ESCOLA DE GUERRA NAVAL

CC (IM) MÁRCIO SELEMEN COELHO / C-Sup 2024

INTELIGÊNCIA ARTIFICIAL GENERATIVA: construção e avaliação de um modelo em apoio à Autoridade Marítima Brasileira

Rio de Janeiro
2024

CC (IM) MÁRCIO SELEMEN COELHO / C-Sup 2024

INTELIGÊNCIA ARTIFICIAL GENERATIVA: construção e avaliação de um modelo em apoio à Autoridade Marítima Brasileira

Monografia apresentada à Escola de Guerra Naval, como requisito parcial para a conclusão do Curso Superior.

Orientadora: CC (RM3-T) Ketia

Rio de Janeiro

Escola de Guerra Naval

2024

DECLARAÇÃO DA NÃO EXISTÊNCIA DE APROPRIAÇÃO INTELECTUAL IRREGULAR

Declaro que este trabalho acadêmico: a) corresponde ao resultado de investigação por mim desenvolvida, enquanto discente da Escola de Guerra Naval (EGN); b) é um trabalho original, ou seja, que não foi por mim anteriormente utilizado para fins acadêmicos ou quaisquer outros; c) é inédito, isto é, não foi ainda objeto de publicação; e d) é de minha integral e exclusiva autoria.

Declaro também que tenho ciência de que a utilização de ideias ou palavras de autoria de outrem, sem a devida identificação da fonte, e o uso de recursos de inteligência artificial no processo de escrita constituem grave falta ética, moral, legal e disciplinar. Ademais, assumo o compromisso de que este trabalho possa, a qualquer tempo, ser analisado para verificação de sua originalidade e ineditismo, por meio de ferramentas de detecção de similaridades ou por profissionais qualificados.

Assinatura digital gov.br

Documento assinado digitalmente
gov.br
MARCIO SELEMEN COELHO
Data: 18/11/2024 14:29:01-0300
Verifique em https://validar.iti.gov.br

RESUMO

INTELIGÊNCIA ARTIFICIAL GENERATIVA: construção e avaliação de um modelo em apoio à Autoridade Marítima Brasileira

Este trabalho apresenta o desenvolvimento de um modelo de inteligência artificial generativa com arquitetura de recuperação, ampliação e geração para a aplicação de normas da Autoridade Marítima Brasileira e legislações correlatas, a fim de determinar o seu desempenho. O estudo aborda a construção de uma base vetorial de dados a partir de normativos selecionados, a implementação de uma ferramenta utilizando um grande modelo de linguagem e a avaliação de sua performance. A metodologia técnica empregada envolveu a utilização de modelos do Google e da OpenAI, além de algoritmos e *frameworks* na linguagem de programação Python. O modelo desenvolvido foi avaliado utilizando questões de um concurso público da Marinha do Brasil para ingresso no quadro técnico na área de segurança do tráfego aquaviário, alcançando um grau de acurácia de 75%. Métricas adicionais como fidelidade da resposta, relevância da resposta e relevância do contexto foram também mensuradas. Os resultados alcançados indicam um desempenho robusto da ferramenta, comparável a *benchmarks* de modelos avançados no mercado. Não obstante os números promissores, o estudo discute ainda as limitações e desafios na implementação de uma inteligência artificial generativa, incluindo alucinações, vieses, questões de interpretabilidade, segurança e privacidade. Por fim, o trabalho conclui que o modelo desenvolvido representa um passo significativo para a consideração da aplicação dessa tecnologia emergente como ferramenta de apoio à autoridade marítima do Brasil, com potencial para automatizar consultas, agilizar interpretações normativas e aprimorar o treinamento de pessoal, contribuindo para garantir a conformidade regulatória e o fortalecimento do poder marítimo do Brasil.

Palavras-chave: Inteligência artificial generativa. Grandes modelos de linguagem. Recuperação, ampliação e geração. Normas da Autoridade Marítima Brasileira. Avaliação de desempenho.

ABSTRACT

GENERATIVE ARTIFICIAL INTELLIGENCE: construction and evaluation of a model in support of the Brazilian Maritime Authority

This work presents the development of a generative artificial intelligence model with a retrieval, augmentation, and generation architecture for the application of Brazilian Maritime Authority regulations and related legislation, to determine its performance. The study addresses the construction of a vector database from selected normative documents, the implementation of the tool using a large language model and the evaluation of its performance. The technical methodology employed involved the use of models from Google and OpenAI, as well as algorithms and frameworks in the Python programming language. The developed model was evaluated using questions from a public contest of the Brazilian Navy for entry into the technical staff in the area of water traffic safety, achieving an accuracy rate of 75%. Additional metrics such as response fidelity, response relevance and context relevance were also measured. The results achieved indicate a robust performance of the tool, comparable to benchmarks of advanced models on the market. Despite the promising numbers, the study also discusses limitations and challenges in the implementation of generative artificial intelligence, including hallucinations, biases, issues of interpretability, security and privacy. Finally, the work concludes that the developed model represents a significant step towards considering the application of this emerging technology as a support tool for the Brazilian Maritime Authority, with the potential to automate queries, speed up normative interpretations and improve personnel training, contributing to ensuring regulatory compliance and strengthening Brazil's maritime power.

Keywords: Generative artificial intelligence. Large language models. Retrieval-augmented generation. Brazilian Maritime Authority regulations. Performance evaluation.

SUMÁRIO

1 INTRODUÇÃO

A evolução da Inteligência Artificial (IA) é notável, abrangendo várias aplicações destinadas a aprimorar a segurança, a eficiência e a conformidade regulamentar na indústria marítima (Liu *et al.*, 2021). Dentre os avanços mais recentes, a Inteligência Artificial Generativa (IAG) emerge como um novo campo para exploração da IA. Especializada na geração de novos dados que podem replicar conteúdo gerado por humanos, a IAG evidencia seu potencial como um assistente de conhecimento (Adrian Cevallos *et al.*, 2023).

Nesse contexto, a implementação de modelos de IAG na indústria marítima sugere uma oportunidade de avanço na maneira como informações e dados normativos, como regulamentos e leis que a governam, possam ser processados e interpretados. Particularmente, no âmbito da Marinha do Brasil (MB), a utilização de uma IAG como ferramenta de apoio para busca, interpretação e implementação eficaz das regulamentações marítimas poderia revolucionar como a Autoridade Marítima Brasileira lida com seu arcabouço normativo.

Considerando o vasto volume de normas que regulam a atividade marítima e suas frequentes atualizações, a busca e interpretação manual de informações nesses documentos são atividades que demandam tempo e são propensas a erros. Tendo em vista essas dificuldades, a proposta desta pesquisa foi construir e avaliar a utilização de IAG como ferramenta de apoio para realizar tais tarefas nas Normas da Autoridade Marítima (NORMAM), Leis e Decretos correlatos.

Mais especificamente, o presente estudo teve como objetivo a construção de um modelo de IAG com arquitetura de Recuperação, Ampliação e Geração (RAG - *Retrieval Augmentation Generation*) para a aplicação de NORMAM e legislações correspondentes, a fim de determinar o seu desempenho.

Ao atingir esse objetivo, a pesquisa oferece à MB uma melhor compreensão sobre uma potencial ferramenta inovadora, capaz de otimizar processos, aumentar a assertividade, e melhorar a compreensão e aplicação de suas normas. Isso pode promover um desempenho mais eficiente, dinâmico e seguro dos profissionais da MB que operam sob a supervisão da Autoridade Marítima, contribuindo para garantir a conformidade com as normas regulatórias e reforçar a capacidade marítima do Brasil.

A fim de alcançar esse propósito, o trabalho foi orientado por teorias e conhecimentos relacionados à IA, mais especificamente à IAG, que abrange Grandes

Modelos de Linguagem (LLM), arquiteturas de RAG, e “loja de vetores”. No arcabouço do poder marítimo, o estudo será delimitado no tempo pelos documentos vigentes no ano de 2023, e no espaço pela aplicação específica na área de Segurança do Tráfego Aquaviário (STA).

Para a realização de todas as avaliações de desempenho e análises propostas, e viabilização dos cálculos e procedimentos, foi empregada uma abordagem metodológica técnica, utilizando algoritmos e *frameworks* na linguagem de programação Python. Nela, o *framework* LangChain foi empregado na construção da arquitetura RAG. Para a avaliação do desempenho do modelo, utilizou-se o *framework* RAGAS (RAG Assessment) com API da OpenAI (ChatGPT). O LLM a ser utilizado na pesquisa foi o do Google (modelo Gemini Pro) e, para a construção da “loja de vetores”, foi utilizado o modelo “embedding-001” (Google) e o banco de dados ChromaDB.

Quanto à organização da pesquisa, a estrutura encontra-se disposta da seguinte forma: inicialmente, a revisão da literatura apresenta os conceitos relacionados à IA, IAG e avaliação de modelos. Em seguida, a seção de desenvolvimento descreve a definição, construção e utilização da ferramenta, além de revelar a avaliação de seu desempenho. A seção de análise dos resultados, discute as métricas obtidas e versa sobre as limitações e desafios existentes. Por fim, a conclusão aborda as possíveis possibilidades de aplicações e as contribuições para a Marinha do Brasil.

2 REFERÊNCIAL TEÓRICO

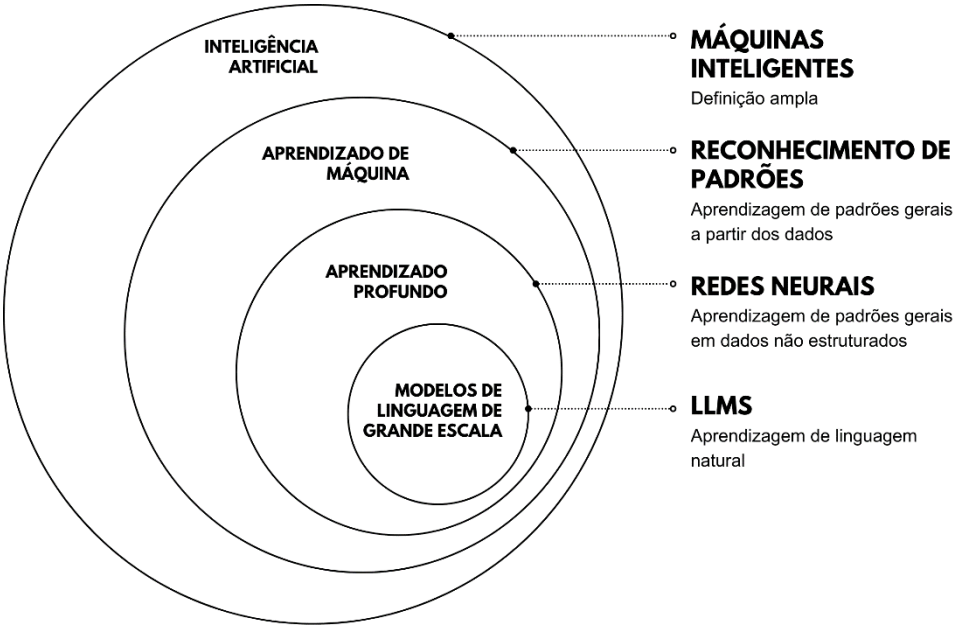
Nesta seção, será apresentado o referencial teórico que fundamenta a pesquisa. Inicialmente, serão explorados os conceitos e definições de Inteligência Artificial, abordando suas principais áreas e aplicações relevantes para o estudo. Em seguida, serão discutidos os Modelos de Linguagem de Grande Escala, com ênfase na arquitetura *Generative Pre-trained Transformer* (GPT) e suas implicações no Processamento de Linguagem Natural (PLN). Também serão examinadas as limitações dos LLM e as possíveis soluções, como o *Fine-tuning* e a Geração Aumentada por Recuperação, visando melhorar a adaptabilidade e precisão dos modelos. Finalmente, será abordado o processo de avaliação dos modelos sob a arquitetura de RAG, destacando a importância de técnicas como o RAGAS para assegurar a qualidade das respostas geradas.

2.1 INTELIGÊNCIA ARTIFICIAL

A Inteligência Artificial é uma disciplina da ciência da computação que se dedica ao desenvolvimento de sistemas capazes de executar tarefas que normalmente requerem inteligência humana, como reconhecimento de padrões e interpretação de linguagem (Mogali, 2014). De acordo com Agrawal, Tan e Rathore (2023), as definições de IA variam, mas geralmente envolvem a simulação por máquinas, especialmente computadores, de processos cognitivos humanos.

O termo IA é extremamente abrangente e engloba uma variedade de subcampos e tecnologias, conforme ilustrado na figura 1, a seguir:

Figura 1 - Inteligência Artificial



Fonte: Elaborada pelo autor (2024)

Entre essas subáreas, o Aprendizado de Máquina, ou *Machine Learning*, se destaca como um campo fundamental, em que algoritmos aprendem a partir de dados e fazem previsões sem intervenção humana direta (Janiesch, Zschech e Heinrich, 2021). Segundo os autores, esse campo é um dos pilares da IA, na qual decisões são tomadas por meio de uso dos dados, sem uma programação específica para cada tarefa.

Inserido no campo do Aprendizado de Máquina, encontra-se o Aprendizado Profundo, também conhecido como *Deep Learning*, que pode ser entendido como uma especialização dessa área. Nele, redes neurais complexas, com muitas camadas,

são utilizadas para descobrir padrões em grandes volumes de dados (Lecun, Bengio e Hinton, 2015). O Aprendizado Profundo tem desempenhado um papel revolucionário no avanço do PLN, especialmente após a introdução do conceito de mecanismos de atenção, com a arquitetura *Transformer* (Vaswani et. al., 2017).

Com base nessa arquitetura, os Modelos de Linguagem de Grande Escala, ou, de acordo com a tradução, *Large Language Models*, surgiram como uma evolução natural da IA, expandindo as fronteiras de interpretação e de geração de linguagem por máquinas. Atualmente, devido ao seu tamanho e poder de processamento, os LLMs são capazes de entender nuances complexas da linguagem e responder de maneira contextualizada, demonstrando avanços significativos em tarefas como diálogo automatizado, compreensão de texto e criação de conteúdo original (Naveed *et al.*, 2023).

Dada a sua relevância, a próxima seção focará nos LLMs. Esses modelos serão essenciais para esta pesquisa, permitindo o alcance dos objetivos propostos.

2.2 MODELOS DE LINGUAGEM DE GRANDE ESCALA

Os LLMs representam uma das mais avançadas formas de Inteligência Artificial, dedicadas a emular a capacidade humana de compreender e gerar linguagem. Segundo os autores Brown *et al.* (2020), esses modelos são descritos como "grandes" devido ao seu vasto número de “neurônios”, conhecidos como parâmetros, que podem chegar a bilhões em determinados casos, permitindo uma modelagem de linguagem excepcionalmente rica e complexa.

Um exemplo notável de LLM é o *Generative Pre-trained Transformer*, uma arquitetura que revolucionou o campo de PNL.

