregamento de arquivos, pré-processamento e comunicação com a GPU.

As Figuras 5 e 6 ilustram visualmente a configuração dos parâmetros descritos, conforme realizada diretamente na interface do *LM Studio*. A Figura 5 apresenta os parâmetros de inferência ajustados para garantir a consistência e a precisão das respostas durante a análise de conformidade. Já a Figura 6 exibe os parâmetros relacionados ao carregamento do modelo, como a alocação de camadas na GPU e o número de *threads* da CPU. Essas imagens complementam a descrição textual ao demonstrar, de forma prática, como cada configuração foi aplicada no ambiente utilizado.

Info

Inference

Load

 Settings

Basic

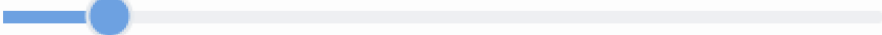
All

▼

Temperature



0,1



Limit Response Length

☐

Context Overflow

Truncate Middle ▼

Stop Strings

Enter a string and press ↵

CPU Threads



6



 Sampling

▼

Top K Sampling

40

Repeat Penalty ☒

1,1

Top P Sampling ☒



0,9



Min P Sampling ☒

0,05

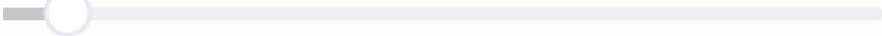


Figura 5 – Parâmetros de inferência modelo Llama.

The screenshot displays the configuration window for the 'Meta Llama 3.1 8B Instruct' model, which has a size of 4.92 GB. The settings are as follows:

- API Identifier:** meta-llama-3.1-8b-instruct
- Auto Unload If Idle (TTL):** ☐ (If set, the model will be automatically unloaded after being idle for the specified amount of time.)
- Context Length:** 20000 (Model supports up to 131072 tokens)
- GPU Offload:** 32 / 32
- CPU Thread Pool Size:** 6
- Evaluation Batch Size:** 512
- RoPE Frequency Base:** ☐ Auto
- RoPE Frequency Scale:** ☐ Auto
- Keep Model in Memory:** ☒
- Try mmap():** ☒
- Seed:** ☐ Random Seed
- Flash Attention:** ☐ Experimental
- K Cache Quantization Type:** ☐ Experimental
- V Cache Quantization Type:** ☐ Experimental

At the bottom, there is a checkbox 'Remember settings for meta-llama-3.1-8b-instruct' which is checked. To the right are 'Cancel' and 'Load Model' buttons, with a keyboard shortcut 'Ctrl + ↵' shown next to the 'Load Model' button.

Figura 6 – Parâmetros de carregamento modelo Llama.

Com o modelo carregado e parametrizado, o próximo passo envolve a inicialização do servidor de API, permitindo a comunicação do modelo com aplicações externas. Para ativá-lo, acessa-se a aba de configurações de API no *LM Studio* e habilita-se a opção correspondente. O servidor, por padrão, opera na porta 1234, podendo ser acessado via <http://localhost:1234>. Caso seja necessário o acesso por outras origens, a funcionalidade *CORS* pode ser ativada.

A API do *LM Studio* disponibiliza diferentes rotas para interação com o modelo. A principal delas é a rota `/v1/completions`, que permite o envio de *prompts* e a obtenção de respostas do modelo. Além disso, a API conta com a rota `/v1/models`, que retorna a lista de modelos carregados, e a rota `/v1/health`, que verifica o estado do servidor. Para interações mais elaboradas, como diálogos, a API disponibiliza a rota `/v1/chat/completions`, que mantém o contexto das conversas, como pode ser observado na figura 7.

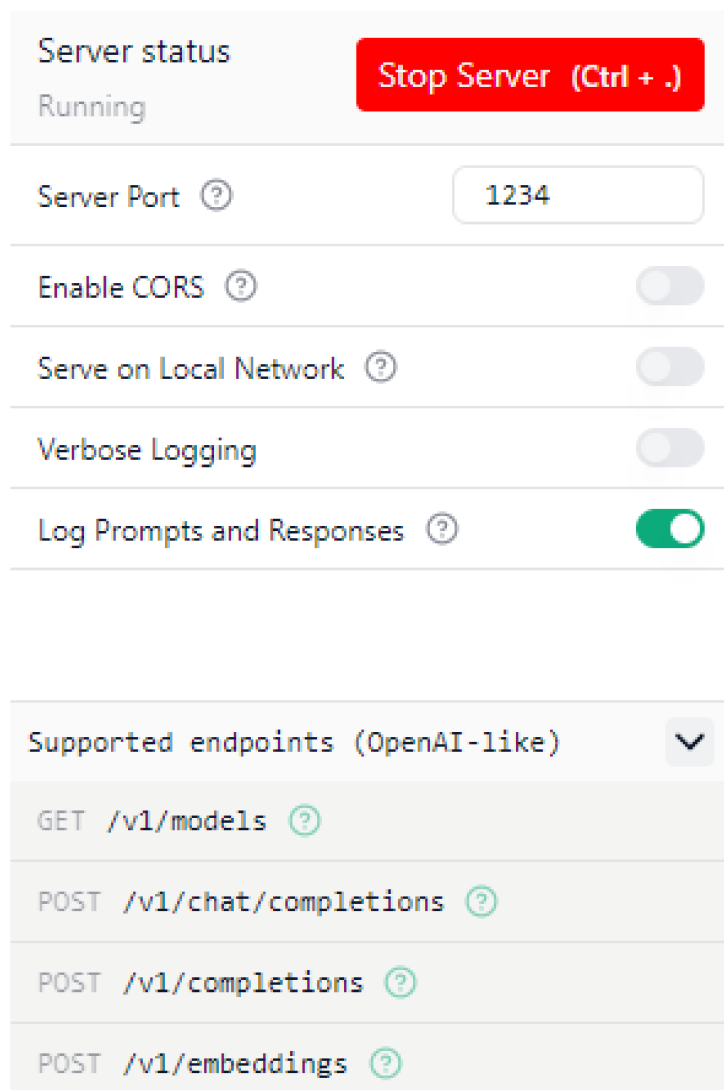


Figura 7 – Servidor LM Studio.

Ao concluir essa configuração, o ambiente estará pronto para a análise de conformidade dos planos de ensino, garantindo a correta comunicação entre o modelo e a aplicação responsável pela avaliação automatizada. A definição precisa dos parâmetros do modelo e a ativação do servidor de API são etapas fundamentais para assegurar um funcionamento eficiente e respostas coerentes com o contexto analisado.

4.4 Recebimento e Processamento dos Arquivos

A interface da aplicação faz o uso da biblioteca *gradio* do Python para criar campos de entrada e saída de dados que permitem o usuário fazer o envio dos arquivos necessários para a validação do plano de ensino e visualizar a resposta do programa.

Os campos de entrada são os seguintes:

- Instruções de elaboração do plano de ensino;
- Plano de ensino a ser analisado;
- Ficha da disciplina relacionada ao plano analisado.

O campo de saída consiste em um campo de texto contendo a resposta do [LLM](#).

The interface consists of two main sections. The top section contains three file upload areas, each with a label in a small box at the top left: 'pdf_regras', 'pdf_plano', and 'pdf_ficha_disciplina'. Each area has a dashed border and contains a central upload icon (an upward arrow inside a square) with the text 'Solte o Arquivo Aqui' above it, '- ou -' in the middle, and 'Clique para o Upload' below it. Below these three areas are two buttons: a grey 'Clear' button and an orange 'Submit' button. The bottom section contains a text output field with the label 'Retorno Llama' and a small copy icon on the right. Below the text field is a grey 'Flag' button.

