



UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS
DEPARTAMENTO DE ENGENHARIA ELÉTRICA
CURSO DE GRADUAÇÃO EM ENGENHARIA DE CONTROLE E AUTOMAÇÃO

JOSÉ RICARDO CABRAL VIANNA

**APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA NO SETOR
ELÉTRICO: ANÁLISE COMPARATIVA DA GESTÃO ENERGÉTICA COM
AGENTES DE IA**

Recife
2025

JOSÉ RICARDO CABRAL VIANNA

**APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA NO SETOR
ELÉTRICO: ANÁLISE COMPARATIVA DA GESTÃO ENERGÉTICA COM
AGENTES DE IA**

Trabalho de Conclusão de Curso
apresentado ao Curso de Graduação em
Engenharia de Controle e Automação da
Universidade Federal de Pernambuco,
como requisito parcial para obtenção do
grau de Bacharel em Engenharia de
Controle e Automação.

Orientador(a): Prof. Dr. Jeydson Lopes da Silva

Recife
2025

**Ficha de identificação da obra elaborada pelo autor,
através do programa de geração automática do SIB/UFPE**

Vianna, José Ricardo Cabral.

Aplicação de inteligência artificial generativa no setor elétrico: análise comparativa da gestão energética com agentes de IA / José Ricardo Cabral Vianna. - Recife, 2025.

66 p. : il., tab.

Orientador(a): Jeydson Lopes da Silva

Trabalho de Conclusão de Curso (Graduação) - Universidade Federal de Pernambuco, Centro de Tecnologia e Geociências, Engenharia de Controle e Automação - Bacharelado, 2025.

Inclui referências, apêndices.

1. inteligência artificial. 2. engenharia elétrica. 3. eficiência energética. 4. modelos de linguagem. 5. gestão energética. I. Silva, Jeydson Lopes da. (Orientação). II. Título.

620 CDD (22.ed.)

JOSÉ RICARDO CABRAL VIANNA

**APLICAÇÃO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA NO SETOR
ELÉTRICO: GESTÃO ENERGÉTICA COM AGENTE DE IA**

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Engenharia de Controle e Automação da Universidade Federal de Pernambuco, como requisito parcial para obtenção do grau de Bacharel em Engenharia de Controle e Automação.

Aprovado em: 17/12/2025

BANCA EXAMINADORA

Prof. Dr Jeydson Lopes da Silva (Orientador)
Universidade Federal de Pernambuco

Prof. Dr. Douglas Contente Pimentel Barbosa (Examinador interno)
Universidade Federal de Pernambuco

Eng. M.Sc. Joas Ronaldy de Holanda Santos (Examinador externo)
Universidade Federal de Pernambuco

AGRADECIMENTOS

Gostaria de agradecer, primeiramente, à minha mãe e meu pai que sempre apoiaram os meus sonhos, comemoraram minhas conquistas e me proporcionaram uma base para o meu crescimento tanto pessoal, quanto como engenheiro.

Segundamente, gostaria de agradecer a todos os professores que fizeram parte da minha caminhada e dividiram uma parte dos seus conhecimentos e vivências ao longo da minha formação.

Agradeço aos amigos e colegas que fiz durante essa caminhada, além do projeto de extensão Maracatronics, que foi meu primeiro contato com temas de robótica dentro da faculdade.

Por fim, sou grato à UFPE por me proporcionar um ambiente que me permitiu aprender sobre engenharia e realizar um intercâmbio acadêmico para a França.

O treinamento de usuários consiste em parte do processo de educação, em base repetitiva, compreende ações e/ou estratégias para desenvolver determinadas habilidades ou habilidades específicas do usuário por desconhecer situações específicas de uso da biblioteca e seus recursos informacionais, que envolvem o conjunto de meios necessários para tal (DIAS; PIRES, 2004, p. 36).

RESUMO

Este trabalho consiste no desenvolvimento de uma solução para engenharia elétrica utilizando técnicas de inteligência artificial como ferramentas para auxiliar engenheiros ou qualquer usuário em temas de gestão energética de uma residência, com foco principal em manutenção preventiva/corretiva e eficiência energética de equipamentos. Os temas de: *machine learning*, inteligência artificial, *fine-tuning*, modelos de linguagem e eficiência energética são explicitados ao longo do trabalho. As ferramentas utilizadas incluem *softwares* de simulação de dados, *frameworks* e bibliotecas especializadas no treinamento de modelos de inteligência artificial generativa, banco de dados e uma plataforma de criação de fluxos de trabalho para automatização de processos. Foi desenvolvido um agente de inteligência artificial capaz de recuperar dados de um banco de dados, somá-los a dados fornecidos em um *drive* para responder questões pertinentes sobre gestão energética. O resultado desse trabalho demonstra algumas possibilidades e limites que uma agente de inteligência artificial generativa possui, tanto como ferramenta de estudo acadêmico quanto como uma ferramenta de uso cotidiano na gestão energética por um usuário comum. São apresentados resultados técnicos, financeiros e sugestões de melhorias que visam o aprimoramento da ferramenta.

Palavras-chave: inteligência artificial, engenharia elétrica, eficiência energética, modelos de linguagem, gestão energética.

ABSTRACT

This work consists of developing a solution for electrical engineering using artificial intelligence techniques as tools to assist engineers or any user in topics related to electrical engineering and residential energy management, with a primary focus on preventive/corrective maintenance and equipment energy efficiency. The topics of machine learning, artificial intelligence, fine-tuning, language models, and energy efficiency are explained throughout the work. The tools used include data simulation software, frameworks and specialized libraries for training generative artificial intelligence models, databases, and a workflow creation platform for process automation. An artificial intelligence agent was developed capable of retrieving data from a database, combining it with data stored on a drive to answer relevant questions about energy management. The results of this work demonstrate some possibilities and limitations that a generative artificial intelligence agent has both as an academic study tool and as an everyday tool for energy management for individuals outside the field. Technical and financial results are presented, along with suggestions for improvements aimed at enhancing the tool.

Keywords: artificial intelligence, electrical engineering, energy efficiency, language models, energy management.