O termo "*Generative*", do GPT, indica que o modelo pode prever a próxima palavra em uma sequência, gerando texto coerente e contextualmente relevante baseado em seu treinamento. De acordo com os autores Yenduri *et al.* (2023), esse aspecto generativo é fundamental, pois permite que o GPT produza respostas e textos que fluem naturalmente, emulando o estilo e o conteúdo dos dados sobre os quais foi treinado.

A palavra "*Pre-trained*" é outra característica crucial do GPT, destacando que o modelo é inicialmente treinado em um vasto conjunto de dados textuais antes de ser especializado em tarefas específicas. Durante esta fase de pré-treinamento, o GPT

aprende uma rica variedade de padrões linguísticos e relações contextuais. Essa aprendizagem abrangente permite que o modelo identifique e reproduza a estrutura da linguagem, incluindo gramática, estilo e nuances semânticas, o que é essencial para suas capacidades de geração e compreensão de texto (Yenduri *et al.*, 2023).

O componente "*Transformer*", por sua vez, refere-se à arquitetura específica sobre a qual o GPT é construído. Introduzida no artigo de Vaswani et. al. (2017), "*Attention is All You Need*", a arquitetura *Transformer* utiliza mecanismos de atenção para modelar interdependências de longo alcance no texto, avaliando a importância relativa das palavras independentemente de sua posição. Isso permite ao modelo entender contextos complexos e manter a coerência em textos longos, beneficiando tarefas como resumo automático, tradução e geração de conteúdo.

Apesar das funcionalidades avançadas do GPT, uma limitação dessa arquitetura é que a capacidade de o modelo gerar respostas novas e precisas está limitada ao conteúdo e conhecimento incorporado durante o seu treinamento (Kandpal *et al.*, 2022). Dessa forma, os LLMs não têm a capacidade de acessar ou adquirir conhecimento fora desse escopo diretamente, o que pode ser um desafio em situações que requerem informações atualizadas ou altamente específicas (Soong *et al.*, 2023).

Para superar essa restrição, técnicas de *Fine-tuning* e *Retrieval-Augmented Generation* são frequentemente empregadas. Na língua portuguesa, "Ajuste Fino" e "Geração Aumentada por Recuperação", respectivamente, são técnicas que, segundo os autores Soong *et al.* (2023), oferecem soluções avançadas para melhorar a performance e adaptabilidade dos modelos de linguagem.

O Ajuste Fino (*Fine-tuning*) é uma técnica que ajusta os parâmetros de um modelo pré-treinado em um conjunto de dados específico, o que melhora significativamente a performance em tarefas particulares. Esse processo envolve treinar novamente o modelo com dados adicionais que são relevantes para a tarefa específica, permitindo que o modelo aprenda nuances e detalhes que não estavam presentes no treinamento inicial. Assim, ele torna o modelo mais especializado e preciso em contextos específicos (Lakatos, Pollner, Hajdu e Joó, 2024).

Por outro lado, a técnica de Geração Aumentada por Recuperação permite que o modelo consulte bases de dados externas para enriquecer suas respostas. Isso é feito combinando as capacidades generativas do modelo com informações recuperadas em tempo real, o que amplia o escopo de conhecimento do modelo

(Lakatos, Pollner, Hajdu e Joó, 2024). Por ser objeto deste estudo, a exploração mais detalhada deste método será iniciada a seguir.

2.3 GERAÇÃO AUMENTADA POR RECUPERAÇÃO

Os LLMs são capazes de responder perguntas sobre uma vasta gama de tópicos. No entanto, conforme exposto, um LLM isolado possui conhecimento limitado ao que foi incluído em seu treinamento. Isso significa que ele não pode acessar seus dados pessoais ou documentos proprietários, como aqueles encontrados em uma empresa e que não estão disponíveis na internet. Além disso, o conhecimento de um LLM não inclui informações ou artigos escritos após a data de seu treinamento (Ovadia, Brief, Mishaeli e Elisha, 2024).

Para superar essas limitações, técnicas como a RAG são empregadas para permitir que o modelo consulte bases de dados externas e atualizadas em tempo real, ampliando significativamente sua base de conhecimento. Essa abordagem torna o modelo mais flexível e adaptável a novos contextos e perguntas que podem surgir, aproveitando ao máximo os dados disponíveis (Huang e Huang, 2024).

2.3.1 Carregadores de documentos

De acordo com Huang e Huang (2024), o primeiro passo para criar uma aplicação RAG é carregar os dados recuperados em um formato adequado para manipulação, de modo que possam ser integrados ao contexto do modelo. Nesse sentido, os carregadores de documentos como, por exemplo, o disponível no *framework* LangChain, são fundamentais para esse processo (Auffarth, 2023).

Os carregadores de documentos são ferramentas que suportam diversos formatos de arquivos, como PDF (*Portable Document Format*), HTML (*HyperText Markup Language*) e JSON (*JavaScript Object Notation*), além de acessarem uma ampla gama de fontes de dados, incluindo websites e bancos de dados. Essas ferramentas têm a capacidade de processar tanto dados estruturados quanto não estruturados, o que expande significativamente sua aplicabilidade em diferentes contextos (Gao *et al.*, 2023).

O objetivo principal dos carregadores de documentos, conforme apontado por Auffarth (2023), é transformar essa diversidade de fontes, formatos e estruturas de

dados em um objeto de documento padrão. Importante destacar que esse objeto padrão é composto não apensa pelo conteúdo, mas também pelos metadados associados, o que facilita a organização e a recuperação de informações.

2.3.2 Divisão dos documentos em pedaços

Após o carregamento dos documentos em um formato padrão, é necessário dividi-los em partes menores, chamadas de *chunk*. Entretanto, embora essa pareça ser uma tarefa trivial, ela envolve algumas complexidades. Uma divisão simples baseada no comprimento, por exemplo, pode resultar em partes de uma sentença sendo separadas, o que impede a correta interpretação da informação. Portanto, conforme atenta Gao *et al.* (2023), é fundamental dividir os documentos de maneira a manter trechos semanticamente relevantes juntos.

Utilizando o *framework* LangChain, existem vários tipos diferentes de divisores de texto à disposição. Esses divisores variam em relação à maneira como dividem os *chunks*, os caracteres que consideram, e como medem seu o comprimento (Auffarth, 2023).

Além disso, Alan, Karaarslan e Aydin (2024) destacam que é importante haver alguma sobreposição nessa divisão, atribuindo um valor ao *overlap* para indicar quanto um *chunk* se sobrepõe ao anterior. Essa sobreposição garante que o LLM compreenda o conteúdo e mantenha conexões lógicas entre as partes do documento, criando uma noção de consistência. Outro aspecto importante na divisão em *chunks*, apontado por Ke *et al.* (2024), é a manutenção dos metadados.

2.3.3 Lojas de vetores e embeddings

Depois do carregamento e da divisão de documentos em pequenas partes semanticamente significativas, Gao *et al.* (2023) alerta que é crucial indexar esses trechos para facilitar sua recuperação quando necessário responder a perguntas sobre o conteúdo. Para alcançar esse objetivo, são utilizados os chamados *embeddings* e os vetores de armazenamento.

De acordo com Pilehvar e Camacho-Collados (2020), *embeddings* são representações numéricas de trechos de texto. Eles permitem que textos com conteúdo semelhantes possuam vetores similares em um espaço numérico. Isso

possibilita a comparação desses vetores para encontrar partes de texto que são semanticamente semelhantes.

Vetores de armazenamento, por sua vez, são bancos de dados que facilitam a busca por vetores semelhantes posteriormente. Os autores Ke *et al.* (2024) destacam que essa funcionalidade é fundamental para recuperar os “k” trechos mais semelhantes e utilizá-los, juntamente com uma pergunta, para obter respostas precisas de um LLM.

2.3.4 Busca por Similaridade

Após o carregamento, divisão em pedaços, vetorização e armazenamento dos trechos vetorizados dos documentos, a aplicação RAG está pronta para realizar a busca por similaridade. Também conhecida como busca semântica, esse processo permite identificar partes de texto semanticamente semelhantes por meio do uso de *embeddings*, permitindo comparações eficientes (Ke *et al.*, 2024).

Para realizar a busca por similaridade, utiliza-se um vetor de consulta para comparar com os vetores armazenados e identificar aqueles que estão mais próximos no espaço vetorial (Alan, Karaarslan e Aydin, 2024). De acordo com Moradi *et al.* (2024), uma das vantagens desse método de busca é a capacidade de recuperar informações relevantes mesmo quando a consulta não coincide exatamente com os documentos armazenados, mas compartilha semelhanças contextuais e semânticas

No entanto, os autores Es, James, Espinosa-Anke e Schockaert (2023) elucidam que existem algumas limitações na busca por similaridade. A própria definição de similaridade nessa técnica é frequentemente inconsistente e insatisfatória para algumas consultas, dificultando o controle e a previsibilidade dos resultados. Isso ocorre porque esse método, por vezes, pode recuperar textos genericamente semelhantes à descrição em vez de instâncias específicas, ou, ao contrário, concentra-se em detalhes específicos, o que não se alinha com buscas baseadas em descrições abstratas. Por esse motivo, dentre outros, tornar-se essencial a realização da avaliação dos modelos RAG.

2.4 AVALIAÇÃO DE LLMS SOB A ARQUITETURA RAG

A avaliação de LLMs sob a arquitetura de RAG é um campo emergente de

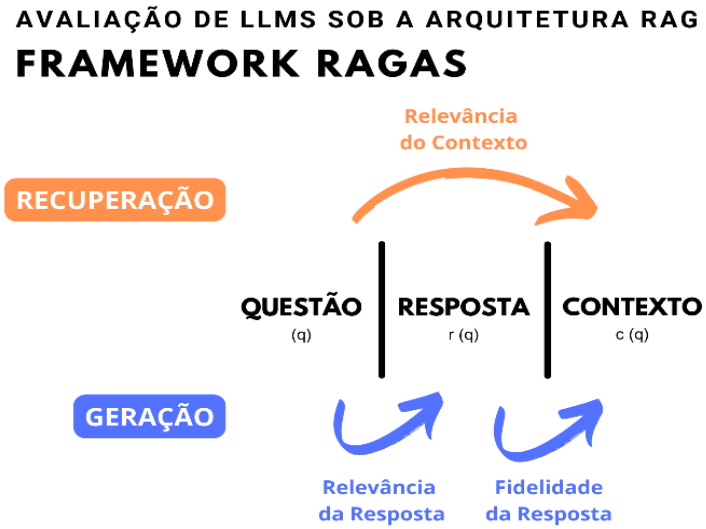
pesquisa que visa melhorar a precisão e a relevância das respostas geradas por esses modelos. Entretanto, segundo os autores Huang e Huang (2024), um dos principais desafios é garantir que a integração dos componentes de recuperação e geração funcione de maneira harmoniosa, sem introduzir vieses ou erros adicionais no processo.

Uma pesquisa sistemática realizada por Chen, Lin, Han e Sun (2024) revela que, embora a abordagem de RAG seja promissora, ainda existem obstáculos significativos em sua implementação eficaz. A análise do desempenho dessa arquitetura mostrou que, apesar de exibirem alguma robustez ao ruído de informações, os modelos ainda têm dificuldades consideráveis em rejeitar respostas quando a informação necessária não está presente, integrar informações de múltiplos documentos e lidar com informações falsas.

2.4.1 RAGAS

O RAGAS (*Retrieval Augmented Generation Assessment*) é um *framework* projetado para a avaliação de pipelines de modelos RAG. De acordo com Es, James, Espinosa-Anke e Schockaert (2023), o RAGAS propõe um conjunto de métricas que podem ser usadas sem a necessidade de anotações humanas, por meio de prompts em LLMs, focando em três aspectos principais: fidelidade da resposta, relevância da resposta e relevância do contexto, conforme ilustrado na figura 2, a seguir:

Figura 2 - RAGAS



Fonte: Elaborada pelo autor (2024)

A métrica Fidelidade refere-se à ideia de que a resposta (r(q)) deve estar fundamentada no contexto fornecido (c(q)), evitando alucinações e garantindo que o contexto recuperado possa servir como justificativa para a resposta gerada. A avaliação da fidelidade se dá por meio da decomposição de sentenças mais longas em afirmações mais curtas e focadas, verificando se tais afirmações podem ser inferidas do contexto (Es, James, Espinosa-Anke e Schockaert, 2023).

$$Fidelidade = \frac{N° \; de \; afirmações \; na \; resposta \; que \; podem \; ser \; inferidas \; do \; contexto}{N° \; total \; de \; afirmações \; na \; resposta \; gerada}$$

A relevância da resposta, por sua vez, diz respeito à medida em que a resposta gerada (r(q)) aborda a pergunta (q) de forma apropriada. De acordo com os autores, essa métrica é estimada gerando perguntas potenciais a partir da resposta dada e calculando a similaridade delas com a pergunta original.

$$Relevância \; da \; resposta = \frac{1}{N} \sum_{i=1}^{N} cos \left(E_{gi} , E_o \right)$$

Onde: E_{gi} = *embedding* da questão gerada i; E_o = *embedding* da questão original; e N = número de questões geradas (3, por padrão).

Por fim, a relevância do contexto destina-se a mensurar se o contexto recuperado (c(q)) é focado e contém o mínimo de informações irrelevantes possível. Essa métrica é avaliada pela extração de sentenças cruciais do contexto que são necessárias para responder à pergunta (Es, James, Espinosa-Anke e Schockaert, 2023).

$$Relevância \; do \; contexto = \frac{N° \; sentenças \; cruciais \; extraídas \; do \; contexto}{N° \; total \; de \; sentenças \; do \; contexto}$$

3 DESENVOLVIMENTO

Nesta seção, o desenvolvimento, que consiste na definição, construção, utilização e avaliação do modelo de IAG com arquitetura de RAG, será apresentado. Primeiramente, serão definidos os normativos da Autoridade Marítima Brasileira que serão objeto do estudo. Em seguida, será detalhado o processo de construção da base de dados, incluindo a ingestão dos documentos, extração de conteúdo e metadados, e outras etapas essenciais para a criação do índice semântico e

armazenamento dos vetores. Após, a utilização da ferramenta será demonstrada e, por fim, será realizada a avaliação do modelo, a fim de determinar o seu desempenho.

3.1 DEFINIÇÃO DOS NORMATIVOS DA AUTORIDADE MARÍTIMA BRASILEIRA

A definição dos normativos da Autoridade Marítima Brasileira envolveu a identificação e seleção dos documentos regulatórios que serão utilizados para a construção da base de dados. Neste estudo, os normativos escolhidos abrangeram as Normas da Autoridade Marítima (NORMAM), bem como leis e decretos que regulamentam a segurança do tráfego aquaviário no Brasil.

A seleção dos documentos foi realizada com base na bibliografia oficial do último concurso público (2023) para ingresso no quadro técnico da Marinha do Brasil, na área de segurança do tráfego aquaviário: CP-T/STA (2023). Tal escolha visou garantir que o modelo desenvolvido refletisse as práticas e exigências regulatórias mais atuais da autoridade marítima do Brasil. A lista completa desses documentos encontra-se no **Apêndice A**.