Figura 8 – Interface da aplicação antes do recebimento dos arquivos.

pdf_regras

Instruções Para Elaboração Plano de Ensino.pdf278.9 KB

pdf_plano

Plano de Ensino Preliminar.pdf97.1 KB

pdf_ficha_disciplina

Ficha da Disciplina.pdf173.2 KB

Clear

Submit

Retorno Llama

* A metodologia descreve como será a integralização da carga horária total da disciplina? ****Sim****: O plano de ensino menciona que as atividades presenciais e extras serão utilizadas para cumprir a carga horária.

* Atente para a quantidade de dias do calendário acadêmico, e deverá haver previsão de aula extra ou AAEs (Atividades acadêmicas extras), que também devem ser discriminadas. ****Sim****: O plano de ensino menciona as atividades presenciais e extras.

****7. AVALIAÇÃO****

* A disciplina contempla mais de uma avaliação, e cada uma com valor individual que não ultrapassa 50 pontos? ****Parcialmente de acordo****: Embora o plano de ensino mencione várias provas e exercícios, a distribuição dos pontos entre elas não é clara.

****8. BIBLIOGRAFIA****

* As bibliografias básica e complementar estão iguais à ficha da disciplina? ****Sim****: A lista de referências parece estar consistente entre o plano de ensino e a ficha da disciplina.

Em geral, o plano de ensino parece estar em conformidade com as instruções e a ficha da disciplina. No entanto, há alguns pontos que precisam ser verificados ou esclarecidos, como a atualização das resoluções referenciadas e a clareza sobre os objetivos e avaliações.

Flag

Figura 9 – Interface da aplicação depois do processamento dos arquivos.

Após o envio dos arquivos por meio da interface, a aplicação executa a etapa de processamento dos documentos em PDF utilizando a biblioteca “pdfplumber”, que permite extrair o conteúdo textual de cada arquivo. Essa etapa é fundamental para preparar os dados que serão enviados ao modelo de linguagem.

Cada um dos três arquivos fornecidos pelo usuário, instruções, plano de ensino preliminar e ficha da disciplina, é lido separadamente e o conteúdo textual de cada docu-

mento é armazenado em variáveis distintas. A extração considera apenas o texto, descon siderando elementos gráficos ou tabelas complexas, garantindo uma representação fiel do conteúdo relevante para análise de conformidade.

Esse processamento automatizado permite que o modelo LLM receba as informa ções em um formato compatível, permitindo a comparação entre o que foi solicitado nas instruções, o que está descrito na ficha da disciplina e o que foi apresentado no plano de ensino enviado.

4.5 Criação do *Prompt*

A construção do *prompt* utilizado na aplicação passou por diversas iterações, com o objetivo de garantir maior consistência, precisão e capacidade de generalização por parte do modelo de linguagem. Inicialmente, foi adotada uma abordagem monolítica, na qual um único *prompt* instruía o modelo a analisar o plano de ensino preliminar em comparação com dois documentos de referência: a ficha da disciplina e o documento de instruções para elaboração do plano. Este *prompt* solicitava que o modelo avaliasse a conformidade de cada seção do plano com base nos critérios estabelecidos nos documentos auxiliares e que classificasse cada item como *De acordo*, *Parcialmente de acordo* ou *Não está de acordo*, sempre apresentando uma justificativa para a decisão.

Apesar de fornecer informações abrangentes, essa abordagem apresentou duas li mitações significativas: (i) o excesso de instruções em uma única chamada comprometia a execução correta de todas as tarefas, levando a omissões ou interpretações errôneas; e (ii) o comprimento do *prompt*, somado aos documentos de entrada, resultava em um elevado consumo de *tokens*, prejudicando a performance do modelo.

Em uma segunda tentativa, foram incorporadas diretamente no *prompt* instruções detalhadas acompanhadas de exemplos de respostas estruturadas, caracterizando uma abordagem do tipo *few-shot prompting*. Essa estratégia resultou em melhorias na consis tência e formatação das respostas, uma vez que o modelo passou a replicar com maior fidelidade a estrutura fornecida. No entanto, tal solução mostrou-se pouco flexível: a inclusão das instruções diretamente no *prompt* suprimia o papel do documento oficial de orientações, além de engessar o comportamento do modelo, limitando sua capacidade de generalização para diferentes estruturas de planos de ensino.

A fim de mitigar esses problemas e promover maior modularidade, a abordagem final adotada foi a de dividir o processo em quatro etapas, cada qual com um *prompt* específico. Essa estrutura modular permitiu maior controle sobre o comportamento do modelo e melhor aproveitamento do contexto textual. A figura 10 descreve as quatro etapas do processo:

Diagrama de Atividades - Verificação de Conformidade (Processo em 4 Etapas)

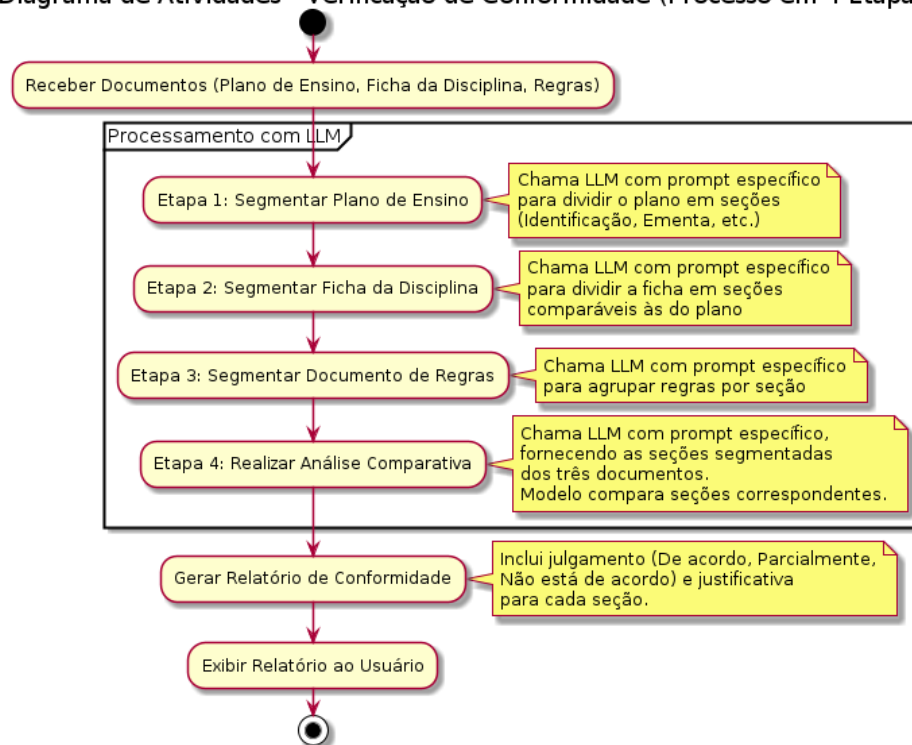


Figura 10 – Diagrama de atividades - Verificação de Conformidade.

1. **Primeira chamada – Divisão do plano de ensino preliminar:** O plano de ensino é analisado individualmente, e o *prompt* solicita a segmentação do conteúdo em seções bem definidas, tais como *Identificação*, *Ementa*, *Justificativa*, entre outras, garantindo que cada trecho contenha apenas as informações pertinentes à respectiva seção.
2. **Segunda chamada – Divisão da ficha da disciplina:** De forma análoga à primeira etapa, este *prompt* visa estruturar o conteúdo da ficha da disciplina em seções comparáveis àsquelas do plano de ensino, facilitando a análise cruzada posterior.
3. **Terceira chamada – Divisão das instruções de elaboração:** O documento de orientações para elaboração do plano é segmentado em regras agrupadas por seções, possibilitando a identificação clara dos critérios de conformidade exigidos.
4. **Quarta chamada – Análise comparativa:** Nesta etapa, um *prompt* semelhante ao inicialmente proposto é utilizado, mas com entrada controlada: as seções extraídas dos três documentos anteriores. O modelo é instruído a comparar os conteúdos correspondentes e a emitir um julgamento de conformidade para cada seção, seguindo a estrutura predefinida (*De acordo*, *Parcialmente de acordo*, *Não está de acordo*) e apresentando uma justificativa fundamentada com base nos documentos analisados.