LISTA DE ILUSTRAÇÕES

Figura 1 – Diagrama comparativo da inteligência artificial	16
Figura 2 – Resumo ilustrativo sobre <i>Machine Learning</i>	19
Figura 3 – Modelo simples de rede neural	20
Figura 4 – Funções de ativação usuais (azul) e suas derivadas (vermelho).....	22
Figura 5 – Diagrama da arquitetura de transformers.....	26
Figura 6 – Diagrama do <i>fine-tuning</i>	28
Figura 7 - Diagrama do <i>RAG</i>	29
Figura 8 - Engenharia de <i>prompt</i>	31
Figura 9 - Selo Procel.....	33
Figura 10 – Custo de um motor elétrico após 10 anos	34
Figura 11 – Harmônicas de 3 ^a e 5 ^a ordem	35
Figura 12 – Diagrama geral do agente	38
Figura 13 – <i>Workflow</i> de alimentação	44
Figura 14 – Visão geral do <i>workflow</i> principal do agente	44
Figura 15 – Etapa de <i>prompt</i>	45
Figura 16 – Etapa do modelo	46

LISTA DE EQUAÇÕES

Forma geral do neurônio	21
Atualização dos pesos.....	22
Erro quadrático médio (MSE).....	22
Equação de entropia cruzada.....	23
Erro percentual médio absoluto (MAPE).....	23

LISTA DE TABELAS

Tabela 1 - Tabela resumo de conteúdo inserido no RAG	39
Tabela 2 - Critérios de avaliação.....	47
Tabela 3 – Tempo de geração e qualidade da resposta gerada por cada modelo....	48
Tabela 4 – Comparativo resumido de performance do tempo de execução e performance média	54
Tabela 5 – Custos financeiros dos modelos.....	55

LISTA DE SIGLAS

IA	Inteligência Artificial
LLM	Large Language Models
RAG	Retrieval Augmented Generation
ReLU	Rectified Linear Unit
SVM	Support Vector Machine

SUMÁRIO

1	INTRODUÇÃO	14
1.1	OBJETIVOS	14
1.1.1	Geral.....	14
1.1.2	Específicos	15
1.2	ORGANIZAÇÃO DO TRABALHO.....	15
2	FUNDAMENTAÇÃO TEÓRICA	16
2.1	INTELIGÊNCIA ARTIFICIAL	16
2.2	<i>MACHINE LEARNING</i>	17
2.3	<i>DEEP LEARNING</i>	20
2.4	<i>LARGE LANGUAGE MODELS</i>	23
2.4.1	Transformers	25
2.5	AGENTES DE INTELIGÊNCIA ARTIFICIAL	26
2.5.1	Fine-Tuning	27
2.5.2	RAG (Retrieval Augmented Generation)	29
2.6	ENGENHARIA DE <i>PROMPT</i>	30
2.7	EFICIÊNCIA ENERGÉTICA	32
2.7.1	Qualidade de energia	34
3	DESENVOLVIMENTO E METODOLOGIA	36
3.1	METODOLOGIA	36
3.2	DEFINIÇÕES INICIAIS	36
3.3	PESQUISA	39
3.4	TREINAMENTO NAS FERRAMENTAS	41
3.5	GERAÇÃO DE DADOS	41
3.6	<i>FINE-TUNING</i>	42
3.7	<i>WORKFLOWS</i> E BANCO DE DADOS	43

3.7.1	Workflow de alimentação do RAG	43
3.7.2	Workflow principal do agente.....	44
3.8	TESTE COM O <i>RAG</i>	47
3.9	ANÁLISE TÉCNICA.....	48
3.10	ANÁLISE FINANCEIRA.....	54
4	CONCLUSÕES E PROPOSTAS DE CONTINUIDADE	56
4.1	CONCLUSÕES TÉCNICAS	56
4.2	CONCLUSÕES FINANCEIRAS	57
4.3	LIMITAÇÕES DO PROJETO.....	57
4.4	PROPOSTAS DE CONTINUIDADE	58
	REFERÊNCIAS.....	59
	APÊNDICE A – EXEMPLO DE COMPARAÇÃO DE RESULTADOS DAS PERGUNTAS AOS MODELOS	62

1 INTRODUÇÃO

O surgimento recente de *chatbots* inteligentes e modelos capazes de gerar imagens originais ou reproduções quase indistinguíveis dos mais renomados artistas gerou o questionamento se seria possível utilizar um modelo de inteligência artificial que fosse capaz de projetar, resolver circuitos, gerar códigos ou até mesmos supervisórios do zero.

A inteligência artificial vem evoluindo exponencialmente nos últimos anos, principalmente na área de geração de conteúdo, ao mesmo tempo em que o desenvolvimento na área da engenharia elétrica também se mostra em constante evolução para suprir as necessidades da sociedade do futuro.

O campo da energia elétrica para o usuário leigo ainda se mostra como um conjunto de assuntos muito inexplorados, incluindo o funcionamento básico de circuitos até o impacto final dessas tecnologias na vida cotidiana. Ainda é comum uma pessoa não entender muito bem como a própria fatura de energia é cobrada e muito menos como identificar alguma fonte de desperdícios ou de falhas na sua própria residência. Somado a esse problema da sociedade brasileira, ferramentas de IA generativa com o intuito de auxiliar em questões mais básicas da gestão energética residencial não são muito difundidas, principalmente na língua portuguesa.

O objetivo deste trabalho consiste em unir as áreas de inteligência artificial e engenharia elétrica para desenvolver uma solução capaz de auxiliar na análise de consumo energético de uma residência.

1.1 Objetivos

1.1.1 Geral

Desenvolver uma aplicação baseada em um modelo de inteligência artificial generativa que seja capaz de auxiliar na gestão de uma residência, edifício, comércio ou estabelecimentos no geral e que seja capaz de sugerir melhorias para tornar o

lugar mais eficiente, reduzindo custos e auxiliando na detecção de falhas da instalação elétrica.

1.1.2 Específicos

- Criação de um fluxo de trabalho focado num agente de IA usando ferramentas como *n8n*, *Pinecone* e modelos variados de *LLM*;
- Utilizar a estratégia de *fine-tuning* de um modelo *open-source* visando principalmente a gestão energética residencial;
- Buscar material em artigos científicos, artigos da internet, normas técnicas e gerar simulações para criação de um *RAG*;
- Realizar análise comparativa de desempenho no tocante a tempo de execução e qualidade das respostas entre 3 diferentes modelos (*Groq LLaMA 4*, *ChatGPT 4.1-mini*, *LLaMA 3.2 fine-tuned*) em cenários diversos de possíveis questionamentos no campo da gestão energética.
- Realizar avaliação financeira de cada modelo como base um protótipo.