3.2 CONSTRUÇÃO DA BASE DE DADOS

Definidos os normativos, a base de dados foi construída com o intuito de aprimorar a precisão e a relevância das respostas geradas pelo modelo. A arquitetura apresentada na figura 3, a seguir, ilustra detalhadamente cada etapa desse processo de construção, desde a ingestão dos documentos até o armazenamento de partes do conteúdo normativo dos documentos em vetores.

Figura 3 - Construção da Base de dados

Fonte: Elaborada pelo autor (2024)

Na sequência, cada etapa da figura 3 encontra-se detalhada.

3.2.1 Ingestão dos normativos e extração de conteúdo e metadados

A primeira etapa da construção da base de dados consistiu na ingestão dos documentos selecionados. Esse processo foi precedido pela busca e organização das NORMAM, Leis e Decretos, que formarão a base do arcabouço normativo para o modelo, conforme elencado no item 3.1. Ao todo, 34 arquivos, no formato PDF, foram carregados, garantindo, assim, que a ferramenta tenha acesso a uma ampla gama de informações regulatórias.

Uma vez carregados, a próxima etapa correspondeu à extração do conteúdo textual e dos metadados associados. Os metadados incluem informações como título, autor, data da publicação e outros dados contextuais que serão essenciais para organizar e referenciar os documentos posteriormente, por ocasião da utilização da ferramenta.

3.2.2 Divisão em Pedaços (*chunks*)

Em uma terceira etapa, o conteúdo extraído dos normativos foi, então, dividido em pedaços menores, conhecidos como *chunks*. Cada *chunk* representará uma

porção lógica do texto, mantendo a coerência e a contextualização da informação.

Para a definição do tamanho de cada *chunk*, utilizam-se *tokens*. Um *token* é equivalente a cerca de 4 caracteres. Assim, 100 *tokens* equivalem a cerca de 60-80 palavras.

Neste estudo, utilizou-se um tamanho de *chunk* de 1.000 *tokens*, com uma sobreposição de 200 *tokens* entre pedaços consecutivos. Essa sobreposição, ou "*overlap",* garante a captura de informações nas extremidades dos pedaços de texto, melhorando a continuidade e a precisão das buscas semânticas. Além disso, é importante atentar que o tamanho dos *chunks* foi definido de forma a não exceder o "*Max Token Length*" do LLM a ser utilizado[1].

O código-fonte completo para a divisão dos *chunks*, contendo essas e as demais configurações relevantes, pode ser encontrado na seção "CHARLIE - Extração do texto dos normativos (PDFs) e divisão em *chuncks*", no **Apêndice B**.

3.2.3 Geração de *embeddings,* construção do índice semântico e armazenamento dos vetores

Na quarta etapa, cada *chunk* de texto foi convertido em um vetor de *embeddings*. Esses vetores são representações vetoriais do texto que capturam o significado semântico.

A transformação foi realizada utilizando um modelo de linguagem pré-treinada, também do Google, denominado *embedding-001*. Esse modelo foi capaz de mapear o texto em um espaço vetorial de alta dimensão, em que a proximidade entre vetores reflete a similaridade semântica entre os *chunks* de texto.

Os *embeddings* gerados foram, então, utilizados para construir um índice semântico. Essa quinta etapa envolveu técnicas de indexação que agruparam os *embeddings* semelhantes, otimizando a busca semântica, de forma a permitir que a ferramenta localize rapidamente informações relevantes em resposta às perguntas dos usuários.

A sexta e última etapa de construção da base de dados consistiu no armazenamento dos *embeddings* e do índice semântico em uma Loja de vetores.

---

[1] No caso, o modelo gemini-1.0-pro, do Google, que possui um limite de 32.760 tokens, assegurando, dessa forma, que cada pedaço de texto seja processável sem erros em uma única entrada. https://cloud.google.com/vertex-ai/generative-ai/docs/learn/models?hl=pt-br#gemini-models

Existem diversos banco de dados disponíveis para esse propósito, entretanto, nesta pesquisa, o ChromaDB foi o escolhido por ser gratuito e pela sua facilidade de integração com modelos de linguagem. A estrutura de dados armazenada no ChromaDB servirá como a base de dados principal para as buscas semânticas futuras, garantindo que a ferramenta possa ser utilizada para acessar informações de maneira rápida e eficiente.

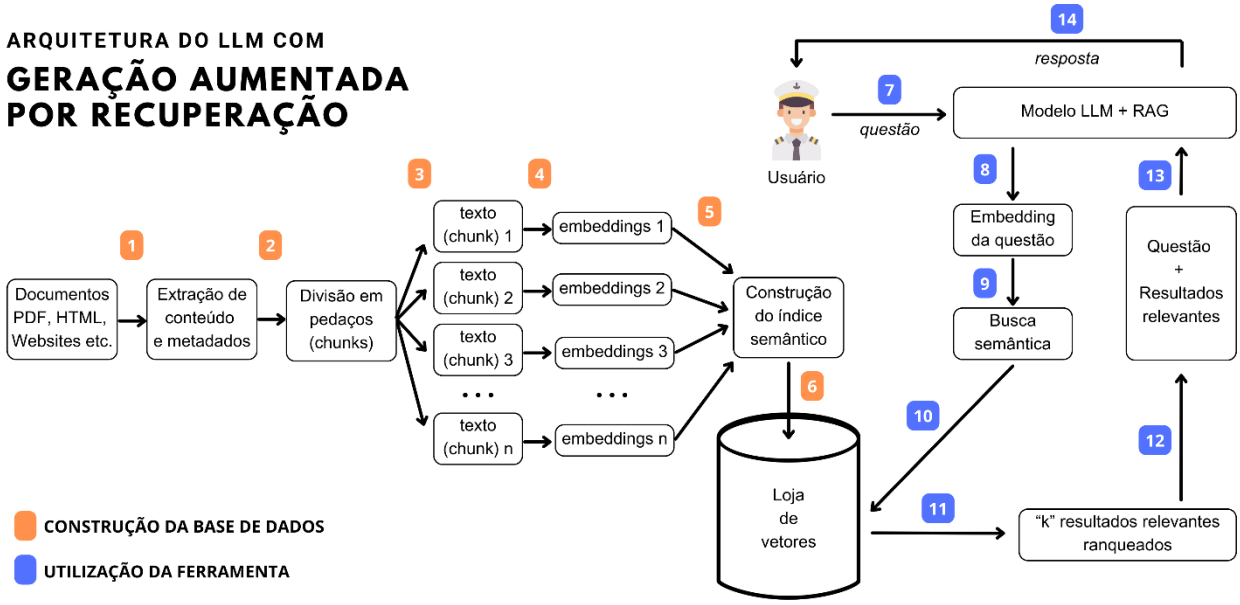
3.3 UTILIZAÇÃO DA FERRAMENTA

Após a construção da base de dados, a técnica de Geração Aumentada por Recuperação foi empregada em uma LLM para responder às perguntas dos usuários de forma eficiente e precisa.

Atualmente, existem diversos LLM que poderiam ter sido utilizados com o *framework* LangChain a fim de cumprir esse propósito. Entretanto, neste estudo, optou-se pelo modelo Gemini (*gemini-1.0-pro*) do Google, por apresentar excelente desempenho e permitir a utilização de sua API (*Application Programming Interface*) de forma gratuita.

A arquitetura, detalhada na Figura 4, apresenta um fluxo sistemático de como a ferramenta processará uma questão e irá gerar uma resposta relevante.

Figura 4 - Utilização da Ferramenta



Fonte: Elaborada pelo autor (2024)

3.3.1 Formulação da questão pelo usuário e o *embedding* da questão

A utilização da ferramenta inicia-se quando um usuário, representado por um Oficial da Marinha na ilustração, formula uma questão e submete ao "Modelo LLM + RAG". Essa questão pode envolver qualquer aspecto dos normativos da Autoridade Marítima Brasileira previamente inseridos na base de dados.

Em uma próxima etapa, a questão do usuário é transformada em uma representação vetorial denominada "*embedding* da questão". É importante destacar que essa transformação vetorial deve ser gerada por meio do mesmo modelo utilizado previamente por ocasião da geração dos *chunks,* qual seja, o modelo do Google *embedding-001,* conforme exposto no tópico 3.2.4. A utilização desse mesmo modelo assegurou que as representações vetoriais fossem compatíveis e que a busca semântica funcionasse perfeitamente.

3.3.2 Busca semântica e retorno dos resultados relevantes

Com o "*embedding* da questão" pronto, a etapa seguinte consiste na busca semântica na Loja de Vetores (ChromaDB), que contém os *embeddings* previamente gerados dos *chunks* de texto dos documentos normativos.

Essa busca por similaridade vetorial permitirá à ferramenta encontrar partes dos documentos semanticamente similares à questão formulada, mesmo que palavras exatas não coincidam. Ela irá recuperar, portanto, informações contextualmente relevantes, que ajudarão a aperfeiçoar a qualidade das respostas geradas.

A busca semântica retornará, então, uma lista de "k" resultados relevantes, ranqueados de acordo com sua similaridade semântica com a questão do usuário.

Na seção "ECHO - Carregamento da Loja de Vetores criada no Google Drive", do código-fonte (**Apêndice B**), encontra-se a definição do valor de "k" igual a 5 (cinco); ou seja, nesta pesquisa, cinco resultados semanticamente relevantes serão recuperados e utilizados na próxima etapa de geração da resposta.

3.3.3 Geração da Resposta

Utilizando os resultados relevantes recuperados, a ferramenta finalmente combinará os dados retornados com a questão apresentada e se valerá da própria

capacidade de geração de texto do LLM para elaborar uma resposta coesa e informativa. As configurações detalhadas de como essa resposta deve ser moldada também podem ser consultadas no **Apêndice B**.

Por exemplo, na seção "FOXTROT - Configuração do modelo Gemini Pro (LLM) + RAG", encontra-se estabelecida o parâmetro relativo à “temperatura” do LLM. Temperaturas mais baixas são adequadas para *prompts* que exigem uma resposta mais determinística ou menos aberta, enquanto temperaturas mais altas podem levar a resultados mais diversos ou criativos. Uma temperatura de 0 (zero), como a configurada neste estudo, é determinística, o que significa que a resposta de maior probabilidade sempre será a selecionada.

Outro exemplo de configuração que merece destaque é o “*template”* a ser utilizado para a elaboração das respostas. A sua escolha impacta diretamente a formatação e a estrutura das respostas geradas. Dessa forma, para a finalidade desta pesquisa, o estabelecimento das regras contidas nessa seção FOXTROT do código-fonte foram fundamentais para o alcance dos resultados pretendidos.

Diante de tais configurações, a resposta final gerada pelo modelo poderia ser, então, simplesmente apresentada ao usuário, completando o ciclo de utilização da ferramenta. Entretanto, conforme pode-se observar a partir da seção GOLF do algoritmo, as respostas do modelo deste estudo foram gravadas em uma planilha do Google *Sheets,* a fim de possibilitar a execução da próxima fase, que diz respeito à avaliação do modelo criado.

3.4 AVALIAÇÃO DO MODELO

Completado o ciclo de construção da base de dados e utilização da ferramenta, torna-se fundamental avaliar o desempenho do modelo desenvolvido. Para tal, o modelo LLM com RAG foi submetido à resolução das questões da prova do CP-T/STA (2023).

A opção pela avaliação por meio de questões do último concurso representa uma aferição fidedigna e justa do modelo, tendo em vista que a base de dados foi construída com base na bibliografia do certame desse mesmo ano de 2023. Os links com as perguntas e o gabarito da prova encontram-se disponíveis no **Apêndice C**.

Os métodos de avaliação empregados incluíram, em um primeiro momento, a comparação das respostas geradas pelo modelo com o gabarito oficial para o cálculo

do grau de acurácia. Em uma segunda etapa, utilizou-se o *framework* RAGAS para medir a fidelidade e a relevância das respostas obtidas em relação aos documentos normativos, bem como a aplicação de técnicas de similaridade semântica para determinar a relevância do contexto recuperado.

Os resultados dessas avaliações encontram-se apresentados a seguir, em uma escala de 0 a 100%, com valores mais altos indicando melhor desempenho.

3.4.1 Grau de acurácia

O grau de acurácia (GA) foi a métrica fundamental utilizada para avaliar o nível de precisão do modelo. Essa avaliação consistiu em resolver as questões das provas mencionadas.

A ferramenta produziu respostas para cada uma das questões, indicando a opção correta (A, B, C, D ou E). Em seguida, compararam-se as respostas geradas pelo modelo com o gabarito oficial da prova. Dessa forma, foi possível avaliar a precisão das respostas, identificando quais estavam corretas e quais não atenderam ao gabarito.

Os resultados da avaliação de acurácia encontram-se expostos adiante em diferentes seções. Essa divisão foi realizada para permitir uma análise específica do desempenho do modelo em diferentes contextos e perspectivas.

3.4.1.1 CP-T/STA (2023) – Modelo LLM com RAG

Esta primeira seção aborda a avaliação do GA por meio da resolução direta das questões do CP-T/STA (2023). Assim sendo, do último concurso, foram submetidas para avaliação da ferramenta 48 questões de múltipla escolha, excluídas as 2 anuladas.

Como resultado, o modelo desenvolvido acertou 36 questões, errando 8 e informando “Não sei a resposta” em 4. O GA do modelo LLM com utilização da arquitetura de RAG foi, portanto, igual a 75%.

3.4.1.2 CP-T/STA (2023) – Modelo LLM sem RAG x Modelo LLM com RAG

Nesta segunda seção, foi realizada uma avaliação comparativa entre o

desempenho do modelo LLM com a utilização da arquitetura de RAG (desenvolvido nesta pesquisa) e o mesmo modelo LLM mas sem a arquitetura de RAG. Isto é, o modelo Gemini do Google, que se encontra disponível para uso do público em geral, com dados treinados até fevereiro de 2023, mas sem a capacidade de recuperar informações da Loja de vetores. Para efeito dessa comparação, foram utilizadas as mesmas 48 questões (excluídas as 2 anuladas), do concurso de 2023.

Diante desse contexto, o modelo LLM sem RAG respondeu de maneira bem menos precisa, obtendo um GA igual a 46%. É interessante também observar que 35% das questões foram acertadas exclusivamente pelo modelo LLM com RAG.

3.4.1.3 CP-T/STA (2023) por categoria de questão

Nesta terceira seção, o desempenho do modelo LLM com RAG foi avaliado separadamente nas diferentes categorias de questões presentes nas provas.

As categorias foram identificadas e padronizadas automaticamente por meio de um outro LLM, que analisou e classificou as 48 questões de acordo com a seguinte distribuição: 20 questões conceituais, 11 questões procedimentais, 7 questões temporais e 10 questões de responsabilidade.

Como resultado, o modelo mostrou variações na acurácia conforme cada categoria. Nas questões conceituais, o GA foi de 65%. Nas questões procedimentais, foi de 100%. Para as questões temporais, o GA foi de 57%, e nas questões de responsabilidade, foi de 80%.