Essa abordagem dividida permitiu uma análise mais robusta, mitigou problemas de inconsistência e melhorou significativamente a adaptabilidade da solução a diferentes formatos e conteúdos de planos de ensino, mantendo a integridade das diretrizes originais.

4.6 Chamada a API do Llama

A aplicação desenvolvida para análise de conformidade dos planos de ensino faz uso do modelo *Llama 3.1 8B*, acessado por meio de uma API local disponibilizada pelo *LM Studio*. A comunicação com o modelo é realizada por meio da função `llama_request()`, implementada em *Python*. Essa função executa uma requisição HTTP do tipo `POST`, enviando um contexto predefinido juntamente com o conteúdo do plano de ensino a ser avaliado.

O fluxo da função segue os seguintes passos:

1. **Definição do Endpoint:** O endereço da API é obtido a partir da variável de ambiente `COMPLETION_ENDPOINT`, permitindo flexibilidade na configuração.
2. **Construção do Payload:** A requisição é estruturada no formato JSON, seguindo o padrão de mensagens do modelo *Llama*¹. O primeiro elemento do vetor `"messages"` define o comportamento do sistema, instruindo o modelo a atuar como um verificador de conformidade de planos de ensino. O segundo elemento contém a entrada do usuário, que corresponde ao plano a ser analisado.
3. **Configuração dos Headers:** O cabeçalho da requisição define que os dados serão enviados no formato JSON.
4. **Envio da Requisição:** A biblioteca `requests` é utilizada para enviar a requisição `POST` ao endpoint da API.
5. **Processamento da Resposta:** A resposta da API é interpretada, extraindo-se o conteúdo gerado pelo modelo a partir do campo `response.json()["choices"][0]["message"]["content"]`.

A implementação em *Python* da função de chamada à API pode ser vista no código abaixo.

¹ Documentação oficial disponível em: <https://lmstudio.ai/docs/app/api/endpoints/rest>

```
def llama_request(prompt):  
    url = os.getenv("COMPLETION_ENDPOINT")  
  
    payload = json.dumps({  
        "messages": [  
            {  
                "role": "system",  
                "content": """"Você é responsável pela verificação dos planos de ensino de uma faculdade.  
                Se atente às regras de confecção do plano presentes em <rules></rules> e diga se o plano  
                apresentado em <document></document> está de acordo com as regras e de acordo com a ficha  
                de disciplina em <ficha_disciplina></ficha_disciplina>. """"  
            },  
            {  
                "role": "user",  
                "content": prompt  
            }  
        ]  
    })  
    headers = {  
        'Content-Type': 'application/json'  
    }  
  
    response = requests.request("POST", url, headers=headers, data=payload)  
    return response.json()["choices"][0]["message"]["content"]
```

Figura 11 – Requisição para a API do LM Studio.

4.7 Repositório, Instalação e Uso do Sistema

O código-fonte da versão original do sistema de verificação de conformidade de planos de ensino, desenvolvido como parte deste trabalho, está disponível publicamente em um repositório Git hospedado em <https://github.com/ChrystopherPOL/conformidade-planos-de-ensino.git>. Esta seção detalha os passos necessários para instalar, configurar e utilizar o sistema em um ambiente local, além de orientações para potenciais contribuições.

4.7.1 Pré-requisitos

Antes de iniciar a instalação, certifique-se de que os seguintes componentes estejam instalados em seu sistema:

- Python;
- pip (gerenciador de pacotes Python, geralmente incluído com Python);
- Um ambiente para executar Jupyter Notebooks (como Jupyter Lab ou a extensão Jupyter para VS Code);
- Acesso a um endpoint de API compatível com OpenAI para um modelo de linguagem grande (LLM), como o fornecido pelo LM Studio.

4.7.2 Instalação

Siga os passos abaixo para instalar as dependências do projeto:

1. Clone o repositório para sua máquina local:

```
git clone
https://github.com/ChrystopherPOL/conformidade-planos-de-ensino.git

cd conformidade-planos-de-ensino
```

2. (Opcional, mas recomendado) Crie e ative um ambiente virtual Python:

```
python -m venv venv
source venv/bin/activate # Linux/macOS
.\venv\Scripts\activate # Windows
```

3. Instale as bibliotecas Python necessárias utilizando o arquivo ‘requirements.txt’ fornecido:

```
pip install -r requirements.txt
```

4. Configure o endpoint da API do LLM. Crie um arquivo chamado ‘.env’ na raiz do diretório do projeto e adicione a seguinte linha, substituindo

```
‘[URL_DO_SEU_ENDPOINT_LLM]‘
```

pela URL correta do seu servidor local (ex: LM Studio):

```
COMPLETION_ENDPOINT="[URL_DO_SEU_ENDPOINT_LLM]"
```

4.7.3 Uso

Após a instalação e configuração, o sistema pode ser executado através do Jupyter Notebook:

1. Inicie o ambiente Jupyter (Jupyter Lab, Jupyter Notebook, ou VS Code com a extensão Jupyter);
2. Abra o arquivo ‘main.ipynb’ localizado na raiz do diretório do projeto;

3. Execute todas as células do notebook sequencialmente. A última célula iniciará a interface Gradio;
4. Uma URL local (geralmente ‘http://127.0.0.1:7865’) será exibida na saída da última célula. Abra esta URL em seu navegador web;
5. Na interface Gradio, faça o upload dos três arquivos PDF necessários: o documento de regras, o plano de ensino a ser verificado e a ficha da disciplina correspondente;
6. O sistema processará os arquivos, enviará as requisições para o LLM configurado e exibirá a análise de conformidade resultante na caixa de texto da interface.

4.7.4 Como Contribuir

Contribuições para a melhoria do sistema são bem-vindas. Se você encontrar bugs, tiver sugestões de novas funcionalidades ou melhorias, por favor, utilize a seção “Issues” do repositório (<https://github.com/ChrystopherPOL/conformidade-planos-de-ensino/issues>) para reportá-los. Para contribuições de código, recomenda-se abrir uma issue para discussão antes de iniciar o desenvolvimento e seguir as práticas padrão de fork e pull request do Git.

4.8 Considerações Finais

O desenvolvimento da aplicação seguiu uma abordagem prática e estruturada, priorizando eficiência e confiabilidade. A escolha do modelo LLaMA 3.1 8B, rodando localmente pelo LM Studio, atendeu às necessidades de operação offline e geração de respostas consistentes. A configuração cuidadosa dos parâmetros de inferência e a organização do processo em múltiplas etapas de análise permitiram maximizar o aproveitamento do modelo, respeitando suas limitações de contexto.

Além disso, a integração com Gradio para a criação da interface e a automação do processamento de PDFs tornaram o sistema acessível e funcional. As decisões técnicas tomadas neste capítulo garantiram uma base sólida para a análise dos resultados e a identificação de melhorias nos próximos capítulos.