1.2 Organização do Trabalho

O trabalho está dividido em quatro capítulos. No Capítulo 1 há uma sessão de introdução que define os objetivos e organiza a estrutura do projeto. O Capítulo 2 consiste na fundamentação teórica, onde os temas abordados no trabalho são expostos e conceitos são exemplificados para facilitar no entendimento do assunto e tema proposto. No Capítulo 3, será realizado o desenvolvimento real do trabalho, onde serão expostas cada etapa do processo do trabalho, assim como os resultados também serão discutidos. No Capítulo 4, haverá uma etapa de conclusão, onde será mostrado se o projeto satisfez os objetivos e possíveis melhorias e sugestões para próximas pesquisas serão propostos.

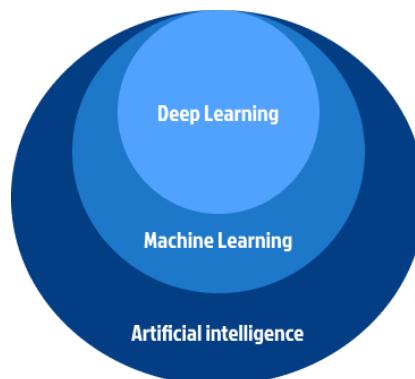
2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo, são apresentados os principais conceitos envolvidos no uso de inteligência artificial e definições relacionadas à eficiência energética. São explicadas as diferenças entre inteligência artificial, *machine learning* e *deep learning*, além de abordar conceitos sobre *LLMs* e transformers. Ao fim, explica-se o que é a eficiência energética e qualidade de energia. Espera-se que ao fim do capítulo, o leitor tenha uma visão contextualizada do problema abordado.

2.1 Inteligência Artificial

O conceito da inteligência artificial é muito amplo (Santos, 2023), sendo considerado qualquer técnica que permita o computador emular o pensamento humano do mais simples ao mais complexo. Ao adicionar o campo da estatística nesse estudo, surgiu o aprendizado de máquina (*Machine learning*), permitindo que os computadores se tornassem ainda mais capazes de desempenhar as suas funções e simular de fato o aprendizado. Buscando soluções ainda mais complexas, surgiu a aprendizagem profunda (*Deep learning*), sendo um subconjunto da aprendizagem de máquina, mas com a inserção da ideia de redes neurais, possibilitando a máquina de treinar a si mesma. Na Figura 1, pode-se ver um diagrama que exemplifica a relação entre essas três áreas.

Figura 1– Diagrama comparativo da inteligência artificial



Fonte: próprio

2.2 Machine Learning

Os algoritmos fora do que é considerado aprendizado de máquina, como sistemas baseados em regras (SE-ENTÃO) ou redes bayesianas, se mostraram bons para resolver uma quantidade limitada de problemas, porém necessitavam de um planejamento considerável por parte do desenvolvedor da aplicação. O trabalho humano começou a ser pougado, pois anteriormente uma pessoa precisaria fazer essa atividade, mas os processos ainda poderiam ser mais bem otimizados no tocante ao tempo e desempenho, assim surgiu o questionamento se o computador poderia aprender como uma pessoa.

O conceito de aprendizado é muito amplo e foram desenvolvidas diversas teorias que ajudam a tentar compreender o assunto. Segundo (MARTINS, 2020) existem 6 teorias da aprendizagem:

- Teoria Experiencial: Baseia-se na experiência individual da pessoa, ou seja, cada elemento aprende de maneira única e deve-se adequar às necessidades de cada um.
- Teoria Cognitiva da Aprendizagem: Baseia-se na construção do conhecimento, ou seja, depende das operações cognitivas como pensar, raciocinar e tomar decisões e não apenas de resultados, além de levar em consideração o conhecimento prévio do indivíduo.
- Teoria Social Cognitiva: Nesse caso, o ambiente é a principal fonte de aprendizado do indivíduo e trabalha a ideia de autorregulação, o que permite a pessoa controlar os próprios comportamentos, pensamentos e sentimentos.
- Andragogia: É uma teoria voltada ao aprendizado de adultos, pois distingue-se a aprendizagem na fase adulta da infância. Baseando-se na necessidade do saber, na independência, na orientação para aprender, no papel da experiência e na disposição para aprender do indivíduo.
- Teoria da Carga Cognitiva: Surgiu como uma necessidade de entender como o cenário tecnológico atual interfere na aprendizagem. Baseia-se no conceito de carga cognitiva e admite que uma pessoa não pode absorver

mais do que sua carga cognitiva permite de uma vez. Existindo a carga intrínseca, a carga externa e a carga relevante.

- Teoria de Fitts e Posner: Essa teoria afirma existirem estágios do aprendizado e cada estágio depende da capacidade de processamento de informação do indivíduo. Sendo elas a fase cognitiva, a fase associativa e a fase da autonomia.

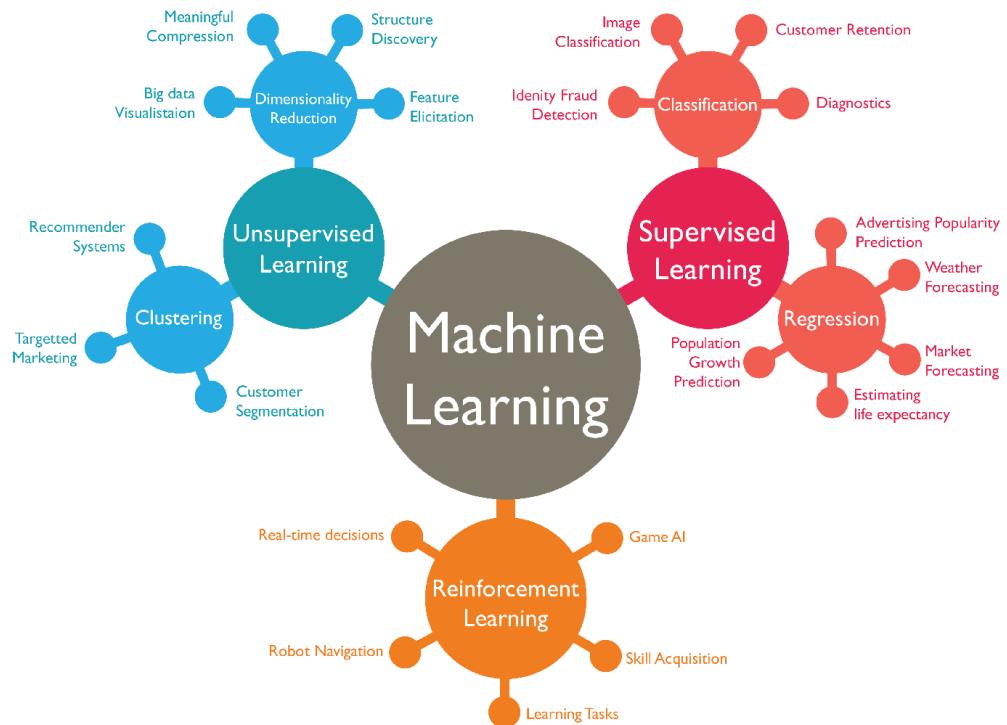
A máquina ainda diverge do ser humano em diversos aspectos e entende-se que a máquina não raciocina de fato, mas é capaz de emular essa operação cognitiva de maneira a resolver problemas como uma pessoa ou, em vários casos, melhor (Lake, 2016).