3.4.1.4 CP-T/STA (2023) por normativo

Esta última seção de avaliação do GA aborda o desempenho do modelo separadamente com base nos normativos que fundamentam as questões, isto é, que serviram de base para a elaboração das questões de múltipla escolha.

De igual maneira, os normativos foram identificados e padronizadas automaticamente por meio de um outro LLM, que analisou e classificou as 48 questões de acordo com a seguinte distribuição: 35 questões fundamentadas pelas NORMAM e 13 questões baseadas em Leis ou Decretos. Então, foi realizada uma avaliação do GA do modelo para cada grupo de normativos.

Como resultado, o modelo obteve um GA de 74% nas questões fundamentadas

pelas NORMAM, enquanto nas questões baseadas em Leis ou Decretos o GA foi de 77%.

Todas essas comparações diretas com o gabarito permitiram uma avaliação clara e objetiva da performance da ferramenta. No entanto, a fim de assegurar que a precisão das respostas não foi algo meramente acidental, ou seja, não foram produtos dos popularmente conhecidos “chutes”, mas sim do resultado de uma compreensão sólida e coerente dos normativos pelo modelo, outras métricas foram também avaliadas por intermédio do *framework* RAGAS.

3.4.2 RAGAS

O RAGAS facilita a avaliação automatizada de modelos RAG, aplicando *prompts* específicos para calcular suas métricas. Com isso, ele elimina a necessidade de anotações manuais e torna o processo de avaliação automatizado por meio de outro modelo de LLM, qual seja, o ChatGPT, da OpenAI.

De maneira semelhante à avaliação do GA do tópico anterior, o cálculo das métricas por meio do RAGAS também consistiu em resolver as questões da prova do CP-T/STA (2023). Entretanto, com duas principais diferenças.

A primeira delas é que somente o enunciado das questões foram utilizados como *input* ao modelo. Assim, a ferramenta forneceu as respostas apenas com base no conhecimento recuperado da “Loja de vetores”, sem ter tido acesso às opções (A, B, C, D ou E) das perguntas da prova.

A segunda é que, devido ao fato de a API da OpenAI ser paga, diferentemente da API gratuita do Google (Gemini) utilizada na seção anterior, a abrangência das avaliações realizadas não foi tão extensa. Dados os custos envolvidos, o cálculo das métricas por meio do RAGAS foi executado apenas para quatro questões, uma para cada categoria padronizada, conforme classificação do tópico 3.4.1.3.

Nesse sentido, os seguintes números das questões do CP-T/STA (2023) foram selecionados para avaliação pelos RAGAS: questão de nº 27 (conceitual), questão de nº 25 (procedimental), questão de nº 45 (temporal) e questão de nº 09 (de responsabilidade). A escolha dessas questões se justifica por serem perguntas interrogativas diretas, ou seja, que não dependiam das alternativas para o modelo oferecer uma resposta.

Diante do enunciado dessas questões, o modelo forneceu as respectivas

respostas, que foram, então, avaliadas sob três critérios: fidelidade da resposta, relevância da resposta e relevância do contexto.

3.4.2.1 Fidelidade da resposta

A fidelidade da resposta (FR) refere-se à precisão com que o modelo gera respostas baseadas nas informações recuperadas. Essa métrica assegura que as respostas fornecidas pelo modelo estão alinhadas com os dados normativos e regulatórios disponíveis na base de dados.

Os resultados de FR do modelo foram de 50% (questão nº 09) e de 100% (três demais questões).

3.4.2.2 Relevância da resposta

A relevância da resposta (RR) avalia a adequação das respostas geradas pelo modelo em relação às perguntas formuladas. Essa avaliação verifica se o conteúdo das respostas é pertinente e endereça de forma direta e clara às questões apresentadas.

Os valores obtidos de RR para o modelo encontraram-se entre 80% e 89%.

3.4.2.3 Relevância do contexto

Diferentemente das duas métricas anteriores, que focam na geração das respostas, a relevância do contexto (RC) tem como objetivo mensurar a capacidade de recuperação do contexto. Essa avaliação mede a eficácia do modelo em recuperar informações contextualmente pertinentes a partir da base de dados.

As métricas apuradas de RC para o modelo variaram entre 5% e 25%.

Os detalhes completos das questões utilizadas, respostas fornecidas pela ferramenta, contextos recuperados e os respectivos resultados podem ser encontrados no **Apêndice D**.

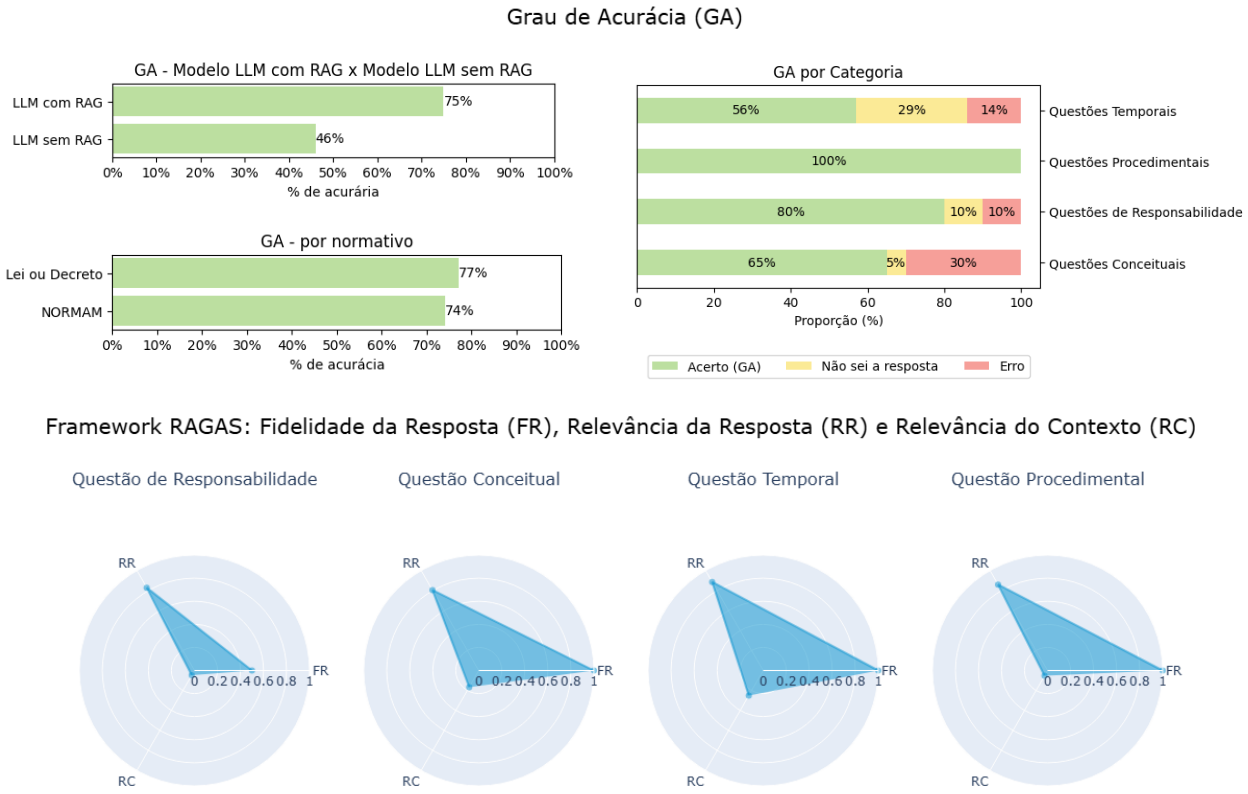
4 ANÁLISE DOS RESULTADOS

Nesta seção, será apresentada a análise dos resultados obtidos com a implementação do modelo de IAG utilizando a arquitetura de RAG para a resolução das questões do CP-T/STA (2023). Essa análise encontra-se dividida em duas partes principais. A primeira, consolida as métricas de avaliação a fim de discutir o desempenho do modelo. A segunda, aborda as limitações e os desafios de sua implementação.

4.1 DESEMPENHO DO MODELO

Conforme exposto no tópico 3.4, o desempenho do modelo foi avaliado com base em diversas métricas. Na figura 5, abaixo, encontra-se a consolidação dos resultados alcançados.

Figura 5 - Resultados consolidados da avaliação de desempenho do modelo



Fonte: Elaborada pelo autor (2024)

O modelo LLM com RAG desenvolvido nesta pesquisa atingiu um GA (Grau de Acurácia) de 75%. Esse valor foi obtido por meio da média dos acertos da ferramenta nas quatro categorias de questões propostas. Tal índice indica um desempenho robusto em termos de acurácia do modelo e posiciona-o em um patamar comparável a valores de *benchmarks* atingidos por LLM poderosas disponíveis no mercado.

Por exemplo, no *benchmark* MMLU (*Massive Multitask Language Understanding*), que consiste em um conjunto de testes projetados para medir o desempenho de modelos em tarefas envolvendo texto e imagens, incluindo compreensão de leitura, matemática de nível universitário e questionários de múltipla escolha em áreas como física, economia e ciências sociais, o LLM Gemini Pro atingiu um nível de 79,13%. O Gemini Ultra, o LLM mais avançado do Google até o momento deste estudo, alcançou 90,04% nesse *benchmark*, o GPT-4 (OpenAI), 87,29% e o GPT-3.5 (OpenAI), 70% (GOOGLE, 2024). Portanto, um resultado de GA de 75% do modelo desta pesquisa é digno de nota.

O valor de GA obtido também se mostra expressivo quando comparado com os resultados conquistados pelos candidatos no concurso CP-T/STA (2023). Na relação final de aprovados desse certame, a média das notas da prova objetiva dos 12 candidatos aprovados e classificados, equivalentes a três vezes o número de vagas conforme o subitem 6.2.8 do Edital, foi de 81,50% (BRASIL, 2024). Essa comparação evidencia a eficácia do modelo proposto em alcançar resultados próximos aos dos candidatos mais bem-sucedidos nesse concurso.

Na análise comparativa entre o modelo desenvolvido utilizando RAG e o modelo sem utilização dessa arquitetura, houve uma diferença notável em termos de GA. O modelo LLM sem RAG alcançou uma acurácia de apenas 46%, o que representa uma diferença de 29 pontos percentuais em relação ao modelo LLM com RAG proposto por este trabalho. Essa vantagem em termos de acurácia para o modelo LLM com RAG na tarefa avaliada destaca a sua superioridade, demonstrando a importância dessa arquitetura na melhoria do desempenho dos modelos de linguagem.

Ainda em relação ao desempenho do GA, apesar de ter havido variações nas métricas por categoria de questão, não foram observadas diferenças significativas de desempenho entre os diferentes normativos (NORMAM, Leis e Decretos). Tal resultado sugere que o modelo desenvolvido neste estudo consegue manter uma consistência no tratamento dos diversos tipos de assuntos.

No que tange às avaliações por meio do *framework* RAGAS, os resultados

revelaram valores satisfatórios em relação à capacidade de geração de texto. Mais especificamente, os valores alcançados pela métrica de FR (fidelidade da resposta) mostraram que, na maioria dos casos, as respostas fornecidas pela ferramenta foram completamente fiéis às informações normativas recuperadas, garantindo sua precisão em relação aos documentos de origem. A métrica RR (relevância da resposta), por sua vez, foi avaliada como altamente adequada às perguntas formuladas, indicando que as respostas não apenas foram corretas, mas também relevantes e contextualmente apropriadas.

Por outro lado, os resultados relativos à capacidade de recuperação do contexto da base de dados não se mostraram tão expressivos. Os valores obtidos em relação à métrica RC (relevância do contexto) indicaram que houve variações significativas na adequação das informações contextuais para diferentes perguntas. Além disso, o percentual de conteúdo relevante recuperado como contexto foi consideravelmente baixo.

Os resultados dessa última métrica foram importantes pois revelaram a necessidade de experimentos com outros LLM e de testes com diferentes configurações de parâmetros, como, por exemplo, variados tamanhos de *chunk,* de *overlaps* e de quantidade (k) de relevantes *chunks* recuperados*,* a fim de otimizar o componente de recuperação do contexto do modelo. Ademais, ele traz à tona que, em que pese os resultados, de forma geral, terem sido positivos, limitações e desafios na sua implementação existem e necessitam ser apresentados e considerados.

4.2 LIMITAÇÕES E DESAFIOS

De acordo com um estudo recente da renomada empresa de pesquisa e consultoria Gartner (2024), a principal barreira enfrentada pelas organizações ao adotar soluções de IAG é a dificuldade em estimar e demonstrar o valor dessas tecnologias. O estudo revelou que 49% das empresas têm dificuldades em mensurar o real impacto e os benefícios tangíveis que essas soluções podem oferecer.

Nesse contexto, a construção e avaliação de um modelo de IAG com arquitetura de RAG, ora desenvolvido por este estudo, consiste no primeiro passo na tentativa de transpor essa relevante barreira. No entanto, em se tratando de modelos de linguagem, diversas outras limitações e desafios ainda persistirão, como: a ocorrência de alucinações, a natureza não determinística dos modelos, vieses,

questões de interpretabilidade, preocupações com segurança e privacidade, restrições das amostras utilizadas nas avaliações, além da rápida evolução da tecnologia e da própria relevância do RAG no futuro. Isso para citar as principais, que passarão a ser brevemente elucidadas a seguir.

A começar pelas alucinações, elas ocorrem quando um LLM gera informações que parecem plausíveis, mas são incorretas ou irrelevantes. Essas ocorrências podem comprometer a confiabilidade das respostas e exige estratégias robustas para mitigá-las. Não por acaso, a palavra *hallucinate* foi selecionada como a Palavra do Ano de 2023 pelo Dicionário Cambridge, tamanha ter sido sua repercussão com a popularização dos modelos de linguagem nessa temporada (CAMBRIDGE, 2024).

Outro desafio que os modelos de linguagem apresentam é a sua natureza não determinística, o que significa que as respostas podem variar para cada pergunta. Essa característica pode ser bastante problemática em contextos que exigem consistência e previsibilidade nas respostas, como o abordado por esta pesquisa.

Adicionalmente, as respostas dos modelos também podem apresentar vieses, a depender dos dados utilizados no treinamento do LLM. Tais vieses podem refletir ou amplificar preconceitos presentes nos dados, levando a respostas tendenciosas e inadequadas em contextos sensíveis.

A interpretabilidade dos modelos é, igualmente, outro desafio dos LLM, já que os modelos funcionam como uma "caixa preta". De forma semelhante à nossa falta de conhecimento sobre o funcionamento de cada neurônio do cérebro humano, também não é possível entender exatamente como cada “neurônio” em uma rede neural do modelo funciona. Essa falta de transparência pode representar uma limitação para a confiança e adoção dessas tecnologias.

Questões de segurança e privacidade também se apresentam como grandes desafios, especialmente no que se refere ao vazamento de informações sensíveis, reservadas e confidenciais. Contudo, dada a evolução da tecnologia, há soluções que podem ser implementadas para mitigar esses riscos e garantir que as informações sejam protegidas, por meio, por exemplo, da configuração dos denominados *GuardRails*.