5 Resultados

Este capítulo apresenta os resultados obtidos a partir da avaliação comparativa entre a análise manual de planos de ensino, realizada por docentes da instituição, e a análise automatizada. Inicialmente, descreve-se a metodologia de avaliação adotada, incluindo o processo de revisão manual e a forma como o modelo foi instruído para gerar relatórios de conformidade. Em seguida, são expostas comparações detalhadas entre as classificações humanas e automatizadas para diferentes planos de ensino, com destaque para as convergências, divergências e padrões de erro identificados. A análise dos resultados busca avaliar o desempenho do modelo na interpretação de documentos acadêmicos e discutir suas potencialidades e limitações enquanto ferramenta de apoio à verificação de conformidade.

5.1 Metodologia de Avaliação

Para validar a eficácia do modelo na verificação da conformidade dos planos de ensino, foi estabelecido um processo de avaliação baseado na comparação entre as respostas geradas automaticamente e a revisão manual realizada por professores. Esta etapa teve como objetivo mensurar a precisão das respostas do modelo e identificar possíveis discrepâncias ou melhorias necessárias.

5.1.1 Processo de Revisão Manual

Nos cursos da FACOM, uma das atribuições regulares de membros do colegiado consiste na análise dos planos de ensino preparados para cada disciplina ofertada. Após consulta, alguns membros disponibilizaram suas análises sobre planos reais submetidos ao colegiado para avaliação. Assim, a revisão manual é baseada em documentos com instruções de preenchimento, fichas de cada disciplina e no plano submetido pelo professor ofertante, culminando em um parecer do revisor para cada seção analisada. A avaliação seguiu uma escala qualitativa composta pelos seguintes critérios:

- **De acordo:** a seção do plano atende integralmente às diretrizes estabelecidas.
- **Parcialmente de acordo:** a seção apresenta pequenos desvios ou omissões que necessitam de ajustes.
- **Não está de acordo:** a seção não atende às diretrizes ou apresenta falhas significativas.

Além da classificação, a análise inclui comentários explicativos destacando os pontos que requerem ajustes e melhorias em cada seção.

5.1.2 Processo de Avaliação do Modelo

O sistema baseado em **RAG** foi alimentado com três documentos principais:

- **Plano de Ensino Preliminar:** documento a ser avaliado.
- **Instruções para Elaboração do Plano de Ensino:** conjunto de regras e diretrizes normativas.
- **Ficha da Disciplina:** informações específicas sobre a disciplina, como objetivos, ementa e carga horária.

O modelo processou esses documentos e gerou um relatório de conformidade para cada seção do plano de ensino preliminar, utilizando a mesma escala de avaliação adotada pelos professores. Além da classificação, foram apresentadas justificativas e sugestões de melhoria.

5.1.3 Comparação entre Modelo e Revisão Manual

Para medir a correspondência entre as respostas do modelo e a revisão manual, os resultados foram organizados em uma planilha comparativa, contendo as classificações atribuídas por ambos os métodos. A análise comparativa buscou responder às seguintes questões:

- Em que proporção o modelo concorda com a avaliação dos professores?
- Em quais tipos de seção há maior divergência entre o modelo e os especialistas?
- Quais padrões de erro podem ser identificados na geração de respostas?

Essa abordagem metodológica permitiu uma avaliação sistemática da capacidade do modelo em auxiliar na verificação de documentos acadêmicos, identificando seus acertos, limitações e oportunidades de aprimoramento.

5.2 Análise Comparativa entre Modelo e Revisão Manual

Esta seção tem como objetivo realizar uma comparação entre os resultados obtidos por meio da análise manual de um plano de ensino, realizada por docentes da instituição, e aqueles gerados automaticamente pelo modelo de linguagem LLaMA. A comparação visa

verificar a aderência do modelo aos critérios estabelecidos para a elaboração de planos de ensino e avaliar sua capacidade de identificar conformidades e não conformidades de forma autônoma.

A análise manual é considerada a referência, por se basear na experiência dos professores e no conhecimento normativo institucional. Já a análise automática foi realizada com base no conteúdo textual extraído do plano de ensino, interpretado pelo modelo LLaMA de forma independente, sem acesso prévio aos critérios utilizados pelos avaliadores humanos.

A comparação foi organizada em uma tabela, contendo cada item avaliado, a resposta humana e a resposta do modelo. Em seguida, é apresentada uma análise crítica das semelhanças, diferenças e limitações observadas, com vistas a compreender o potencial e os desafios da utilização de modelos de linguagem na verificação automatizada de documentos acadêmicos.

No total, foram avaliados 23 planos de ensino diferentes e suas respectivas análises. Desses 23 planos, 17 apresentavam-se no formato de planilhas com observações sobre cada seção do documento de regras de elaboração dos planos separada. Nos demais, as observações foram listadas em parágrafos únicos sem seguir uma estrutura específica.

As três primeiras análises foram escolhidas intencionalmente para representar os erros mais recorrentes identificados nas comparações entre a análise do modelo e a revisão manual. Cada um desses exemplos ilustra falhas específicas, como omissões de elementos obrigatórios, links quebrados na seção de bibliografia ou falhas de formatação do arquivo. Já o exemplo adicional de plano de ensino vazio foi incluído com o propósito de avaliar o comportamento do modelo frente a um cenário-limite, em que não há conteúdo a ser analisado. Essa escolha permitiu verificar se o modelo apresentaria alucinações — ou seja, inferências infundadas — ou se reconheceria corretamente a ausência de dados, reforçando a análise crítica sobre sua confiabilidade e robustez em diferentes contextos.

5.2.1 Primeira Análise

A Tabela 2 apresenta os resultados das análises realizadas sobre um plano de ensino referente a uma disciplina da área de estrutura de dados. A primeira coluna indica as seções avaliadas. A segunda coluna descreve a análise manual realizada por um revisor humano, enquanto a terceira apresenta a análise automatizada gerada pelo sistema proposto com base no modelo LLaMA 3.1.

Item	Análise Manual	Análise LLaMA
Identificação	De acordo	De acordo
Ementa	De acordo	De acordo
Justificativa	De acordo	Parcialmente de acordo: A justificativa do plano de ensino (A disciplina é componente do Núcleo de Formação Básica. Visa capacitar o aluno a identificar e desenvolver modelos matemáticos, determinando classes de problemas que podem ser resolvidos com o uso desses.) não está presente na ficha da disciplina. No entanto, a justificativa é coerente e complementa a informação geral sobre a disciplina.
Objetivos	De acordo	De acordo
Programa	De acordo	De acordo

Item	Análise Manual	Análise LLaMA
Metodologia	Parcialmente de acordo: Remover referências para orientações relativas aos protocolos de segurança do período da pandemia. Carga horária indicada de 64ha com 8ha extras. Revisar considerando que são 15 semanas de aula, devendo a complementação prevista ser de pelo menos 12ha. Esclarecer se haverão aulas extras em horário diferente do regular ou irá complementar com AAE. Informa sobre o uso do Teams, mas sem maiores detalhes (equipe, link, etc). Deve ser previsto ao menos 1 hora de atendimento semanal por disciplina. O horário informado (Quarta, 11:00-12:00), coincide com os horários de atendimento de outra disciplina ofertada pela docente.	Parcialmente de acordo: Parcialmente de acordo Justificativa: O plano de ensino apresenta a metodologia da disciplina (O conteúdo será desenvolvido presencialmente em sala de aula, com material didático apresentado em projetor datashow e lousa. A carga horária será complementada na forma de atividades AAE.), que é coerente com as expectativas. No entanto, não há menção explícita à inclusão de listas de exercício para realização fora do horário de aula e atividades opcionais de aprofundamento presentes na ficha da disciplina.