Baseado em (IBM, s.d.), existem 3 modelos principais de aprendizados de máquina e todos os algoritmos acabam se enquadrando em alguns deles:

- Aprendizado Supervisionado: Nesse modelo, um agrupamento de dados rotulados é utilizado na fase de treinamento para depois analisar dados não rotulados e prever qual será a classificação de determinado elemento ou chegar em um número específico. Alguns exemplos são *Random Forest* (Breiman, 2001) e máquina de vetores de suporte (SVM) (Nascimento, 2009).
- Aprendizado não-Supervisionado: Nesse modelo, os dados não apresentam rótulo e os algoritmos são capazes de identificar padrões escondidos sem que uma pessoa precise definir nada. É muito útil na exploração de dados, segmentação de clientes, reconhecimento de padrões e redução de dimensão de um problema. Alguns exemplos são a análise de componentes principais (PCA) (Santo, 2012) e *K-means* (Oliveira, 2018).
- Aprendizado por reforço: Esse modelo se assemelha ao aprendizado supervisionado, mas nesse caso não se usam dados da amostra, pois o aprendizado ocorre por tentativa e erro na medida que o modelo avança num objetivo. Um exemplo é o *Q-learning* (Data Science Academy, 2025) e o SARSA (Monteiro, 2004), muito úteis na navegação de robôs em ambientes desconhecidos.

Na Figura 2, as definições sobre aprendizado de máquina estão resumidas e ilustradas.

Figura 2 – Resumo ilustrativo sobre *Machine Learning*



Fonte: (Shewan, 2023)

Dois processos muito importantes em diversas resoluções de problemas de aprendizagem de máquina são a normalização (Matos, 2023) e padronização (Shaibu, 2024) dos dados. A normalização consiste em reescalar os dados para um intervalo definido, no caso [0,1]. A padronização transforma dados para a média zero e desvio padrão 1.

De maneira ampla, pode-se dizer que qualquer aplicação mais específica de algoritmos mais avançados de *Deep Learning* e *LLMs* são exemplos de aplicações de aprendizado de máquina. De fato, pode-se dizer que o ramo do aprendizado de máquina possibilitou o florescimento de tecnologias futuras mais potentes e com amplas funcionalidades.

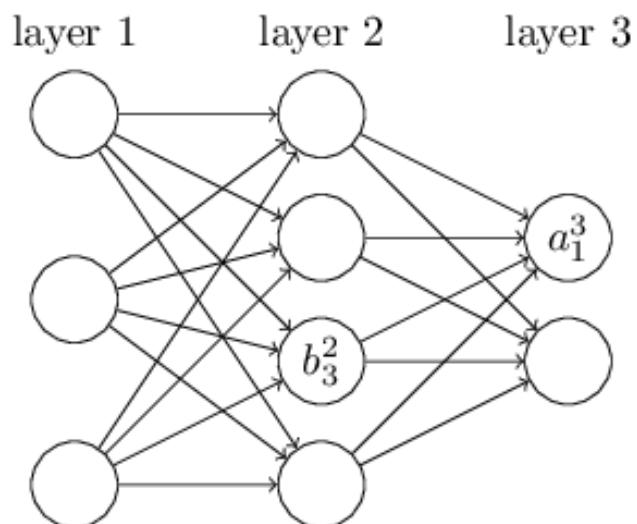
2.3 Deep learning

A ideia de utilizar redes neurais na computação surgiu da comparação com a própria rede neural animal, consistindo num grande conjunto de pequenos elementos que se assemelham aos neurônios e juntos são capazes de gerar respostas e encontrar padrões de maneira que antes era inviável.

A rede neural é dividida em 3 tipos de camadas principais: A camada de entrada (*Input layer*), as camadas ocultas (*Hidden layers*) e a camada de saída (*Output layer*). O trabalho de parametrização de uma rede neural consiste basicamente em encontrar o valor mais satisfatório dos parâmetros, sendo os principais, número de camadas ocultas e número de neurônios por camada. Na Figura 3, é possível observar um exemplo simples de rede neural.

Esse algoritmo segue a lógica de caixa preta, ou seja, não se sabe exatamente o caminho que a máquina vai definir para gerar uma resposta, os valores de entrada são inseridos no modelo junto a alguns modelos de resposta na fase de treinamento e seguindo as fórmulas da função de ativação definida na fase de *forward propagation* (Academy, 2023) e o aprendizado ocorre no retorno, quando a máquina entra na fase de *backpropagation*, o modelo é capaz de gerar pesos que são os dados que permitem o “pensamento” da máquina.