Uma outra limitação, que se aplica diretamente a esta pesquisa, diz respeito à limitada quantidade de questões e métricas utilizadas para determinar o desempenho de um modelo. Elas podem não capturar a complexidade e diversidade dos desafios encontrados em contextos reais. Isso indica que os resultados de avaliação, como os

apresentados no tópico anterior, por exemplo, podem não ser generalizáveis a todos os contextos possíveis.

Além disso, há de se ressaltar, ainda, que as tecnologias utilizadas têm suas próprias limitações e estão em constante evolução. Existem modelos mais avançados e métodos mais elaborados que aqueles utilizados por este estudo. Uma ilustração disso é a existência de abordagens de recuperação de contexto, como o método “Busca Híbrida”, que superam a “Busca Semântica” utilizada por esta pesquisa. Essa abordagem combina a busca semântica com métodos tradicionais, melhorando a precisão e a relevância dos resultados.

Por fim, soma-se a tudo isso discussões sobre a relevância da própria arquitetura de RAG no futuro. Atualmente, os LLM podem gerenciar contextos extremamente longos e essa capacidade tende a crescer à medida que a tecnologia evolui, questionando a necessidade do RAG. Por outro lado, defende-se que a arquitetura ainda é essencial para a melhoraria da eficiência operacional e a precisão nas respostas.

5 CONCLUSÃO

O trabalho desenvolvido neste estudo teve como propósito a utilização de IAG como ferramenta de apoio para busca, interpretação e implementação de normativos da Autoridade Marítima Brasileira. Mais especificamente, a pesquisa teve como objetivo a construção de um modelo com arquitetura de RAG para a aplicação de NORMAM e legislações correlatas, a fim de determinar o seu desempenho.

Para atingir esse objetivo, o estudo recorreu à literatura existente sobre Inteligência Artificial e Modelos de Linguagem de Grande Escala. Após, foram abordadas todas as etapas essenciais necessárias para o desenvolvimento da ferramenta, iniciando pela definição dos normativos, passando pela construção da base de dados e do modelo, e culminando na sua avaliação. O trabalho, então, consolidou os valores das métricas obtidas e discutiu as limitações e desafios para sua implementação.

Os resultados apresentados pelo modelo desenvolvido foram considerados robustos e destacaram-se, ainda, as possibilidades de melhorias e refinamentos. Entretanto, ressaltou-se que as possíveis repercussões devem ser encaradas com cautela. Por se tratar de modelos de linguagem, pontuaram-se as principais limitações

e desafios que ainda precisam ser superados pela tecnologia para tornar possível a plena utilização da IAG no ecossistema do poder marítimo.

A título de contribuição, sugere-se que a presente pesquisa seja encarada como um passo significativo para a consideração da aplicação da IAG como ferramenta de apoio à autoridade marítima do Brasil. No futuro, com a evolução dos LLM em curso, vislumbra-se que os modelos com a arquitetura de RAG poderiam ser utilizados para automatizar e agilizar o processo de consulta e interpretação das NORMAM e legislações correlatas, oferecendo suporte ao pessoal no cumprimento de suas funções.

Além disso, avista-se que os modelos poderiam ser também adaptados para o treinamento de novos Oficiais e Praças da Marinha, proporcionando um meio eficiente de aprendizado e revisão dos regulamentos. Sua capacidade de gerar respostas precisas e relevantes poderia, de igual modo, ser aplicada na gestão de incidentes e na tomada de decisões estratégicas, garantindo a conformidade com as normas e regulamentos vigentes.

Na verdade, as possibilidades de casos de uso serão limitadas somente pela criatividade da mente humana. Ademais, à medida que a tecnologia evolui, certamente surgirão novas oportunidades de aplicação que poderão transformar profundamente a forma de atuação da autoridade marítima no país. Tais inovações poderão resultar em um desempenho mais eficiente, ágil e seguro dos profissionais da MB que exercem suas funções sob a jurisdição dessa autoridade naval, contribuindo para garantir a conformidade regulatória e o fortalecimento do poder marítimo do Brasil.

REFERÊNCIAS

ALAN, Ahmet Yusuf, KARAARSLAN, Enis e AYDIN, Omer. **A RAG-based Question Answering System Proposal for Understanding Islam: MufassirQAS LLM**. SSRN Electronic Journal, 2024.

AGRAWAL, Palaash, TAN, Cheston e RATHORE, Heena. **Advancing Perception in Artificial Intelligence through Principles of Cognitive Science**. 2023. Disponível em: http://arxiv.org/abs/2310.08803.

AUFFARTH, Ben. **Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT, and other LLMs**. Birmingham, England: Packt Publishing, 2023.

BRASIL. Marinha do Brasil. **Serviço de Seleção do Pessoal da Marinha**. Disponível em: https://www.inscricao.marinha.mil.br/marinha/NotasPOCP-T-2023.pdf?id_file=7725. Acesso em: 5 jul. 2024.

BROWN, Tom B. *et al.* **Language Models are Few-Shot Learners**. 2020. Disponível em: http://arxiv.org/abs/2005.14165.

CAMBRIDGE University Press & Assessment. **'Hallucinate' is Cambridge Dictionary's Word of the Year 2023**. Disponível em: https://www.cambridge.org/news-and-insights/hallucinate-is-cambridge-word-of-the-year-2023. Acesso em: 09 jul. 2024.

CEVALLOS, Adrian *et al.* **Tech Report Generative AI**. 2023. https://doi.org/10.18235/0005105. Acesso em: 21 mar. 2024.

CHEN, J. *et al.* **Benchmarking large language models in retrieval-Augmented Generation**. 2023. Disponível em: http://arxiv.org/abs/2309.01431.

ES, Shahul *et al.* **RAGAS: Automated Evaluation of Retrieval Augmented Generation**. 2023. Disponível em: http://arxiv.org/abs/2309.15217.

GAO, Yunfan *et al.* **Retrieval-Augmented Generation for Large Language Models: A Survey**. 2023. Disponível em: http://arxiv.org/abs/2312.10997.

GARTNER. **Gartner Survey Finds Generative AI Is Now the Most Frequently Deployed AI Solution in Organizations**. 2024. Disponível em: https://www.gartner.com/en/newsroom/press-releases/2024-05-07-gartner-survey-finds-generative-ai-is-now-the-most-frequently-deployed-ai-solution-in-organizations. Acesso em: 08 jul. 2024.

GOOGLE. **Gemini: A Family of Highly Capable Multimodal Models**. 2024. Disponível em: https://storage.googleapis.com/deepmind-media/gemini/gemini_1_report.pdf . Acesso em: 5 jul. 2024.

HUANG, Yizheng e HUANG, Jimmy. **A Survey on Retrieval-Augmented Text**

Generation for Large Language Models. 2024. Disponível em: http://arxiv.org/abs/2404.10981.

JANIESCH, Christian e ZSCHECH, Patrick e HEINRICH, Kai. **Machine learning and deep learning**. Electronic Markets, v. 31, n. 3, p. 685–695, 2021.

KANDPAL, Nikhil *et al.* **Large Language Models Struggle to Learn Long-Tail Knowledge**. 2022. Disponível em: http://arxiv.org/abs/2211.08411.

KE, Yu He *et al.* **Development and Testing of Retrieval Augmented Generation in Large Language Models-A Case Study Report**. [s.l: s.n.].

LAKATOS, Robert *et al.* **Investigating the performance of Retrieval-Augmented Generation and fine-tuning for the development of AI-driven knowledge-based systems**. 2024. Disponível em: http://arxiv.org/abs/2403.09727.

LECUN, Y.; BENGIO, Y.; HINTON, G. **Deep learning. Nature**. Maio: Nature Publishing Group, 2015. v. 27

LIU, Ryan Wen *et al.* **An enhanced CNN-enabled learning method for promoting ship detection in maritime surveillance system**. Maritime Safety. 2021.

MOGALI, S.; SHIVAYOGI E, M.; SHIVARANJINI, S. **Artificial Intelligence and its applications in Libraries**. [s.l: s.n.].

MORADI, M. *et al.* **Exploring the landscape of large language models: Foundations, techniques, and challenges**. 2024. Disponível em: http://arxiv.org/abs/2404.11973.

NAVEED, Humza *et al.* **A Comprehensive Overview of Large Language Models**. 2023. Disponível em: http://arxiv.org/abs/2307.06435.

OVADIA, Oded *et al.* **Fine-Tuning or Retrieval? Comparing Knowledge Injection in LLMs**. 2023. Disponível em: http://arxiv.org/abs/2312.05934.

PILEHVAR, M. T.; CAMACHO-COLLADOS, J. Embeddings in natural language processing: Theory and advances in vector representations of meaning. **Synthesis lectures on human language technologies**, v. 13, n. 4, p. 1–175, 2020.

SOONG, David *et al.* **Improving accuracy of GPT-3/4 results on biomedical data using a retrieval-augmented language model**. 2023. Disponível em: http://arxiv.org/abs/2305.17116.

VASWANI, Ashish *et al.* **Attention Is All You Need**. 2017. Disponível em: http://arxiv.org/abs/1706.03762.

YENDURI, Gokul *et al.* **Generative Pre-trained Transformer: A Comprehensive Review on Enabling Technologies, Potential Applications, Emerging Challenges, and Future Directions**. 2023. Disponível em: http://arxiv.org/abs/2305.10435.

APÊNDICE A – Bibliografia CP-T/STA (2023)

Lista de documentos de acordo com a relação constante na bibliografia do último concurso público (2023) para ingresso no quadro técnico (CP-T), na área de segurança do tráfego aquaviário (STA), da Marinha do Brasil (MB).

Disponível em: https://www.marinha.mil.br/dpc/bibliografia/qtsta

1. Lei nº 9.537, de 11 de dezembro de 1997
2. Lei nº 9.966, de 28 de abril de 2000
3. Lei nº 7.203, de 3 de julho de 1984
4. Lei nº 7.273, de 10 de dezembro de 1984
5. Lei nº 7.573, de 23 de dezembro de 1986
6. Lei nº 2.180, de 3 de fevereiro de 1954
7. Lei nº 9.432, de 8 de janeiro de 1997
8. Lei nº 12.815, de 5 de junho de 2013
9. Decreto nº 2.596, de 18 de maio de 1998
10. Decreto nº 4.136, de 20 de fevereiro de 2002
11. Decreto nº 8.033, de 27 de julho de 2013
12. Decreto nº 94.536, de 29 de junho de 1987
13. Decreto nº 1.530, de 22 de junho de 1995
14. NORMAM-01-DPC
15. NORMAM-02-DPC
16. NORMAM-03-DPC
17. NORMAM-04-DPC
18. NORMAM-05-DPC
19. NORMAM-06-DPC
20. NORMAM-07-DPC
21. NORMAM-08-DPC
22. NORMAM-09-DPC
23. NORMAM-10-DPC
24. NORMAM-11-DPC
25. NORMAM-13-DPC
26. NORMAM-15-DPC
27. NORMAM-17-DHN
28. NORMAM-20-DPC
29. NORMAM-24-DPC
30. NORMAN-25-DHN
31. NORMAM-26-DHN
32. NORMAM-30-DPC
33. NORMAM-33-DPC
34. NORMAM-34-DPC

APÊNDICE B - Código-fonte do Modelo (Google Colab)

ALFA - Instalação das Bibliotecas

```
!pip install -q google-generativeai==0.6.0
!pip install -q langchain-google-genai==1.0.3
!pip install -U langchain
!pip install -U langchain-community
!pip install -q pypdf
!pip install -q chromadb
!pip install -q gdown
!pip install -q pandas
!pip install -q ragas
```

BRAVO - Importação dos Módulos, Chave da API e Montagem do Google Drive

```
# Módulos de temporizador e de manipulação de dados
import time
import pandas as pd

# Modulos do Google
import google.generativeai as genai
from google.colab import userdata

# Configurar chave de acesso da API do Google
GOOGLE_API_KEY=userdata.get('GOOGLE_API_KEY')
genai.configure(api_key=GOOGLE_API_KEY)

# Módulos do LangChain
from langchain import PromptTemplate
from langchain.chains.question_answering import load_qa_chain
from langchain.document_loaders import PyPDFLoader, PyPDFDirectoryLoader
from langchain.text splitter import RecursiveCharacterTextSplitter
from langchain.vectorstores import Chroma
from langchain.chains import RetrievalQA
from langchain_google_genai import ChatGoogleGenerativeAI
from langchain_google_genai import GoogleGenerativeAIEmbeddings

# Módulos e função para melhor exibição dos outputs
from IPython.display import display
from IPython.display import Markdown

# Carregar e montar o Google Drive
from google.colab import drive
drive.mount('/content/drive')
```

CHARLIE - Extração do texto dos normativos (PDFs) e divisão em chuncks

```
# Carregar a pasta com os PDFs
pdf_loader = PyPDFDirectoryLoader("/content/drive/MyDrive/NORMAM Leis e Decretos/")

# Dividir o contéudo em pedaços
pages = pdf_loader.load_and_split()

# Divisão em chuncks
text_splitter = RecursiveCharacterTextSplitter(chunk_size=1000, chunk_overlap=200)
context = "\n\n".join(str(p.page_content) for p in pages)
texts = text_splitter.split_text(context)
```

DELTA - Criação dos Embeddings e da Loja de Vetores

```
# Criação dos embeddings
Embeddings = GoogleGenerativeAIEmbeddings(model="models/embedding-001",
google_api_key=GOOGLE_API_KEY)

# Criação de um diretório para armazenar a Loja de Vetores no Google Drive
directory = '/content/drive/MyDrive/vector_store/NORMAM_Leis_Decretos_vector_store'
```

ECHO - Carregamento da Loja de Vetores criada no Google Drive

```
# Carregamento da Loja de Vetores no Google Drive
```

```
persistent_directory = '/content/drive/MyDrive/vector_store/NORMAM_Leis_Decretos_vector_store'
embeddings = GoogleGenerativeAIEmbeddings(model="models/embedding-001",
google_api_key=GOOGLE_API_KEY)

vector_index = Chroma(persist_directory=persistent_directory,
embedding_function=embeddings).as_retriever(search_type="similarity", search_kwargs={"k":5})
```

FOXTROT - Configuração do modelo Gemini Pro (LLM) + RAG

```
# Inicialização do modelo Gemini Pro do Google
model = ChatGoogleGenerativeAI(model="gemini-pro",
                               google_api_key=GOOGLE_API_KEY,
                               temperature=0,
                               convert_system_message_to_human=True,
                               )

# Configuração do modelo
template = """
Você é um assistente útil e usará o contexto fornecido para responder corretamente à pergunta
de uma prova.

Você não deverá usar nenhuma outra informação além desse contexto para responder à pergunta.
Não tente fazer deduções ou inferências, nem tente advinhar a resposta correta.
Ou seja, se você não conseguir encontrar a resposta correta, voce deve informar o seguinte
texto "Não sei a resposta".

As perguntas e respostas serão na seguinte linguagem: português do Brasil.

Na resposta, somente a letra correspondente à resposta deve ser exibida.
Isto é, não deve haver outras informações da resposta.

Contexto: {context}
Questão: {question}
Resposta:
"""
QA_CHAIN_PROMPT = PromptTemplate.from_template(template)
qa_chain = RetrievalQA.from_chain_type(
    model,
    retriever=vector_index,
    return_source_documents=True,
    chain_type_kwargs={"prompt": QA_CHAIN_PROMPT}
)
```