Item	Análise Manual	Análise LLaMA
Avaliação	Parcialmente de acordo: Informa que a avaliação será baseada em trabalhos realizados em todas as aulas... Indica os tópicos a serem cobertos em cada período, mas não dá maiores detalhes (tipo de trabalho, prazo, data de entrega, etc). Explicar como seria a estratégia para recuperação em caso de falta ou não entrega de atividades avaliativas das aulas. Deixar claro também o prazo para entregas. Informa a data e conteúdo para avaliação de recuperação, mas não indica o valor e forma de substituição/composição com a nota final.	Parcialmente de acordo: O plano de ensino apresenta os critérios de avaliação (Os alunos serão avaliados através de diversos trabalhos individuais e em grupo, avaliando e considerando o conteúdo apresentado em sala de aula até aquele ponto do curso.), que são coerentes com as expectativas. No entanto, não há menção explícita à inclusão de atividades avaliativas de recuperação de aprendizagem para os alunos que não obtiverem o rendimento mínimo para aprovação e com frequência mínima de 75% no componente curricular presente na Resolução CONGRAD Nº 46/2022.
Bibliografia	Parcialmente de acordo: Destacar nomes dos livros em negrito (padrão ABNT). Apesar de seguir o que está na ficha, atualizar URL quebrada na bibliografia.	De acordo

Tabela 3 – Comparação entre análise manual e resposta do LLaMA.

Justificativa: A análise manual considerou a seção de *Justificativa* como *De acordo*, uma vez que o conteúdo do plano contemplava adequadamente os elementos previstos na ficha da disciplina. Já a análise realizada pelo Llama 3.1 indicou que, embora a justificativa do plano incluísse todas as informações necessárias, havia a inserção de dados adicionais. No entanto, o modelo ressaltou que essas informações complementares eram pertinentes e coerentes com a proposta da disciplina, classificando a seção como *Parcialmente de Acordo*.

Metodologia: Ambas as análises classificaram a seção de *Metodologia* como *Par-*

cialmente de Acordo. O modelo identificou corretamente a ausência de menções às horas complementares ou Atividades Acadêmicas Extracurriculares (AAEs) no plano de ensino. Por outro lado, o Llama 3.1 não foi capaz de identificar problemas relacionados ao calendário acadêmico reduzido, tampouco conflitos nos horários de atendimento dos professores, uma vez que sua análise é restrita ao conteúdo presente no documento. A análise manual ainda observou que, embora o uso da plataforma Microsoft Teams tenha sido mencionado, não foram fornecidos detalhes sobre sua utilização.

Avaliação: A seção de *Avaliação* também foi classificada como *Parcialmente de Acordo* pelas duas análises. Ambas apontaram a ausência de informações sobre as formas de avaliação de recuperação. A avaliação manual ainda destacou que, embora o plano de ensino indicasse a realização de trabalhos em todas as aulas como forma de avaliação, não foram fornecidos detalhes adicionais, tais como tipo de trabalho, prazos ou datas de entrega.

Bibliografia: Na seção de *Bibliografia*, a análise manual classificou como *Parcialmente de Acordo*, tendo em vista que, apesar de os títulos estarem em conformidade com a ficha da disciplina, alguns links disponibilizados encontravam-se quebrados. O modelo Llama 3.1, contudo, classificou a seção como *De acordo*, uma vez que não realiza verificações sobre a funcionalidade de links e avaliou apenas o conteúdo textual apresentado.

Demais Seções: As seções de *Identificação*, *Ementa*, *Objetivos* e *Programa* foram classificadas como *De acordo* tanto pela análise manual quanto pela automatizada, evidenciando a consistência do conteúdo destas partes do plano em relação às exigências da ficha da disciplina.

Considerações Finais: O modelo Llama 3.1 demonstrou desempenho satisfatório na identificação de conformidades nas seções estruturais e de conteúdo. No entanto, as diferenças observadas nas seções de *Justificativa* e *Bibliografia* ilustram limitações importantes, especialmente no que se refere à interpretação de informações adicionais e à incapacidade de analisar aspectos não textuais, como a funcionalidade de links. Tais limitações reforçam a necessidade de revisão humana complementar, sobretudo em contextos que exigem maior rigor na avaliação da conformidade documental.

5.2.2 Segunda Análise

A Tabela 3 apresenta os resultados das análises realizadas sobre um plano de ensino referente a uma disciplina da área de sistemas distribuídos. A primeira coluna indica as seções avaliadas. A segunda coluna descreve a análise manual realizada por um revisor humano, enquanto a terceira apresenta a análise automatizada gerada pelo sistema proposto com base no modelo LLaMA 3.1.

Item	Análise Manual	Análise LLaMA
Identificação	Parcialmente de acordo: Informações no campo Observações estão desatualizadas. Revisar modelo no documento com orientações (processo SEI 23117.085845/2024-31)	De acordo
Ementa	De acordo	De acordo
Justificativa	De acordo	De acordo
Objetivos	De acordo	De acordo
Programa	De acordo	De acordo
Metodologia	Parcialmente de Acordo: - Remover referências para orientações relativas aos protocolos de segurança do período da pandemia. Carga horária indicada de 64ha com 8ha extras. Revisar considerando que são 15 semanas de aula, devendo a complementação prevista ser de pelo menos 12ha. Esclarecer se haverão aulas extras em horário diferente do regular ou irá complementar com AAE.. - Deve ser previsto ao menos 1 hora de atendimento semanal por disciplina. O horário informado (Quarta, 11:00-12:00), coincide com os horários de atendimento de outra disciplina ofertada pela docente.	Parcialmente de acordo: A metodologia do plano de ensino está presente, mas não contempla a previsão da totalidade da carga horária da disciplina no plano de ensino com atividades presenciais e complementação necessária com aulas extras ou outras AAEs. Além disso, não há informações sobre como será feita a integralização da carga horária total da disciplina.

Item	Análise Manual	Análise LLaMA
Avaliação	Parcialmente de acordo: - Explicar como seria a estratégia para recuperação em caso de falta ou não entrega de atividades avaliativas das aulas. Deixar claro também o prazo para entregas. - Informa a data e conteúdo para avaliação de recuperação, mas não indica o valor e forma de substituição/composição com a nota final.	Parcialmente de acordo: A avaliação do plano de ensino está presente e contempla a distribuição de pontos para as atividades avaliativas, mas não há informações sobre como serão aplicados os critérios de correção das atividades avaliativas. Além disso, não há informações sobre como será feita a validação da assiduidade dos discentes.
Bibliografia	Parcialmente de acordo: Destacar nomes dos livros em negrito (padrão ABNT).	De acordo

Tabela 4 – Comparação entre análise manual e resposta do LLaMA.

Metodologia: Ambas as análises classificaram a seção como *Parcialmente de Acordo*. A análise automatizada com o modelo Llama identificou corretamente a ausência de menção às Atividades Acadêmicas Extracurriculares ou horas complementares. No entanto, não foi capaz de verificar a adequação ao calendário acadêmico reduzido, por não possuir acesso ao calendário institucional. Além disso, não detectou conflitos de horários de atendimento do professor, pois sua análise se restringe ao conteúdo disponível no plano de ensino.

Avaliação: Essa seção foi classificada como *Parcialmente de Acordo* pelas duas abordagens. Foram observadas falhas como a falta de coerência entre os critérios de avaliação e os objetivos da disciplina, além de ausência de detalhamento quanto à distribuição de pesos.

Bibliografia: A análise manual apontou *Parcialmente de Acordo* devido à ausência de destaque em negrito para os livros da bibliografia básica. Já a análise com o modelo Llama classificou como *De acordo*, pois o modelo não tem capacidade de avaliar formatação visual, focando apenas no conteúdo textual.