Figura 3 – Modelo simples de rede neural



Fonte: (Nielsen, 2019)

Na fase de propagação para frente, os valores inseridos na camada são multiplicados pelos pesos de cada neurônio, somados a um valor de *bias*, depois somados às outras somas provenientes de outros neurônios e inseridos na função de ativação referente à próxima camada, como indicado na forma geral do neurônio em (1).

$$Y = \text{ativação} \left(\sum [\text{peso} * \text{entrada}] + \text{bias} \right) \quad (1)$$

O intuito de utilizar uma função de ativação é justamente para tornar o modelo capaz de resolver problemas mais complexos, pois elas servem para tornar a soma de pesos e *bias* em uma função não-linear. Além disso, cada função de ativação tem o seu valor específico para uma determinada funcionalidade, sendo algumas mais úteis para problemas de classificação, como a função sigmóide, ou para problemas em que há neurônios deficientes na rede, como o *Leaky ReLU* (MACHADO, 2023), e algumas são úteis apenas em camadas ocultas, como o *ReLU*. Na

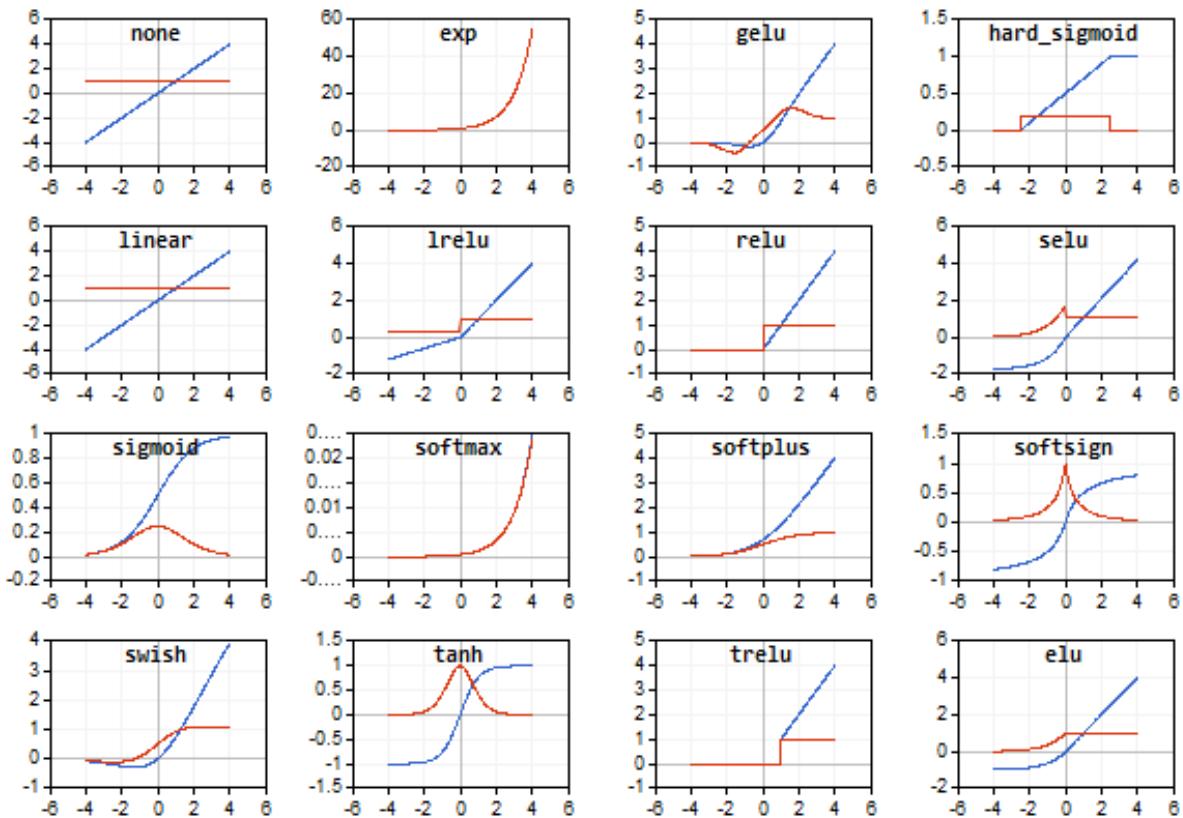
$$u_1 = u_0 - \alpha \nabla L(u) \quad (2)$$

é possível observar um resumo da forma que essas funções têm juntamente com as suas derivadas.

Na fase de propagação para trás é onde ocorre o processo de aprendizado da máquina e ela consiste em calcular os gradientes da função de perdas em relação aos pesos e *bias* dos neurônios, esses gradientes são utilizados para modificar os parâmetros e eventualmente reduzir o erro. A atualização dos pesos funciona como em **Erro! Fonte de referência não encontrada.** (2) e depende também de uma taxa de aprendizado definida pelo desenvolvedor da aplicação. Essa taxa dita a velocidade de aprendizado da máquina, porém, ela não pode ser infinitamente grande, pois sua magnitude pode afetar o quanto cada peso irá mudar e o valor ideal poderá nunca ser alcançado. Portanto, existe um valor ótimo para a taxa de aprendizagem que gerará uma aprendizagem rápida sem perder qualidade.

$$u_1 = u_0 - \alpha \nabla L(u) \quad (2)$$

Figura 4 – Funções de ativação usuais (azul) e suas derivadas (vermelho)



Fonte: (MetaQuotes, 2023)

Outro conceito importante no desenvolvimento da rede neural é o da função de erro, pois é o meio que permite que as mudanças ocorram na rede e, assim como as funções de ativação, é necessário definir a melhor função de erro para o problema em questão.

Para problemas de regressão, a função de erro médio quadrático (*MSE*) (3) se mostra muito eficaz e consiste em dividir o somatório das diferenças quadráticas pelo tamanho do vetor considerado.

$$L = \frac{1}{N} \sum_{i=0}^N (y_i - \hat{y}_i)^2 \quad (3)$$

Outra possibilidade é a função de entropia cruzada, que se mostra muito eficaz para problemas de classificação e consiste no somatório negativo dos produtos entre o valor rotulado e o logaritmo do valor predito pelo modelo, como mostrado na equação de *cross-entropy* (4).

$$L = - \sum_{i=0}^N \hat{y}_i * \log (y_i) \quad (4)$$

Por fim, pode-se utilizar também a média absoluta percentual, ou *MAPE* (5), que mesmo não sendo tão utilizada para redes neurais, vale comentar a respeito. Ela consiste em dividir pelo tamanho do vetor o somatório da razão entre a diferença do valor rotulado pelo valor predito e dividir pelo real.

$$L = \frac{100\%}{N} \sum_{i=0}^N \frac{|y_i - \hat{y}_i|}{\hat{y}_i} \quad (5)$$

2.4 Large Language Models

Uma aplicação recente e que tem se desenvolvido bastante no campo do aprendizado profundo é o processamento de linguagem humana e geração de conteúdo. Duas grandes áreas que juntas desempenham um papel importante na criação das inteligências artificiais generativas. Essas IAs são capazes de entender um prompt escrito por uma pessoa e gerar um texto, imagem, vídeo ou código de forma aparentemente criativa.