GOLF - Configuração da planilha do Google Sheets

```
# Autenticar com as credenciais do Google
from google.colab import auth
auth.authenticate_user()

import gspread
from google.auth import default
creds, _ = default()

gc = gspread.authorize(creds)

# Abrir a planilha do Google Sheets
worksheet_title = "Questoes"
worksheet = gc.open(worksheet_title).sheet1

# Ler todos os valores da aba e pegar os índices das colunas
all_values = worksheet.get_all_values()
headers = all_values[0]

# Pegar o índice da coluna com o conteúdo das questões
conteudo_index = headers.index("conteudo")
```

HOTEL - Gerar as respostas e recuperar os contextos por meio do modelo com RAG, gravando os resultados nas colunas [resposta_llm_com_rag] e [contextos]

```
# Pegar o índice das colunas
resposta_llm_com_rag_index = headers.index("resposta_llm_com_rag")
contextos_index = headers.index("contextos")

# Função para gerar as respostas e o recuperar os contextos utilizados
def get_answer_and_contexts(enunciado_questao):
```

```
  try:
    result = qa_chain({"query": enunciado_questao})
    answer_text = result["result"]
    contexts_list = []
    contexts_list = [str(doc) for doc in result["source_documents"]]
    contexts_list = "\n\n\n".join(contexts_list)
    return answer_text, contexts_list
  except Exception as e:
    print(f'{type(e).__name__}: {e}')
    return f'{type(e).__name__}: {e}'

# Escolher a linha da planilha que deseja comecar e terminar a iteracao (Número mais a
esquerda, do Google Sheet)
linha_inicio_plan = 2
linha_final_plan = 51

# Ajustar a linha escolhida para o index correspondente
linha_inicio_plan -= 1

# Iterar por cada linha a partir da segunda linha (índice 1)
for row in all_values[linha_inicio_plan:linha_final_plan]:
  try:
    # Armazenar Conteúdo da questão
    enunciado_questao = row[conteudo_index]

    if enunciado_questao != 'ANULADA':
      # Obter a resposta e o contexto através da função e exibir
      resposta_llm_com_rag, contextos = get_answer_and_contexts(enunciado_questao)
      print("Resposta:", resposta_llm_com_rag)
      print("Contextos:", contextos)

      # Atualizar a coluna 'resposta_llm_com_rag' e 'contextos' na linha atual com a resposta
      worksheet.update_cell(all_values.index(row) + 1, resposta_llm_com_rag_index + 1,
resposta_llm_com_rag)
      worksheet.update_cell(all_values.index(row) + 1, contextos_index + 1, contextos)
    else:
      # Atualizar a coluna 'resposta_llm_com_rag' e 'contextos' na linha atual com a '-'
      worksheet.update_cell(all_values.index(row) + 1, resposta_llm_com_rag_index + 1, '-')
      worksheet.update_cell(all_values.index(row) + 1, contextos_index + 1, '-')

  except:
    # Atualizar a coluna 'resposta_llm_com_rag' e 'contextos' na linha atual com a 'Erro'
    worksheet.update_cell(all_values.index(row) + 1, resposta_llm_com_rag_index + 1, 'Erro')
    worksheet.update_cell(all_values.index(row) + 1, contextos_index + 1, 'Erro')

  # Aguardar até a próxima requisição
  time.sleep(3)

print("Fim.")
```

INDIA - Gerar as respostas por meio do modelo LLM sem RAG, gravando os resultados na coluna [resposta_llm_sem_rag]

```
# Carregar o modelo Gemini Pro (puro, sem RAG)
model_llm_sem_rag = genai.GenerativeModel('gemini-pro')

def generate_response(conteudo_questao):
  response = model_llm_sem_rag.generate_content(f"""
  Você é um assistente útil e deverá responder corretamente à pergunta de uma prova.

  Não tente fazer deduções ou inferências, nem tente advinhar a resposta correta.
  Ou seja, se você não conseguir encontrar a resposta correta, voce deve informar o seguinte
texto "Não sei a resposta".

  As perguntas e respostas serão na seguinte linguagem: português do Brasil.

  Na resposta, somente a letra correspondente à resposta deve ser exibida.
  Isto é, não deve haver outras informações da resposta.

  Questão: {conteudo_questao}

  Resposta:
  """, stream=True)
  try:
    response.resolve()
    print(response.text)
    return response.text
```

```
  except Exception as e:
    print(f'{type(e).__name__}: {e}')
    text_match = re.search(r'text: "(.*?)"', str(e))
    text_content = text_match.group(1) if text_match else None
    return text_content

# Escolher a linha da planilha que deseja comecar e terminar a iteracao (Numero mais a
esquerda, do Google Sheet)
linha_inicio_plan = 2
linha_final_plan = 51

# Calculo para ajustar a linha escolhida para o index correspondente
linha_inicio_plan -= 1

# Pegar o índice da coluna
resposta_llm_sem_rag_index = headers.index("resposta_llm_sem_rag")

# Iterar por cada linha a partir da segunda linha (índice 1)
for row in all_values[linha_inicio_plan:linha_final_plan]:
  try:
    # Armazenar Conteúdo da questão
    conteudo_questao = row[conteudo_index]

    if conteudo_questao != 'ANULADA':
      # Obter a resposta através da função do modelo e exibir
      resposta_llm_sem_rag = generate_response(conteudo_questao)

       # Atualizar a coluna 'resposta_llm_sem_rag' na linha atual com a resposta
      worksheet.update_cell(all_values.index(row) + 1, resposta_llm_sem_rag_index + 1,
resposta_llm_sem_rag)
    else:
      # Atualizar a coluna 'resposta_llm_sem_rag' na linha atual com "-"
      worksheet.update_cell(all_values.index(row) + 1, resposta_llm_sem_rag_index + 1, '-')

  except Exception as e:
    # Atualizar a coluna 'resposta_llm_sem_rag' na linha atual com "Erro"
    worksheet.update_cell(all_values.index(row) + 1, resposta_llm_sem_rag_index + 1, 'Erro')
    print(f'Erro: {str(e)}')

  # Aguardar até a próxima requisição
  time.sleep(3)

print("Fim.")
```

JULIETT - Preencher a coluna [categoria] da planilha do Google Sheets, baseado no conteúdo da questão

```
# Carregar o modelo Gemini Pro (puro, sem RAG)
model_llm_sem_rag = genai.GenerativeModel('gemini-pro')

def generate_norma(conteudo_questao):
  response = model_llm_sem_rag.generate_content(f"""
  Você é um assistente útil e usará o contexto fornecido para categorizar corretamente uma
pergunta/afirmação.

  Exemplos de categoria:
  - Questão de definição
  - Questão de procedimento
  - Questão de prazo
  - Questão de responsabilidade

  Não se limite a essas categorias, podem haver outras.

  Agora, com base na pergunta/afirmação a seguir, proponha uma categoria: {conteudo_questao}

  """, stream=True)
  response.resolve()
  print(response.text)
  return response.text

# Escolher a linha da planilha que deseja comecar e terminar a iteracao (Numero mais a
esquerda, do Google Sheet)
linha_inicio_plan = 2
linha_final_plan = 51

# Calculo para ajustar a linha escolhida para o index correspondente
linha_inicio_plan -= 1
```

```
# Pegar o índice da coluna
categoria_index = headers.index("categoria")

# Iterar por cada linha a partir da segunda linha (índice 1)
for row in all_values[linha_inicio_plan:linha_final_plan]:
  try:
    # Armazenar Conteúdo da questão
    conteudo_questao = row[conteudo_index]

    if conteudo_questao != 'ANULADA':
      # Pegar a "categoria" por meio da função criada
      categoria = generate_norma(conteudo_questao)

      # Atualizar a coluna 'categoria' na linha atual com a resposta
      worksheet.update_cell(all_values.index(row) + 1, categoria_index + 1, categoria)
    else:
      # Atualizar a coluna 'categoria' na linha atual com "-"
      worksheet.update_cell(all_values.index(row) + 1, categoria_index + 1, '-')

  except Exception as e:
    # Atualizar a coluna 'categoria' na linha atual com "Erro"
    worksheet.update_cell(all_values.index(row) + 1, categoria_index + 1, 'Erro')
    print(f'Erro: {str(e)}')

  # Aguardar até a próxima requisição
  time.sleep(3)

print("Fim.")
```

KILO - Preencher a coluna [categoria_padronizada] da planilha do Google Sheets, baseado na coluna [categoria]

```
# Carregar o modelo Gemini Pro (puro, sem RAG)
model_llm_sem_rag = genai.GenerativeModel('gemini-pro')

# Função para padronizar uma categoria específica
def categorize_with_llm(categoria, categorias_padronizadas):
    response = model_llm_sem_rag.generate_content(f"""
    Você é um assistente útil e usará o contexto fornecido para categorizar corretamente uma
pergunta/afirmação.

    Com base na categoria fornecida: {categoria}, escolha a categoria padronizada mais
apropriada a partir desta lista: {categorias_padronizadas}

    """, stream=True)
    response.resolve()
    print(response.text)
    return response.text.strip()

# Função para gerar uma categoria padronizada com base em várias categorias
def generate_padronizada(categorias):
    response = model_llm_sem_rag.generate_content(f"""
    Você é um assistente útil e usará o contexto fornecido para categorizar corretamente uma
pergunta/afirmação.

    As categorias identificadas são as seguintes: {categorias}

    Agrupe-as em categorias padronizadas mais genéricas, considerando que se tratam de
questões de Concurso Público para ingresso no Quadro Técnico da Marinha do Brasil, na área de
Segurança do Tráfego Aquaviário.
    Ao final, proponha uma lista com os nomes dessas categorias padronizadas.
    A resposta deve ser uma lista separada por vírgulas.

    """, stream=True)
    response.resolve()
    print(response.text)
    return response.text.split('\n')

# Escolher a linha da planilha que deseja comecar e terminar a iteracao (Numero mais a
esquerda, do Google Sheet)
linha_inicio_plan = 2
linha_final_plan = 51

# Calculo para ajustar a linha escolhida para o index correspondente
linha_inicio_plan -= 1
```

```
# Pegar o índice da coluna
categoria_index = headers.index("categoria")

# Pegar todas as categorias existentes na planilha
categorias_existentes = [row[categoria_index] for row in
all_values[linha_inicio_plan:linha_final_plan] if row[categoria_index] not in ('-', 'Erro')]

# Gerar categorias padronizadas
categorias_padronizadas = generate_padronizada(categorias_existentes)

# Pegar o índice da coluna
categoria_padronizada_index = headers.index("categoria_padronizada")

# Iterar por cada linha a partir da linha de início até a linha de fim
for row in all_values[linha_inicio_plan:linha_final_plan]:
    try:
        # Pegar a categoria existente na linha atual
        categoria_existente = row[categoria_index]

        if categoria_existente != '-' and categoria_existente != 'Erro':
            # Categorizar com a LLM
            categoria_padronizada = categorize_with_llm(categoria_existente,
categorias_padronizadas)

            # Atualizar a coluna 'categoria_padronizada' na linha atual com a categoria
padronizada
            worksheet.update_cell(all_values.index(row) + 1, categoria_padronizada_index + 1,
categoria_padronizada)
        else:
            # Atualizar a coluna 'categoria_padronizada' na linha atual com "-"
            worksheet.update_cell(all_values.index(row) + 1, categoria_padronizada_index + 1,
'-')

    except Exception as e:
        # Atualizar a coluna 'categoria padronizada' na linha atual com "Erro"
        worksheet.update_cell(all_values.index(row) + 1, categoria_padronizada_index + 1,
'Erro')
        print(f'Erro: {str(e)}')

    # Aguardar até a próxima requisição
    time.sleep(3)

print("Padronização de categorias concluída.")
```

LIMA - Preencher a coluna [norma] da planilha no Google Sheets, baseado no conteúdo da questão

```
# Carregar o modelo Gemini Pro (puro, sem RAG)
model_llm_sem_rag = genai.GenerativeModel('gemini-pro')

def generate_norma(conteudo_questao):
  response = model_llm_sem_rag.generate_content(f"""
  Você é um assistente útil e usará o contexto fornecido para retornar corretamente a NORMAM,
Lei ou Decreto a que uma pergunta/afirmação se refere.

  Exemplo de pergunta/afirmação 1: "Em conformidade com a NORMAM-02/DPC, os produtos perigosos
são divididos em classes de acordo com suas características. Desse modo, assinale a opção
correta:"
  Resposta esperada: "NORMAM-02/DPC"

  Exemplo de pergunta/afirmação 2: "De acordo com o art.22 da lei nº 9.966/2000, qualquer
incidente ocorrido em portos organizados, instalações portuárias, dutos, navios, plataformas e
suas instalações de apoio, que possa provocar poluição das águas sob jurisdição nacional,"
  Resposta esperada: "Lei nº 9.966/2000"

  Exemplo de pergunta/afirmação 3: "A NORMAM-10/DPC versa sobre as Normas da Autoridade
Marítima:"
  Resposta esperada: "NORMAM-10/DPC"

  Agora, com base no início do texto da pergunta/afirmação a seguir, extraia a qual norma ela
se refere: {conteudo_questao}

  Você deve tentar enquadrar a sua resposta em uma das 34 normas enumeradas a seguir.
  Entretanto, se não for possível extrair a norma pelo conteúdo da questão, não tente fazer
deduções ou inferências.
  Isto é, se a norma não estiver explícita na questão, você deve retornar "Verificar norma
manualmente".

  Normas:
```

```
  1. Lei nº 9.537, de 11 de dezembro de 1997
  2. Lei nº 9.966, de 28 de abril de 2000
  3. Lei nº 7.203, de 3 de julho de 1984
  4. Lei nº 7.273, de 10 de dezembro de 1984
  5. Lei nº 7.573, de 23 de dezembro de 1986
  6. Lei nº 2.180, de 3 de fevereiro de 1954
  7. Lei nº 9.432, de 8 de janeiro de 1997
  8. Lei nº 12.815, de 5 de junho de 2013
  9. Decreto nº 2.596, de 18 de maio de 1998
  10. Decreto nº 4.136, de 20 de fevereiro de 2002
  11. Decreto nº 8.033, de 27 de julho de 2013
  12. Decreto nº 94.536, de 29 de junho de 1987
  13. Decreto nº 1.530, de 22 de junho de 1995
  14. NORMAM-01-DPC
  15. NORMAM-02-DPC
  16. NORMAM-03-DPC
  17. NORMAM-04-DPC
  18. NORMAM-05-DPC
  19. NORMAM-06-DPC
  20. NORMAM-07-DPC
  21. NORMAM-08-DPC
  22. NORMAM-09-DPC
  23. NORMAM-10-DPC
  24. NORMAM-11-DPC
  25. NORMAM-13-DPC
  26. NORMAM-15-DPC
  27. NORMAM-17-DHN
  28. NORMAM-20-DPC
  29. NORMAM-24-DPC
  30. NORMAN-25-DHN
  31. NORMAM-26-DHN
  32. NORMAM-30-DPC
  33. NORMAM-33-DPC
  34. NORMAM-34-DPC

  Após o enquadramento, você deve retornar como reposta somente uma das duas opões: "Lei ou
Decreto" ou "NORMAM", a depender da norma.
  Exemplo de resposta para Lei nº 7.273, de 10 de dezembro de 1984: "Lei ou Decreto"
  Exemplo de resposta para Decreto nº 94.536, de 29 de junho de 1987: "Lei ou Decreto"
  Exemplo de resposta para NORMAM-11-DPC: NORMAM"

  """, stream=True)
  response.resolve()
  print(response.text)
  return response.text

# Escolher a linha da planilha que deseja comecar e terminar a iteracao (Numero mais a
esquerda, do Google Sheet)
linha_inicio_plan = 4
linha_final_plan = 51

# Calculo para ajustar a linha escolhida para o index correspondente
linha_inicio_plan -= 1

# Pegar o índice da coluna
norma_index = headers.index("norma")

# Iterar por cada linha a partir da segunda linha (índice 1)
for row in all_values[linha_inicio_plan:linha_final_plan]:
  try:
    # Armazenar Conteúdo da questão
    conteudo_questao = row[conteudo_index]

    if conteudo_questao != 'ANULADA':
      # Pegar a "norma" por meio da função criada
      norma = generate_norma(conteudo_questao)

      # Atualizar a coluna 'norma' na linha atual com a resposta
      worksheet.update_cell(all_values.index(row) + 1, norma_index + 1, norma)
    else:
      # Atualizar a coluna 'norma' na linha atual com "-"
      worksheet.update_cell(all_values.index(row) + 1, norma_index + 1, '-')

  except Exception as e:
    # Atualizar a coluna 'norma' na linha atual com "Erro"
    worksheet.update_cell(all_values.index(row) + 1, norma_index + 1, 'Erro')
    print(f'Erro: {str(e)}')
```

```
  # Aguardar até a próxima requisição
  time.sleep(3)

print("Fim.")
```