Considerações Finais: O modelo Llama 3.1 demonstrou bom desempenho na identificação de conformidades baseadas no conteúdo textual do plano de ensino. No entanto, apresenta limitações ao lidar com informações contextuais externas (como o

calendário acadêmico) e aspectos de formatação visual (como negrito). Portanto, seu uso é recomendado como ferramenta de apoio à análise manual, principalmente em contextos que exijam automação preliminar ou em larga escala.

5.2.3 Terceira Análise

A Tabela 4 apresenta os resultados das análises realizadas sobre um plano de ensino referente a uma disciplina vinculada ao trabalho de conclusão de curso. A primeira coluna indica as seções avaliadas. A segunda coluna descreve a análise manual realizada por um revisor humano, enquanto a terceira apresenta a análise automatizada gerada pelo sistema proposto com base no modelo LLaMA 3.1.

Item	Análise Manual	Análise LLaMA
Identificação	De acordo	De acordo
Ementa	De acordo	De acordo
Justificativa	De acordo	De acordo
Objetivos	Parcialmente de acordo: No objetivo geral, há uma pequena diferença em relação a ficha. A parte “associada a área de computação” foi removida.	De acordo
Programa	De acordo	De acordo
Metodologia	De acordo	De acordo
Avaliação	De acordo	De acordo
Bibliografia	Parcialmente de acordo: Há links para os livros da biblioteca (conforme ficha), mas que estão quebrados, devendo ser atualizados.	De acordo

Tabela 5 – Comparação entre análise manual e resposta do LLaMA.

Objetivos: A análise manual classificou esta seção como *Parcialmente de Acordo*, pois foi observada uma alteração sutil mas significativa no objetivo geral: a expressão “associada à área de computação”, presente na ficha da disciplina, foi removida no plano de ensino. O modelo Llama, por outro lado, não indicou essa discrepância, classificando a seção como *De acordo* — o que evidencia sua limitação em detectar pequenas omissões conceituais que exigem comparação contextual mais refinada.

Bibliografia: A análise manual também identificou um problema relevante nesta seção: embora os livros estejam de acordo com a ficha e contenham links para a biblioteca,

esses links estão quebrados e precisam ser atualizados. O modelo Llama classificou como *De acordo*, pois analisa apenas o conteúdo textual e não verifica a funcionalidade de links externos.

Considerações Finais: O modelo Llama 3.1 se mostra eficiente na verificação de aspectos estruturais e de conteúdo textual, mas ainda possui limitações na identificação de alterações conceituais sutis e problemas relacionados à funcionalidade de elementos não textuais (como links). Assim, reforça-se o papel complementar da análise automatizada à revisão humana, especialmente em contextos que exigem precisão e aderência total às fichas e diretrizes institucionais.

5.2.4 Análise de documento em branco

Neste experimento, foi empregado um plano de ensino propositalmente esvaziado, no qual todas as informações foram removidas das seções, mantendo-se apenas a estrutura do documento original. O plano utilizado corresponde exatamente ao mesmo documento analisado nos testes anteriores, porém com o conteúdo textual de cada seção integralmente suprimido. Essa abordagem teve como objetivo avaliar a capacidade do modelo Llama 3.1 em detectar a ausência total de informações nas seções obrigatórias, simulando um cenário extremo de não conformidade e analisar possíveis alucinações.

Item	Análise LLaMA
Identificação	Não está de acordo.
Ementa	Não está de acordo: EA seção “Ementa” do plano de ensino está vazia, enquanto a ficha da disciplina apresenta uma descrição detalhada sobre o conteúdo da disciplina (Introdução a algoritmos; Ambientes de programação; Fundamentos de testes e depuração; Linguagens procedimentais: tipos de dados e estruturas de controle).
Justificativa	Não está de acordo: A seção “Justificativa” do plano de ensino está vazia, enquanto a ficha da disciplina apresenta uma justificativa para o desenvolvimento dos objetivos específicos da disciplina.
Objetivos	Não está de acordo: A seção “Objetivo” do plano de ensino está vazia, enquanto a ficha da disciplina apresenta os objetivos gerais e específicos da disciplina (compreender a sequência lógica para criação de um algoritmo; aprender uma linguagem de programação estruturada; aprender recursos lógicos básicos presentes nas linguagens de programação).

Item	Análise LLaMA
Programa	Não está de acordo: A seção “Programa” do plano de ensino está vazia, enquanto a ficha da disciplina apresenta um detalhado programa com as seguintes unidades: Introdução a Algoritmos; Ambientes de Programação; Fundamentos de Testes e Depuração; Linguagens Procedimentais: tipos de dados e estruturas de controle.
Metodologia	Não está de acordo: A seção “Metodologia” do plano de ensino está vazia, enquanto a ficha da disciplina não apresenta informações sobre como será integralizada a carga horária total da disciplina ou como serão realizadas as atividades acadêmicas extras.
Avaliação	Não está de acordo: Justificativa: A seção “Avaliação” do plano de ensino está vazia, enquanto a ficha da disciplina não apresenta informações sobre os critérios de correção das atividades avaliativas ou como serão realizadas as avaliações parciais e finais.
Bibliografia	Parcialmente de acordo: Justificativa: A seção “Bibliografia” do plano de ensino está vazia, enquanto a ficha da disciplina apresenta uma lista de bibliografia básica e complementar. No entanto, não há informações sobre como os discentes terão acesso a essas fontes ou como elas serão utilizadas no curso.

Tabela 6 – Comparação entre análise manual e resposta do LLaMA.

Os resultados indicam que o modelo é eficaz na identificação de seções completamente ausentes, classificando corretamente seis das sete seções como *Não está de Acordo*.

Contudo, observou-se uma incongruência na classificação atribuída à seção *Bibliografia*. Apesar de a seção encontrar-se vazia, o que também foi explicitamente reconhecido na justificativa textual gerada pelo modelo, a classificação atribuída foi *Parcialmente de Acordo*.

Essa discrepância entre a descrição textual e a classificação formal evidencia uma limitação do modelo em alinhar sua avaliação descritiva ao sistema de categorização adotado para fins de conformidade. Tal comportamento parece derivar de uma estratégia de cautela do modelo, que tende a evitar classificações extremas em casos ambíguos ou com ausência explícita de conteúdo. Isso pode ocorrer quando o modelo, mesmo reconhecendo a ausência da seção no texto, opta por um julgamento intermediário (“Parcialmente de Acordo”) ao invés de “Não está de Acordo”, como forma de mitigar falsos negativos. Essa abordagem, embora prudente do ponto de vista do modelo, pode comprometer a precisão

da classificação formal e reforça a necessidade de supervisão humana para ajustes nos casos-limite.

5.3 Análise Geral das Comparações

Ao todo, 23 planos de ensino foram analisados neste estudo. Esses planos foram divididos em dois grupos distintos:

- **Grupo A – Planos Não Estruturados (6 planos):** Avaliados por meio de parecer geral, tanto pelo sistema quanto por revisão manual docente. Permitiram uma análise qualitativa comparativa, embora sem padronização por seção.
- **Grupo B – Planos Estruturados (17 planos):** Avaliados com base em 8 seções específicas, com julgamentos independentes por parte do sistema e dos revisores humanos. Essa estrutura possibilitou uma avaliação quantitativa e mais objetiva.

5.3.1 Grupo A – Planos Não Estruturados

Dos 6 planos analisados neste grupo:

- Todos os planos foram comparados diretamente com a avaliação manual.
- A análise focou em identificar o **nível de convergência entre os pareceres**, especialmente nos seguintes critérios: objetivos, metodologia, avaliação, bibliografia, ementa, justificativa, conteúdos e competências.