Elas também são chamadas de *LLMs* ou *Large Language Models*, (ZHAO, et al., 2023), que são grandes redes neurais em modelo de transformador treinadas com uma vasta quantidade de dados.

O diferencial das *LLMs* em relação aos modelos anteriores de aprendizado de máquina é a forma em que as palavras são representadas, pois, anteriormente, cada palavra era representada por uma tabela própria, o que não permitia o reconhecimento sequencial e nem de sinônimos. Já nos grandes modelos de linguagem, são utilizados vetores multidimensionais, assim palavras com significados parecidos ficam próximas no espaço vetorial.

Assim como outros modelos de aprendizagem de máquina, as *LLMs* possuem a possibilidade de *fine-tuning*, que permite o aprimoramento de um modelo para um uso específico. Considerando que o treinamento desses modelos é muito custoso dos pontos de vista computacional e consequentemente financeiro, torna-se inviável que eles sejam desenvolvidos em escalas pequenas e para aplicações específicas, logo, as grandes empresas que têm acesso vasto à tecnologia de ponta pré-treinam os modelos para serem capazes de realizar tarefas gerais, como entender a relação entre palavras, e o usuário com menos recursos pode utilizar um banco de dados menor para desenvolver uma aplicação específica.

Atualmente, existem diversas divisões para os *LLMs* (Sharma, 2025), uma possível é a seguinte:

- Modelos genéricos de linguagem: São capazes de predizer a próxima palavra baseado no treinamento.
- Modelos de representação de linguagem: Aqui são utilizados transformers e são capazes de realmente fazer o processamento de linguagem natural.
- Modelos de linguagem aprimorados por instrução: Nesse caso, os modelos são treinados para responder uma determinada forma de entrada.
- Modelos aprimorados para diálogo: Esses modelos são desenvolvidos de forma que eles são capazes de predizer diversas trocas de mensagem.
- Modelos multimodais: Esses são os modelos mais robustos atualmente e permitem o processamento de imagens juntamente com as capacidades de processamento de texto citados acima.

No tocante à aplicação, um agente de IA necessita de um modelo de linguagem para existir, sendo considerado o “cérebro” que realiza o processamento de informações. Além disso, para tornar os dados legíveis para o modelo, é necessário criar *embeddings* (PETUKHOVA & al, 2024), que são vetores criados para identificar os dados mais significativos de cada objeto.

2.4.1 *Transformers*

Os transformers são um grupo de redes neurais que possuem um codificador e um decodificador com propriedades de autodesenvolvimento (SHABANI, 2024), esse par é capaz de entender a relação entre as sequências de dados. Além dessa capacidade, essa arquitetura permite a autoaprendizagem, uma característica de modelos de aprendizagem não-supervisionada, facilitando ainda mais o seu uso para o entendimento de línguas e sequências de palavras.

Vale ressaltar que a arquitetura de transformadores permite o processamento em paralelo dos dados, tornando o aprendizado muito mais rápido e o número de parâmetros utilizados muito maior. A estrutura básica do modelo consiste em:

- Camada de incorporação: Onde cada palavra é convertida num vetor multidimensional e esse processo permite o modelo entender a semântica e sintaxe das palavras.
- Camadas de avanço: Onde ocorrem as transformações não-lineares com as funções de ativação e permitem o modelo a ter um maior nível de abstração no aprendizado.
- Camadas recorrentes: Onde o modelo se torna capaz de interpretar a sequência de palavras e entender a relação entre elas.
- Mecanismo de atenção: Essa característica é o que permite o modelo de definir os pontos mais importantes do texto, ou seja, ele “presta atenção”

Baseado em (Vaswani, et al., 2017), o codificador interpreta a sequência de símbolos (letras) e as transforma numa sequência de representação contínua, após isso o decodificador gera a sequência de saída gerando cada elemento de cada vez e o modelo sempre utiliza o símbolo gerado anteriormente como entrada auxiliar na geração de mais texto.

Na Figura 5 é possível observar um diagrama mostrando o como é estruturado o modelo de transformador. Na esquerda é analisado o input e sua estrutura é codificada para a parte da direita decodificar e realizar as operações. Todos os elementos do texto enviado à máquina são comparados entre si e são atribuídos pesos para cada relação, permitindo elencar cada relação. Vários processos com foco específico podem ser feitos em paralelo, ou seja, pode existir um transformer voltado para relações semânticas e outro para relações gramaticais.

Figura 5 – Diagrama da arquitetura de transformers

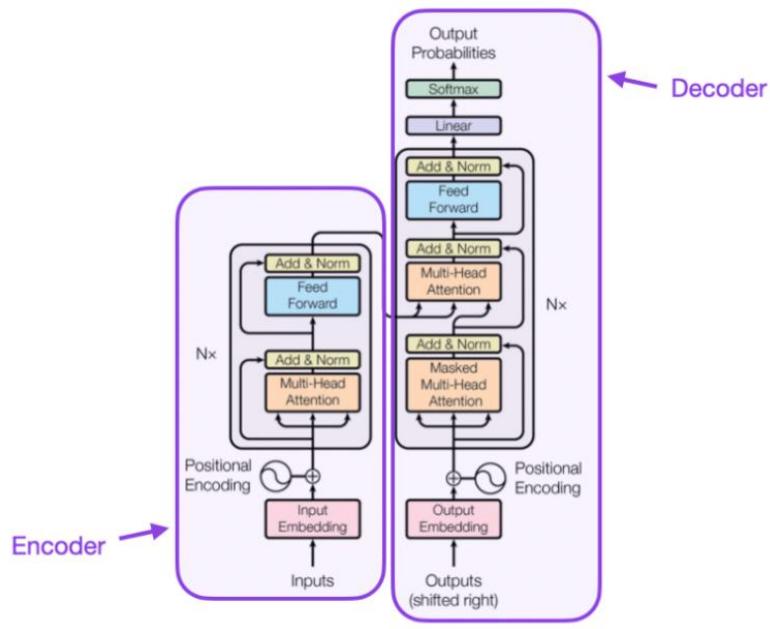


Figure 1: The Transformer - model architecture.