MIKE - Framework RAGAS - Responder às perguntas, mas sem considerar as opções das questões

```
template = """
Você é um assistente útil e usará o contexto fornecido para responder corretamente à pergunta
de uma prova.

Você não deverá usar nenhuma outra informação além desse contexto para responder à pergunta.
Não tente fazer deduções ou inferências, nem tente advinhar a resposta correta.
Ou seja, se você não conseguir encontrar a resposta correta, voce deve informar o seguinte
texto "Não sei a resposta".

As perguntas e respostas serão na seguinte linguagem: português do Brasil.

Contexto: {context}
Questão: {question}
Resposta:
"""
QA_CHAIN_PROMPT = PromptTemplate.from_template(template)# Run chain
qa_chain = RetrievalQA.from_chain_type(
    model,
    retriever=vector_index,
    return_source_documents=True,
    chain_type_kwargs={"prompt": QA_CHAIN_PROMPT}
)

# Função para gerar a resposta e recuperar os contextos
def get_answer_and_contexts(enunciado_questao):
  try:
    result = qa_chain({"query": enunciado_questao})
    answer_text = result["result"]
    contexts_list = []
    contexts_list = [str(doc) for doc in result["source_documents"]]
    contexts_list = "\n\n\n".join(contexts_list)
    return answer_text, contexts_list
  except Exception as e:
    print(f'{type(e).__name__}: {e}')
    return f'{type(e).__name__}: {e}'

# Autenticar com as credenciais do Google
from google.colab import auth
auth.authenticate_user()

import gspread
from google.auth import default
creds, _ = default()

gc = gspread.authorize(creds)

# Abrir a planilha do Google Sheets, aba "RAGAS"
worksheet_title = "Questoes"
worksheet = gc.open(worksheet_title).worksheet("RAGAS")

# Ler todos os valores da aba e pegar os índices das colunas
all_values = worksheet.get_all_values()
headers = all_values[0]

# Pegar os índices das colunas
enunciado index = headers.index("question")
answer_index = headers.index("answer")
answer_contexts_index = headers.index("contexts")

# Escolher a linha da planilha que deseja comecar e terminar a iteracao (Numero mais a
esquerda, do Google Sheet)
linha inicio plan = 2
linha_final_plan = 5

# Calculo para ajustar a linha escolhida para o index correspondente
linha_inicio_plan -= 1

for row in all_values[linha_inicio_plan:linha_final_plan]:
  try:
    # Pegar a questão
```

```
    enunciado_questao = row[enunciado_index]

    # Obter a resposta do modelo e os contextos
    answer_text, contexts_list = get_answer_and_contexts(enunciado_questao)
    print("answer_text:", answer_text)
    print("contexts_list:", contexts_list)

    # Preencher 'answer' e 'answer_contexts' colunas na planilha
    worksheet.update_cell(all_values.index(row) + 1, answer_index + 1, answer_text)
    worksheet.update_cell(all_values.index(row) + 1, answer_contexts_index + 1, contexts_list)

  except:
    # Atualizar a coluna 'answer' e 'answer_contexts' na linha atual com a 'Erro'
    worksheet.update_cell(all_values.index(row) + 1, answer_index + 1, 'Erro')
    worksheet.update_cell(all_values.index(row) + 1, answer_contexts_index + 1, 'Erro')
```

NOVEMBER - Framework RAGAS - Avaliação do modelo

```
# Importação dos módulos e requisição da chave de acesso da API da OpenAI
import os
import getpass
open_ai_key = getpass.getpass('Enter your OPENAI API Key')
os.environ['OPENAI_API_KEY'] = open_ai_key

# Ler a planilha do Google Sheets, aba "RAGAS"
worksheet_title = "Questoes"
worksheet = gc.open(worksheet_title).worksheet("RAGAS")
data = worksheet.get_all_values()

# Gravar os dados da planilha em um dataframe
df_ragas = pd.DataFrame(data[1:], columns=data[0])
df = df_ragas[['question', 'answer', 'contexts']]

# Função para organizar os contextos recuperados em uma lista de listas
def extract_page_content(text):
    # Split the text by 'page_content=' and ignore the first segment
    segments = text.split("page_content=")[1:]

    # Remove leading and trailing quotes
    segments = [segment.strip().strip("'") for segment in segments]

    # Further split each segment by whitespace to get lists of strings
    flat_list = [item for segment in segments for item in segment.split('/n/n/n')]
    return flat_list

# Extrair os dados das colunas [questions], [answers], e [contexts] e gravar nas respectivas
variáveis
questions = df['question'].tolist()
answers = df['answer'].tolist()
contexts = df['contexts'].apply(extract_page_content).tolist()

# Transformar os dados do dataframe em um dataset, a ser utilizado pelos RAGAS
from datasets import Dataset

data = {
    'question': questions,
    'answer': answers,
    'contexts' : contexts
}

dataset = Dataset.from_dict(data)

# Importar as métricas para avaliação do RAGAS
from ragas.metrics import (
    faithfulness,
    answer_relevancy,
    context_relevancy)

from ragas import evaluate

# Obter os resultados da avaliação e exibir
score = evaluate(dataset, metrics=[faithfulness,
                                   answer_relevancy,
                                   context_relevancy])
score.to pandas()
score_df = score.to_pandas()
display(score_df)
```

APÊNDICE C – Links dos Arquivos, Prova e Gabarito do CP-T/STA (2023)

Link dos arquivos do concurso público para ingresso no quadro técnico do corpo auxiliar (CP-T) – 2023:
https://www.inscricao.marinha.mil.br/marinha/index_concursos.jsp?id_concurso=442

Link para a prova do CP-T/STA (2023):
https://www.marinha.mil.br/sspm/sites/www.marinha.mil.br.sspm/files/provas/CP-T-2023__PROVA__SEGURAN%C3%87A__TRAFEGO__AQUAVIARIO__AMARELA.pdf

Link para o gabarito do CP-T/STA (2023):
https://www.marinha.mil.br/sspm/sites/www.marinha.mil.br.sspm/files/arquivo/gabaritos/GabFINAL_CPT_2023.pdf

APÊNDICE D – Tabela com resultados da avalição pelo RAGAS

CATEGORIA	QUESTÃO	RESPOSTA	CONTEXTOS RECUPERADO (5)	FR	RR	RC
QUESTÃO DE RESPONSABILIDADE	A quem compete julgar em última instância, como representante da autoridade marítima para o meio ambiente, os recursos sobre multas aplicadas relativas ao descumprimento da NORMAM-20/DPC?	Diretor de Portos e Costas (DPC)	['Normas da Autoridade MarítimaNormas da Autoridade Marítima\\nrelativa à assistência e salvamentorelativa à assistência e salvamento\\ne às atividades de pesquisa,e às atividades de pesquisa,\\nexploração, remoção e demoliçãoexploração, remoção e demolição\\nde coisas e bens afundados,de coisas e bens afundados,\\nsubmersos, encalhados e perdidossubmersos, encalhados e perdidos\\nNORMAM-10/DPCNORMAM-10/DPC\\n-20222022 -MARINHA DO BRASILMARINHA DO BRASIL\\nDIRETORIA DE PORTOS E COSTASDIRETORIA DE PORTOS E COSTAS\\n1ª Revisão1ª Revisão' 'gados e Agentes sendo, para efeito do contido na alínea c, do inciso 4.3.1 anterior, designados\\ncomo AUTORIDADE COMPETENTE.\\n4.3.3 - Compete ao Diretor de Portos e Costas, como REPRESENTANTE DA AUTORIDADE\\nMARÍTIMA PARA O MEIO AMBIENTE:\\na) coordenar as ações decorrentes da aplicação da legislação ambiental por parte\\ndos Agentes da Autoridade Marítima;\\n- 4-1 - NORMAM - 07/DPC\\nRev. 1' '2.6.4.1. Agentes da Autoridade Marítima\\nCompete aos Agentes da Autoridade Marítima (Art. 70, §1º da Lei nº 9.605/1998) lavrar\\nautos de infração ambiental e instaurar processo administrativo.\\n2.6.4.2. Diretor de Portos e Costas (DPC)\\nCompete ao DPC, como REPRESENTANTE DA AUTORIDADE MARÍTIMA PARA O MEIO\\nAMBIENTE, julgar, em última instância, os recursos sobre multas aplicadas relativas ao\\ndescumprimento deste Capítulo.\\nOSTENSIVO REV.3 MOD.1 - 2-11 -' '4.4.3 - do Recurso\\nCaso o infrator não concorde com a penalidade imposta, poderá, sem necessidade\\ndo pagamento da multa, recorrer da decisão, através de recurso interposto junto à CP/DL/AG\\nque o julgou, e dirigido ao Diretor de Portos e Costas (DPC), em última instância.\\nO referido recurso deverá ser interposto dentro do prazo de até vinte dias, contados\\nda data em que o infrator tomar ciência do julgamento.\\nO DPC dentro do prazo de até trinta dias efetuará o julgamento do recurso interpos -\\nto.\\n4.5 - VALORAÇÃO DE MULTA ADMINISTRATIVA POR POLUIÇÃO HÍDRICA\\nCom a publicação da Lei nº 9.966, de 28 de abril de 2000 (Lei de Poluição das Águas), a Au -\\ntoridade Marítima manteve a sua competência legal para aplicar multas em embarcações, plata -\\nformas e suas instalações de apoio que provoquem poluição das águas, estendendo seus limites\\nàs Águas Jurisdicionais Brasileiras (AJB).\\nA presente NORMAM estabelece o rito do processo administrativo ambiental (4.4), garan -' 'Penalidade: multa do Grupo C.\\n Parágrafo único. Cabe ao órgão ambiental competente autuar e multar os infratores na situação prevista\\nneste artigo.\\n Art. 39. Efetuarem o navio ou plataforma com suas instalações de apoio a descarga de água de processo ou\\nde produção em desacordo com os procedimentos aprovados pela autoridade marítima:\\n Penalidade: multa do Grupo C.D4136 http://www.planalto.gov.br/ccivil_03/decreto/2002/d413...\\n10 of 15 18/05/2022 09:45']	0.5	0.82925193848982176	0.04761904761904761
QUESTÃO CONCEITUAL	Segundo a NORMAM-13/DPC, qual é o documento hábil, obrigatório, para embarcações empregadas na navegação em mar aberto e interior, e serve	Rol de Equipagem (modelo DPC - 2303)	['- 2 - 13 - NORMAM -13/DPC \\nREV.1 SEÇÃO II \\n \\n ROL DE EQUIPAGEM E ROL PORTUÁRIO \\n0207 – ROL DE EQUIPAGEM \\na) Conceituação \\nO Rol de Equipagem (modelo DPC -2303) é o documento hábil, obrigatório, para \\nembarcações empregadas na navegação em mar aberto e interior. Serve para garantir os \\ndireitos decorrentes dos embarques e desembarques de tripulantes verificados em uma única \\nembarcação. \\n \\nDeve conter as seguintes anotações: \\n1) dados da embarcação, do(s) proprietário(s) e do armador; \\n \\n2) assinatura e nome legível do Comandante do navio, proprietário, armador ou seu \\npreposto (representante legal); \\n \\n3) dados dos tripulantes; e \\n \\n4) dados dos embarques e desembarques dos tripulantes. \\n \\nb) Emissão \\nO Rol de Equipagem deverá ser adquirido na Empresa Gerencial de Projetos Navais' '- 4 - 13 - NORMAM-01/DPC \\nMod 41 b) Certificado de Registro de Embarcações Estrangei ras emitido pelo país de origem (para \\nnavios estrangeiros afretados); \\nc) Certificado de Autorização de Afretamento (CAA), emitido pela ANTAQ (navios \\nestrangeiros afretados); \\nd) Atestado de Inscrição Temporária (para navios es trangeiros afretados), documento \\noriginal; \\ne) Bilhete de Seguro Obrigatório de Danos Pessoais Causados por Embarcações e sua \\nCarga (DPEM). Esta obrigatoriedade está suspensa, em conformidade com a Lei nº 13.313 de 14 de \\njulho de 2016. Qualquer alteração referente ao assu nto será divulgada oportunamente. \\nf) Certificado de Compensação de Agulha/Curva de De svio, documento original; e \\ng) As embarcações SOLAS deverão possuir os Certific ados e demais documentos \\nreferentes aos instrumentos pertinentes das Convenç ões Internacionais adotadas pelo Brasil e \\nsuas emendas (SOLAS 74/78, MARPOL 73/78, Linhas de Carga/66, Arqueação/69, STCW/78 e' '- 1 - 38 - NORMAM -13/DPC \\nREV.1 d) Documentação e pré-requisitos necessários para revalidação do Certificado modelo \\nDPC-1034: \\n1) Requerimento do interessado; \\n2) CIR (cópia autenticada ou cópia simples com apresentação da original para \\nautenticação na CP/DL/AG da folha de rosto com etiqueta de dados pessoais e das folhas de \\nregistros de embarque da CIR). A CIR não deverá ser retida na OM, salvo fundamentação legal; \\n3) Comprovação de embarque em navios de bandeira estrangeira (Anexo 1-G da \\nNORMAM -13 (quando aplicável); \\n4) Documento que comprove tempo de embarque, conforme previsto no item 0126 da \\nNORMAM -13/DPC, (quando aplicável); \\n5) Carteira de identidade dentro da validade (cópia autenticada ou cópia simples com \\napresentação do original) ou, no caso de estrangeiro , Carteira de Registro Nacional Migratório – \\nCRNM dentro da validade (cópia autenticada ou cópia simples	1.0	0.80519919334191266	0.16666666666666666