Principais Resultados:

- Total de critérios avaliados: 90
- Julgamentos coincidentes entre o modelo e a revisão manual: 72
- Índice de conformidade: $72 / 90 = 0,80$ (80%)
- Média de convergência por plano: entre 5 e 6 critérios de 8, o que representa aproximadamente 62% a 75% de concordância.
- Erros de omissão foram os mais frequentes. Em uma análise de metodologia, por exemplo, a revisão manual destacou 6 pontos críticos, nenhum dos quais foi mencionado pelo sistema.
- Na seção de Avaliação, os pareceres humanos destacaram 4 a 5 falhas, enquanto o sistema reconheceu apenas a ausência da recuperação de aprendizagem.

- Em outros casos, a ferramenta classificou como “parcialmente de acordo” itens que foram julgados como “não conforme” na análise humana.
- Em 3 planos, a análise manual sugeriu *evitar o horário de almoço para o atendimento ao aluno*, mas o modelo não fez essa observação. Além disso, em um dos outros planos houve a divergência nas análises devido ao conflito de horários de atendimento do mesmo professor, que mostra uma limitação do modelo atual levar em consideração apenas 1 plano de ensino por vez.

5.3.2 Grupo B – Planos Estruturados (17 planos)

Neste grupo:

- Todos os 17 planos foram avaliados por seção com base em 8 critérios padronizados.
- Foi possível realizar comparações diretas com a revisão manual para cada um dos 136 julgamentos (17 planos $\times$ 8 critérios).

Percentuais Consolidados:

Convergência por plano	Quantidade de planos
7 de 8 critérios	2
6 de 8 critérios	8
5 de 8 critérios	4
4 de 8 critérios	1
3 de 8 critérios	2

Tabela 7 – Distribuição de convergência entre sistema e análise manual

- Total de critérios avaliados: 136
- Convergência entre sistema e revisão humana: 97 critérios (aproximadamente 71,3%)
- Divergências: 39 critérios (aproximadamente 28,7%)

Principais Divergências:

- **Bibliografia:** 7 divergências. Todos os 7 casos representam erros de formatação e links quebrados.
- **Justificativa:** 13 divergências. Em 9 delas, o modelo considerou a seção como “parcialmente de acordo” porque o conteúdo era igual ao da ficha da disciplina, mas com algumas informações extras. O modelo achou essas adições coerentes, mas recomenda-se uma revisão para confirmar se elas realmente fazem sentido para a disciplina.

- Em outros casos, a divergência aconteceu porque o modelo sugeriu revisar as adições feitas à ficha para garantir que elas realmente são adequadas. Um exemplo claro disso é um dos planos em que só 3 dos 8 critérios bateram com a análise manual. Nesse caso, 4 dos critérios divergentes eram apenas sugestões de revisão — o modelo achou as adições coerentes, mas ainda assim recomendou revisar — enquanto a análise manual considerou tudo totalmente correto.

5.3.3 Considerações Finais das Comparações

- A convergência média geral entre os 6 planos do Grupo A e os 17 do Grupo B foi de aproximadamente 70% com a avaliação manual, o que é positivo, mas insuficiente para validação final automática.
- Os pontos críticos de erro concentram-se nas seções de Metodologia, Justificativa e Avaliação, refletindo a dificuldade do modelo em capturar nuances interpretativas mais sutis.
- O sistema se mostra útil como ferramenta de triagem, identificando seções que precisam de maior atenção, mas não substitui a avaliação humana especializada.

5.4 Considerações Finais

Apesar dos avanços proporcionados pelo uso de modelos de linguagem para a verificação automatizada de planos de ensino, a aplicação apresenta algumas limitações que devem ser consideradas na análise dos resultados obtidos. Tais limitações decorrem, em grande parte, da natureza textual da entrada processada, da ausência de certos metadados e da incapacidade do modelo em interpretar aspectos que exigem contexto externo ou elementos visuais.

Um dos principais desafios identificados refere-se à análise da seção de bibliografia. O modelo não é capaz de identificar, por meio do texto extraído do PDF, a ausência de formatação em negrito para as obras obrigatórias, uma exigência comum nas diretrizes de elaboração dos planos. Como o processo de extração textual descarta qualquer formatação visual, essas informações são perdidas antes mesmo de chegarem ao modelo. Além disso, o modelo também não consegue verificar a validade dos links apresentados na bibliografia ou em outras seções, o que compromete a capacidade de avaliar a funcionalidade de recursos complementares indicados.

Outro tipo de limitação observada envolve situações que exigem verificação cruzada de dados externos, como no caso do conflito de horários de atendimento docente entre diferentes disciplinas. Como a aplicação analisa um plano de ensino isoladamente, sem

acesso ao conjunto completo de planos atribuídos ao mesmo professor, o modelo não possui informações suficientes para detectar esse tipo de inconsistência.

Dessa forma, observa-se que o modelo atua de maneira mais precisa em casos em que a não conformidade está explicitamente descrita no conteúdo textual do plano analisado, especialmente quando relacionada às diretrizes normativas fornecidas nas instruções. Entretanto, aspectos que dependem de formatação visual, metadados externos ou comparações interdocumentais não são capturados pelo modelo, o que pode levar a análises incompletas.

Como sugestões para trabalhos futuros, destaca-se a possibilidade de:

- Integrar ferramentas de processamento de PDF que preservem formatação visual, como identificação de negrito e itálico;
- Utilizar verificadores automáticos de links para complementar a análise da bibliografia;
- Desenvolver uma base de dados relacional com planos de ensino e professores, permitindo a checagem de conflitos de horários e outras inconsistências interdisciplinares.
- Realizar mais experimentações com os parâmetros e a formulação dos *prompts*, visando melhorar a precisão e a coerência das análises automatizadas.

Essas melhorias podem ampliar significativamente a assertividade da aplicação e sua aplicabilidade em contextos institucionais de validação de planos de ensino.

6 Conclusão

Este trabalho investigou a aplicação de técnicas de Inteligência Artificial Generativa, com ênfase em modelos de linguagem de grande porte [LLMs](#), no processo de verificação de conformidade de documentos acadêmicos, especificamente planos de ensino. A abordagem proposta baseou-se na integração de um mecanismo de Recuperação Aumentada por Geração [RAG](#) com a utilização do modelo LLaMA 3.1 8B, executado localmente por meio da plataforma LM Studio. A solução desenvolvida foi capaz de processar documentos em formato PDF, recuperar instruções normativas e fichas disciplinares associadas, e, a partir dessas informações, gerar relatórios avaliando o grau de aderência de cada seção do plano às diretrizes estabelecidas.

Os resultados obtidos indicam que a ferramenta possui potencial para produzir análises coerentes e alinhadas com os julgamentos manuais em grande parte dos casos, com um índice de conformidade geral de 70%. As principais convergências foram observadas em seções com requisitos objetivos e bem definidos, enquanto as divergências mais recorrentes ocorreram em critérios de natureza mais interpretativa, como justificativas e metodologias. Limitações técnicas também afetaram o desempenho, especialmente em situações que demandavam leitura de metadados, verificação de formatação ou comparação entre planos distintos.

Apesar dessas limitações, os resultados sugerem que a ferramenta pode se tornar um recurso valioso no apoio ao processo avaliativo, sobretudo como suporte preliminar à revisão manual. A estruturação padronizada das respostas contribuiu para aumentar a transparência e a reprodutibilidade do processo avaliativo, enquanto a execução local do modelo assegurou a confidencialidade dos dados institucionais.