Fonte: (Vaswani, et al., 2017)

2.5 Agentes de Inteligência Artificial

Pode-se tratar o agente de IA como uma aplicação especializada em realizar uma determinada atividade ou função, segundo (YAO, et al., 2022), porém com o diferencial de ter como elemento primordial um modelo de linguagem. Portanto, ele se torna capaz de tomar decisões de forma ativa, diferentemente dos modelos padrões.

Ele é composto por 3 componentes principais, (RDD10+, s.d.):

- Modelo: O “cérebro” do agente, responsável pelo processamento de informações e pela tomada efetiva de decisões;
- Ferramentas (*tools*): As “mãos” do agente, responsáveis pelas ações do agente, podendo ser auxiliares para a geração de uma resposta ou meios cruciais no processo de geração de conteúdo. A exemplo de funções programadas pelos desenvolvedores ou alguma chamada de API que permita o agente acessar um banco de dados externo;
- Ambiente: basicamente, onde e como o modelo irá se comunicar com as ferramentas de maneira satisfatória.

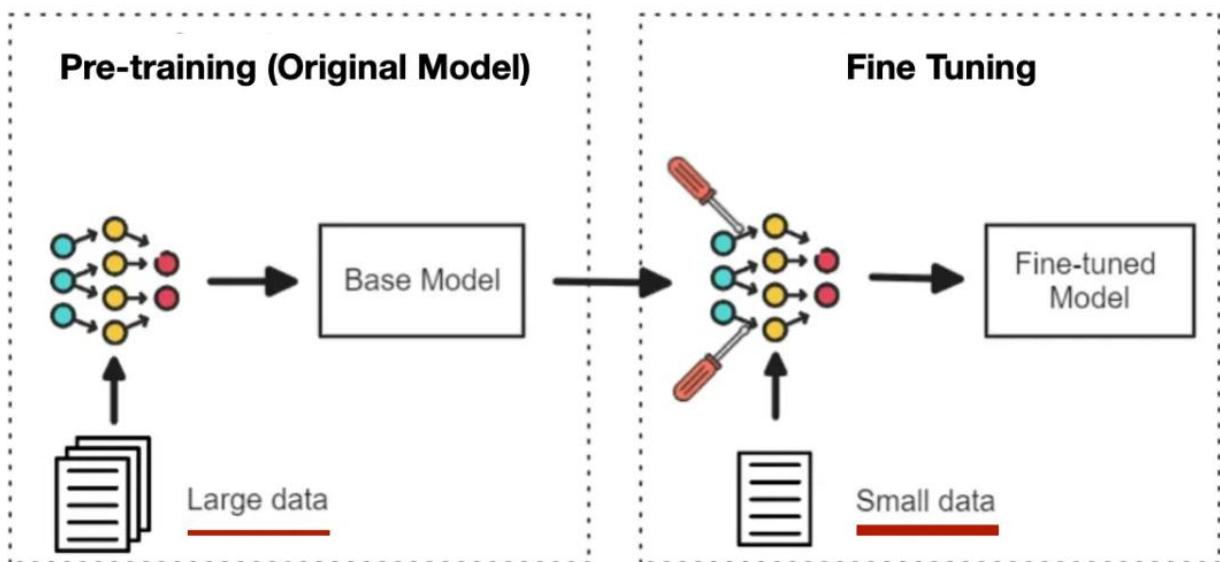
A criação de agentes pode se dar de duas maneiras diferentes, porém não excludentes. Primeiramente por meio do *Fine-Tuning*, que consiste em treinar uma parcela menor de um modelo já treinado com alguns dados de uma área específica, com intuito de tornar o agente um especialista em determinada função, possivelmente diminuindo a capacidade generalista do modelo. Tende a ser um processo mais custoso do ponto vista computacional e altera o modelo diretamente. A segunda forma é por meio da recuperação com base em contexto, nesse caso, o modelo tem acesso a um banco de dados e pode fazer consultas de acordo com o que lhe foi solicitado. Assim, o agente também ganha caráter especialista, mas sem alterar o modelo de fato. É possível combinar ambos os métodos, ou usá-los de forma independente, dependendo somente da aplicação desejada e dos recursos computacionais que o desenvolvedor tenha acesso.

2.5.1 Fine-Tuning

Como introduzido e presente na Figura 6, o *fine-tuning* consiste em treinar o modelo novamente com um conjunto de dados novo e específico para a função proposta, segundo (Anisuzzaman, 2025), e esse processo requer algumas etapas para ocorrer de forma satisfatória. O treinamento padrão de modelos segue o seguinte padrão:

1. Preparação dos dados: A etapa mais importante, consiste em adquirir dados que façam sentido para a aplicação, tratá-los de maneira que as informações estejam dentro de um padrão legível pelo modelo, estejam isentos de inconsistências ou erros e divididos de forma a validar o processo

Figura 6 – Diagrama do *fine-tuning*



Fonte: (FENEK, 2025)

2. Escolha de *framework* e modelo: É crucial definir a *framework* que seja capaz de lidar com todos os requerimentos do projeto. Também é de vital importância definir um modelo a ser treinado, baseado nas limitações impostas pelo projeto e nas necessidades da aplicação. Ou seja, modelos mais antigos ou com

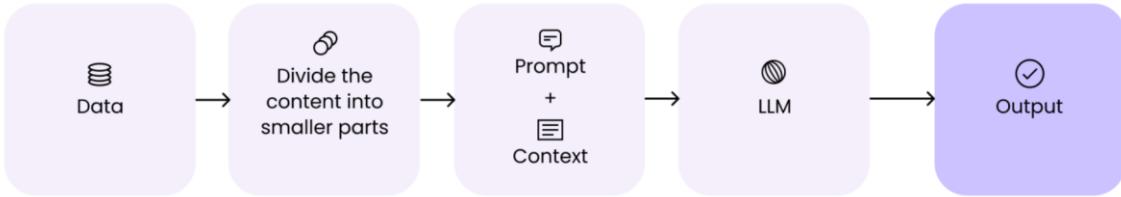
menos parâmetros, tendem a ser menos eficazes, mas podem ter suas aplicações, dependendo das necessidades do desenvolvedor.