	para garantir os direitos decorrentes dos embarques e desembarques de tripulantes verificados em uma única embarcação?		com apresentação do original);' '- 1 - 37 - NORMAM -13/DPC \\nREV.1 Medicina (CRM), que comprove bom estado mental e físico e, explicitamente, as condições \\nvisuais e auditivas; \\n9) Documento que comprove tempo de embarque (conforme previsto no item 0126 da \\nNORMAM -13/DPC); \\n10) Documento que comprove tempo de embarque em navios de bandeira estrangeira \\n(Anexo 1-G da NORMAM -13/DPC) (quando aplicável); \\n11) Certificado de competência e outros que comprovem habilitações específicas a serem \\nregistradas no novo certificado (cópia autenticada ou cópia simples com apresentação do \\noriginal); \\n12) Uma foto de frente, com fundo branco e sem chapéu (a ser capturada nos locais de \\natendimento nas Capitanias, Delegacias ou Agências); \\n13) Documento, emitido pela empresa/navio, atestando que o marítimo tenha sido \\nsubmetido a treinamentos específicos em instalações apropriadas a bordo, compreendendo' '- 4 - 34 - NORMAM -13/DPC \\nREV.1 SEÇÃO III \\nATRIBUIÇÕES COMUNS A TODOS OS TRIPULANTES \\n \\n0419 – PRECEITOS PARA OS TRIPULANTES NA NAVEGAÇÃO EM MAR ABERTO E INTERIOR \\nA todos os tripulantes, compete: \\n1) executar com zelo e eficiência os serviços que lhe são afetos; \\n2) cumprir as leis em vigor e as presentes Normas; \\n3) obedecer ao Comandante e demais autoridades de bordo; \\n4) cumprir a organização de bordo e as instruções expedidas pelo Armador, ou por seu \\npreposto, representante legal ou Proprietário; \\n5) abster -se de rixas e desordens a bordo; \\n6) manter decência no tratamento com os demais tripulantes; \\n7) não se ausentar de bordo sem prévio consentimento do Comandante; \\n8) apresentar -se a bordo pronto para seguir viagem no tempo contratado; \\n9) não se recusar a seguir viagem; \\n10) auxiliar o Comandante em caso de ataque à embarcação ou sobrevindo qualquer']			
QUESTÃO PROCEDIMENTAL	De acordo com as normas para a Investigação de Segurança dos Acidentes e Incidentes Marítimos (ISAIM) previstos na NORMAM-09/DPC, qual documento deve ser produzido ao final da investigação?	Relatório da investigação de segurança marítima	['NORMAM-09/DPC\\nRev 1 Mod 1- 2-1 -CAPÍTULO 2\\nNORMAS PARA A INVESTIGAÇÃO DE SEGURANÇA DOS ACIDEN TES E INCIDENTES\\nMARÍTIMOS (ISAIM)\\n0201 - PROPÓSITO\\nDivulgar o “Código de Normas Internacionais e Práticas Recomendadas para uma\\nInvestigação de Segurança de um Acidente Marítimo, ou de um Incidente Marítimo”, também\\nconhecido como “Código de Investigação de Aciden tes (CIA)”, aprovado pela Resolução MSC.255(84)\\nda Organização Marítima Internacional (IMO), os procedimen tos da Investigação de Segurança dos\\nAcidentes e Incidentes Marít imos (ISAIM) e seu encaminhamento à Diretoria de Portos e Costas\\n(DPC).\\n0202 – APLICAÇÃO\\nSob as Convenções SOLAS Regulação I/21 e MARPOL artigos 8 e 12, c ada Administração\\ndeve conduzir uma investigação quando da ocorrência de um acidente envolvendo navios de sua\\nbandeira sujeitos a essas convenções, e deverá suprir a Organi zação Marítima Internacional (IMO)\\ncom as informações concernentes às conclusões de tais investigações.' 'NORMAM-09/DPC\\nRev 1 Mod 12-2jurisdição. O mesmo prazo será observado quando determinada a instauração de ISAIM\\npela DPC por conta da ocorrência de um ac idente ou incidente marítimo muito grave envolvendo\\nembarcações de bande ira estrangeira, e de acidentes ou in cidentes marítimos envolvendo\\nembarcações nacionais que não sejam classificados como muito grave.\\nApós sua instauração a ISAIM somente poderá ser cancelada com autorização da DPC.\\n0204 – DEFINIÇÕES\\nAs definições a segui r, entre outras, constam do Código de Normas Internacionais e Práticas\\nRecomendadas para uma Investigação de Segurança de um Acidente Marítimo, ou de um Incidente\\nMarítimo (Código de Investigação de A cidentes - CIA), aprovado pela Resolução MSC.255(84) da\\nOrganização Marítima Internacional (IMO).\\nQuando forem empregados nas normas obrigatórias e nas práticas recomendas para as\\ninvestigações de segurança marítima, os termos a seguir pos suem o seguinte signifi cado:' 'NORMAM-09/DPC\\nRev 1 Mod 1informações, uma vez que são facilmente disponíveis:\\n.1 o nome do navio e do seu Estado da Bandeira;\\n.2 o número IMO de identi ficação do navio;\\n.3 a natureza do acidente marítimo;\\n.4 o local do acidente marítimo;\\n.5 a hora e a data do acidente marítimo;\\n.6 o número de qua isquer pessoas gravemente feridas ou mortas;\\n.7 as consequências do acidente marítimo para pessoas, propriedades e meio ambiente; e\\n.8 a identificação de qualquer outro navio envolvido.\\nCapítulo 6\\nEXIGÊNCIA DE INVESTIGAR ACID ENTES MARÍTIMOS MUITO GRAVES\\n6.1 Deve ser realizada uma investigação de segurança marítima na ocorrência de qualquer\\nacidente marítimo muito grave.\\n6.2 Sujeito a qualquer Acordo feito em consonância com o Capítulo 7, o Estado da bandeira\\nde um navio envolvido num acidente marítimo muito grave é responsável por assegurar que uma\\ninvestigação de segurança marítima seja realizada e concluída de acordo com este Código.\\nCapítulo 7' 'sua instauração.\\nEm situações especiais e ex cepcionais, em que a ISAIM não tenha sido con cluída no prazo\\nmáximo de noventa (90) dias, a prorrogação poderá ser autorizada pelo Diretor de Portos e Costas,\\nque avaliará o pedido devidamente circunstanciado e decidirá a respeito.\\n0206 – CULPADOS E/OU RESPONSÁVEIS\\nAs ISAIM não procuram atribuir culpa nem determinar responsabilidades. Em vez disso,\\numa ISAIM, conforme definida no Código, é u ma investigação reali zada com o propósito de impedir\\nque no futuro ocorram acidentes e incidentes marítimos semelhantes ou, no caso de ocorrerem,\\nque suas consequências se jam minimizadas.\\n0207 - NOTIFICAÇÃO SOBRE INVESTIGAÇÃO DE SEGURANÇA DE ACIDENTE MARÍTIMO\\nConforme disposto no Código de Investigação de Acidentes, quando um acidente ou\\nincidente marítimo ocorre em alto-mar ou uma zona econômica exclusiva, o Estado da bandeira do\\navio ou navios envolvidos, deverá notificar outros Estados substancialmente interessados logo que' 'NORMAM-09/DPC\\nRev 1 Mod 12.11 Uma investigação de segurança marítima significa uma investig ação, ou um inquérito, (como\\nfor denominado por um Estado) de um acidente marítimo, ou de um incidente marítimo, realizado\\ncom o propósito de impedir a ocorrência de acidentes e de incidentes marítimos no futuro. A\\ninvestigação abrange a coleta e a análise de provas, a identificação dos fatores causais e a\\nelaboração das recomendações de segurança que forem necessárias.\\n2.12 Um relatório da investigação de segurança marítima significa um relatório que contenha:\\n.1 um resumo expondo em linhas gerais os fatos básicos do acidente marítimo, ou do incidente\\nmarítimo, e informando se em decorrência daqueles fatos ocorreram quaisquer mortes,\\nferimentos ou poluição;\\n.2 a identidade do Estado da bandeira, dos armadores, operadores, da companhia\\nidentificada no certificado de gere nciamento da segurança e da sociedade classificadora (sujeito a']	1.0	0.85 982 252 515 468 35	0.05 263 157 894 736 842

QUESTÃO TEMPORAL	De acordo a NORMAM-30/DPC, o orgao de execução oferece informações essenciais ao controle e ao aperfeiçoamento do Ensino Profissional Marítimo (EPM), mediante o preenchimento de dois relatórios: Relatório de Curso de Aquaviário (RECO) e o Relatório de Disciplina (REDIS). Assim, em qual prazo o RECO deve ser elaborado no SISGEPM?	Até dez dias após a conclusão do curso.	['OSTENSIVO NORMAM-30/DPC\\nRELATÓRIOS REFERENTES AOS CUR SOS\\n2.16 - RELATÓRIO DE CUR SO DE AQUAVIÁRIOS (RECO)\\nDestina-se a oferecer informações essenciais ao controle e ao aperfeiçoamento\\ndo EPM. Deverá ser elaborado p elo OE que realizou o curso ou, no caso de Entidade\\ncredenciada, pelo OE a ela vinculado, com base nos Relatórios de Disciplina (REDIS) e nos\\nQuestionários Pedagógicos (QP), no SISGEPM, até dez dias após a conclusão. Ao término\\nde cada curso, o OE deverá enviar mensagem notificando à DPC, com as seguintes\\ninformações:\\n- Nome e turma do curso;\\n- Período de realização;\\n- Número de inscritos;\\n- Número de matriculados;\\n- Número de aprovados;\\n- Número de reprovados; e\\n- Número de desistentes.\\n2.17 - RELATÓRIO DE DISCIPLINA (R EDIS) (Anexo F)\\nSerá preenchido nos cursos para Aquaviários, por disciplina, pelo professor ou\\ninstrutor. Destina-se a reunir informações que, analisadas pelo OE, possam contribuir para o' 'OSTENSIVO NORMAM -30/DPC \\nOSTENSIVO - 3-1 - REV.1 \\n CAPÍTULO 3 \\nPROGRAMA DO ENSINO PROFISSIONAL MARÍTIMO (PREPOM) \\n3.1 - PROPÓSITO \\nO Programa do Ensino Profission al Marítimo para Aquaviários (PREPOM-\\nAquaviários) tem o propósito de divu lgar aos Órgãos de Execução (OE), aos Órgãos de Apoio \\n(OA), aos Órgãos Conveniados ou Terceirizados (OC/T) e à comunid ade aquaviária em \\ngeral, a programação dos cursos e estágios do EPM aprovada pelo Órgão Central do \\nSistema do Ensino Profission al Marítimo (SEPM) p ara determinado ano. Do PREPOM \\nconst am, também, in formações específicas sobre os cursos e estágios, tais como: condi ções \\npara in scrição, propósito dos cursos, facilidades oferecidas aos alunos, certificados \\nconcedidos, lo cal de realização, nú mero de v agas, requisitos para matrícula, etc., \\nrespeitados os recursos financeiros disponív eis. \\n3.2 - ELABORAÇÃO \\nO PREPOM -Aquaviários é elaborado, anualmente, pela Superintendência do Ensino' 'Para cursos realizados no Sistema do Ensino Profissional Marítimo (SEPM), o Certificado \\nserá emitido automaticamente pelos Órgãos de Execução (OE), após aprovação no curso \\ncorrespondente. Para os demais casos, observar os seguintes procedimentos: \\n1) Emissão de um único certificado relativo aos extintos cursos EBPS, ECIN, ESPE e ESRS, \\nde acordo com a Portaria nº 347/2013/DPC; \\n2) Emissão de um certificado relativo à familiarização em navio -tanque, de acordo com a \\nPortaria nº 347/2013/DPC; \\n3) Emissão de um certificado de equivalência de cursos previstos na NORMAM -24/DPC \\naos do SEPM, conforme a correspondência constante em Portaria específica emitida pela DPC; e \\n4) Emissão de um certificado referente ao ingresso no grupo marítimo, em consequência \\nde uma transferência de grupo (recebimento de Regra). \\n \\nREV.1 MOD.2' 'OSTENSIVO NORMAM -30/DPC \\nOSTENSIVO - 3-2 - REV.1 \\n 3.3 - SISTEMA DE G ERENCIA MENTO DO ENSINO PROFISSIONAL MARÍTIMO (SISGEPM) \\nPara facilitar e aperfeiçoar as ativid ades do EPM, a DPC desenvolv eu o SISGEPM, \\ncom o propósito de facilitar a elaboração das propos tas de cursos pelos OE, a montagem \\ne o acompanhamento do PREPOM-Aquaviários, bem como controlar o desempenho dos \\nalunos e dos professores. Sendo assim, tod as as propost as deverão ser elaboradas \\ndiretamente no S ISGEPM. Os OE não devem perder de vista que o FDEPM, que custeia a \\nestrutura do SEPM, advém de contribuições da comunid ade aquaviária. Assim, no processo \\nde montagem do PREPOM-Aquaviários, deve-se ter presente como uma d as metas do E PM a \\nadequabilid ade da p rogramação dos cursos, de modo que eles sejam condu zidos em épocas \\noportunas evitando, sempre que possív el, cursos em períodos conflitantes com os \\ninteresses das empresas ou de outras Enti dades da comunidade usuária.' '- 6 - 11 - NORMAM -13/DPC \\nREV.1 \\n e) Comprovante de residência com CEP, expedido no prazo máximo de noventa (90) dias \\ncorridos, em nome do interessado (cópia autenticada ou cópia simples com apresentação do \\noriginal), ou Declaração de Residência assinad a pelo Aquaviário, conforme modelo constante do \\nAnexo 1 -L (com reconhecimento por semelhança , caso o declarante não esteja presente); e \\nf) Declaração do requerente, expondo o(s) motivo(s) da solicitação da 2ª via. \\nOBSERVAÇÃO: \\nO requerimento poderá ser remetido pelo correio para um dos Centros de Instrução (CI), \\nde acordo com o local onde o curso foi realizado. Neste caso, as cópias enviadas dos \\ndocumentos necessários deverão estar autenticadas. \\n \\n0616 – INFORMAÇÕES DAS ESCOLAS DE FORMAÇÃO DE OFICIAIS DA MARINHA MERCANTE \\n(EFOMM) \\nA EFOMM levantará as informações que incluam, sempre que possível: \\na) nome do aluno; \\nb) filiação;']	1.0	0.88673340000848 8	0.25