Como perspectivas para trabalhos futuros, sugere-se a incorporação de técnicas de processamento de documentos que preservem elementos visuais, a adoção de verificadores automáticos de links e a construção de uma base de dados relacional que permita a análise contextualizada entre múltiplos planos de ensino. Tais aprimoramentos têm o potencial de ampliar a precisão e a abrangência da aplicação desenvolvida, consolidando-a como uma ferramenta robusta de apoio à gestão pedagógica. Além disso, destaca-se a necessidade de realizar mais testes e análises em diferentes cenários, a fim de validar a eficácia da aplicação em contextos diversos e garantir sua adaptabilidade.

Conclui-se que a utilização de [LLMs](#) em conjunto com técnicas de recuperação de informação oferece uma alternativa viável para apoiar a análise de conformidade de documentos educacionais, promovendo ganhos em agilidade e padronização. A ferramenta apresentou bons resultados na identificação de inconformidades textuais e demonstra po-

tencial para evoluir com novas funcionalidades que ampliem sua abrangência e precisão.

Referências

- ALQAHTANI, T.; BADRELDIN, H. A.; ALRASHED, M.; ALSHAYA, A. I.; ALGHAMD, S. S.; SALEH, K. bin; ALOWAIS, S. A.; ALSHAYA, O. A.; RAHME, I.; YAMI, M. S. A.; ALBEKAIRY, A. M. The emergent role of artificial intelligence, natural learning processing, and large language models in higher education and research. **Research in Social and Administrative Pharmacy**, v. 19, n. 8, p. 1236–1242, 2023. Citado 2 vezes nas páginas 16 e 17.
- ARORA, C.; GRUNDY, J.; PULI, L.; LAYTON, N. **Towards Standards-Compliant Assistive Technology Product Specifications via LLMs**. 2024. Disponível em: <<https://arxiv.org/abs/2404.03122>>. Citado 2 vezes nas páginas 16 e 17.
- BROWN, T. B.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A.; AGARWAL, S.; HERBERT-VOSS, A.; KRUEGER, G.; HENIGHAN, T.; CHILD, R.; RAMESH, A.; ZIEGLER, D. M.; WU, J.; WINTER, C.; HESSE, C.; CHEN, M.; SIGLER, E.; LITWIN, M.; GRAY, S.; CHESSE, B.; CLARK, J.; BERNER, C.; MCCANDLISH, S.; RADFORD, A.; SUTSKEVER, I.; AMODEI, D. **Language Models are Few-Shot Learners**. 2020. Disponível em: <<https://arxiv.org/abs/2005.14165>>. Citado 2 vezes nas páginas 11 e 13.
- CEJAS, O. A.; AZEEM, M. I.; ABUALHAIJA, S.; BRIAND, L. C. Nlp-based automated compliance checking of data processing agreements against gdpr. **IEEE Transactions on Software Engineering**, v. 49, n. 9, p. 4282–4303, 2023. Citado 2 vezes nas páginas 16 e 17.
- COMMUNITY, O. L. **Awesome Open LLMs**. 2023. <<https://github.com/open-llms/awesome-open-llms>>. Acesso em: 08 abr. 2025. Citado na página 14.
- CUTTING, G.; CUTTING-DECELLE, A. **Intelligent Document Processing – Methods and Tools in the real world**. 2021. Citado na página 12.
- FAN, A.; LEWIS, M.; DAUPHIN, Y. **Hierarchical Neural Story Generation**. 2018. Disponível em: <<https://arxiv.org/abs/1805.04833>>. Citado na página 13.
- FOUNDATION, P. S. **Python**. 2024. <<https://www.python.org/>>. [Online; acessado em 8-Outubro-2024]. Citado na página 10.
- GAO, Y.; XIONG, Y.; GAO, X.; JIA, K.; PAN, J.; BI, Y.; DAI, Y.; SUN, J.; WANG, M.; WANG, H. Retrieval-augmented generation for large language models: A survey. **arXiv preprint arXiv:2312.10997v5**, 2024. Citado 2 vezes nas páginas 9 e 12.
- GARLYAL, J. S.; HARIHARAN, B.; SINGH, A. K. An Analysis on Integrating Advanced Conversational AI in Legal Summarization and Information Retrieval . In: **2024 Second International Conference on Inventive Computing and Informatics (ICICI)**. Los Alamitos, CA, USA: IEEE Computer Society, 2024. p. 43–46. Disponível em: <<https://doi.ieeecomputersociety.org/10.1109/ICICI62254.2024.00016>>. Citado 3 vezes nas páginas 12, 16 e 17.

- HOLTZMAN, A.; BUYS, J.; DU, L.; FORBES, M.; CHOI, Y. **The Curious Case of Neural Text Degeneration**. 2020. Disponível em: <<https://arxiv.org/abs/1904.09751>>. Citado na página 13.
- IBM. **O que é engenharia de prompts?** 2025. <<https://www.ibm.com/br-pt/think/topics/prompt-engineering>>. [Online; acessado em 20-Março-2025]. Citado na página 12.
- IOANNIDIS JOSHUA HARPER, M. S. Q. D. H. J. Gracenote.ai: Legal generative ai for regulatory compliance. **Proceedings of the Third International Workshop on Artificial Intelligence and Intelligent Assistance for Legal Professionals in the Digital Workplace (LegalAIIA 2023)**, 2023. Citado na página 9.
- LIU, X.; LI, H.; ZHU, X. A gpt-based method of automated compliance checking through prompt engineering. 2023. Citado na página 17.
- LM Studio. **LM Studio - Run local large language models**. 2024. <<https://lmstudio.ai>>. Acesso em: 08 abr. 2025. Citado na página 14.
- _____. **LM Studio Community**. 2025. <<https://huggingface.co/lmstudio-community>>. [Online; acessado em 11-Abril-2024]. Citado na página 15.
- MACEY-DARE, R. How chatgpt and generative ai systems will revolutionize legal services and the legal profession. **SSRN**, 2023. Citado na página 9.
- MCCANN, B.; KESKAR, N. S.; XIONG, C.; SOCHER, R. **The Natural Language Decathlon: Multitask Learning as Question Answering**. 2018. Disponível em: <<https://arxiv.org/abs/1806.08730>>. Citado na página 13.
- META. **Llama 3.1**. 2024. <<https://ai.meta.com/blog/meta-llama-3-1/>>. [Online; acessado em 8-Outubro-2024]. Citado 2 vezes nas páginas 10 e 11.
- NAYAK, S. P.; PASUMARTHI, S.; RAJAGOPAL, B.; VERMA, A. K. Gdpr compliant chatgpt playground. In: **2024 International Conference on Emerging Technologies in Computer Science for Interdisciplinary Applications (ICETCS)**. [S.l.: s.n.], 2024. p. 1–6. Citado na página 12.
- TOUVRON, H.; MARTIN, L.; STONE, K. et al. **LLaMA: Open and Efficient Foundation Language Models**. 2023. <<https://arxiv.org/abs/2302.13971>>. Acesso em: 08 abr. 2025. Citado na página 14.
- WOLF, T.; DEBUT, L.; SANH, V.; CHAUMOND, J.; DELANGUE, C.; MOI, A.; CISTAC, P.; RAULT, T.; LOUF, R.; FUNTOWICZ, M.; DAVISON, J.; SHLEIFER, S.; PLATEN, P. von; MA, C.; JERNITE, Y.; PLU, J.; XU, C.; SCAO, T. L.; GUGGER, S.; DRAME, M.; LHOEST, Q.; RUSH, A. Transformers: State-of-the-art natural language processing. In: LIU, Q.; SCHLANGEN, D. (Ed.). **Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations**. Online: Association for Computational Linguistics, 2020. p. 38–45. Disponível em: <<https://aclanthology.org/2020.emnlp-demos.6/>>. Citado na página 14.