3. Definição de hiperparâmetros: É a etapa imediatamente anterior ao treinamento, onde será definido a taxa de aprendizagem do modelo, o número de lotes de dados passados ao modelo, o número de épocas (vezes) que os dados serão passados ao modelo e definir o otimizador, que é o algoritmo usado para atualizar os pesos do modelo durante o treinamento.
4. Validação: Após o treino, deve-se validar o processo de alguma forma, ou seja, é necessário definir métricas que possibilitem validar o novo modelo.
5. Lançamento para produção e ajustes: Para finalizar, ao lançar o modelo, é necessário validar como o agente irá se comportar no mundo real e refinar o seu comportamento baseado num *feedback*.

O fine-tuning de um modelo de IA generativa diverge no tamanho da base de dados utilizada no treinamento, pois deve ser muito menor, já que não há necessidade de solucionar questões gerais, além disso, há um direcionamento dos tipos de perguntas utilizados no treinamento. Esses dois pontos possibilitam o modelo performar melhor em questões mais específicas.

2.5.2 RAG (*Retrieval Augmented Generation*)

O RAG (Geração Aumentada de Recuperação) consiste em buscar dados num banco de dados e disponibilizá-lo para uma LLM, segundo (Savage T, 2025), ou seja, dados importantes são pré-selecionados, de forma a melhorar a resposta do modelo de linguagem. Essa prática reduz drasticamente as chances de alucinações e a intensidade delas também. A alucinação ocorre quando a LLM não consegue encontrar fontes para uma pergunta e cria uma resposta não condizente com o que foi perguntado. Na Figura 7, é possível ver um diagrama explicativo do processo por trás do RAG.

Figura 7 - Diagrama do *RAG*

Fonte: (WEI, 2023)

Eventualmente, será feita uma busca vetorial desses dados armazenados e os *embeddings* são de suma importância para encontrar o que se busca.

Além da tradução, os dados também são divididos em partes menores antes de serem enviados, facilitando o processo de lançamento e acelerando o processo como um todo.

O *RAG* possui característica crucial no desenvolvimento do trabalho, pois permite uma busca rápida por conteúdos confiáveis dentro de um banco de dados, permitindo que o agente refcrcie conteúdos de grande valor técnico, como a NBR 5410 (ABNT, 2004), sem custar muito poder de processamento do modelo.

2.6 Engenharia de *Prompt*

Ao utilizar um modelo de linguagem já pronto ou ao criar um agente, sempre é importante realizar a engenharia de *prompt* (WHITE, et al., 2023) quando se faz um requerimento ao modelo. Esse processo, consiste em entender como a máquina pensa e escrever a mensagem da melhor forma possível para que ela retorne uma resposta coerente e evite alucinações. Além disso, existem parâmetros associados à resposta em si, como:

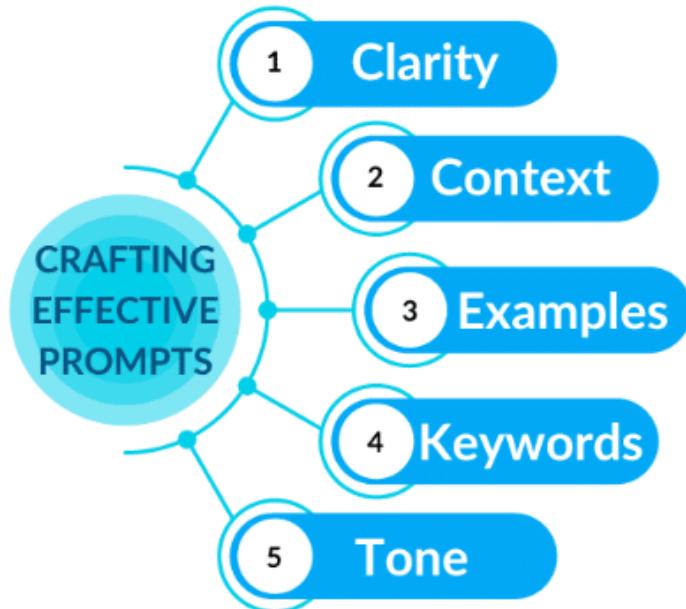
- Top K/ Top P: Define quais escolhas de palavra o modelo dever ter, com valor máximo de 1, se for escolhido um valor alto, o modelo irá retornar somente os termos mais prováveis, ao baixar, irá retornar algo mais imprevisível;
- Temperatura: Define a criatividade do modelo, sendo extremamente lógico no 0 e muito criativo no 1.

As principais estratégias de construção de *prompts* (WANG, et al., 2022) são:

- *0 shot*: O comando é dado diretamente e sem exemplificação;
- *Fuse-shot*: Alguns exemplos são dados ao modelo e ele aprende pela demonstração;
- *Contexto*: São dadas informações importantes ao modelo, permitindo-o usar essas referências para responder;
- *Persona*: Cria-se um personagem que o modelo deve se espelhar, como um roteiro que permite o modelo responder algo mais específico, ainda requer algum contexto;
- *Step back*: Encoraja o modelo a “dar um passo para trás” e examinar o todo, incrementando a resposta;
- *Linha de pensamento*: O modelo será guiado a responder o pedido, sempre explicando a própria lógica até chegar no resultado;
- *Self-consistency*: O modelo recebe a mesma requisição diversas vezes, criando diferentes respostas e removendo enviesamentos.

De maneira geral, deve-se manter a comunicação clara, sem considerar que está se conversando com um especialista, experimentar estilos de contato diferentes, entender a limitação temporal do modelo e mostrar um modelo de resposta satisfatório. Na Figura 8, vê-se um diagrama exemplo das boas práticas. Um prompt bem escrito requer:

- Clareza: um texto bem escrito que diminua outras interpretações;
- Contexto: pode ser necessário situar o modelo numa situação específica, mostrar elementos que facilitem a interpretação da requisição;
- Exemplos: para o modelo ter referência do que é pedido;
- Palavras-chave: podem ser úteis para buscas e entendimento do contexto;
- Tom: se há a necessidade de uma resposta formal, melhor que a requisição seja formal.

Figura 8 - Engenharia de *prompt*

Fonte: (Otten, 2023)

2.7 Eficiência energética

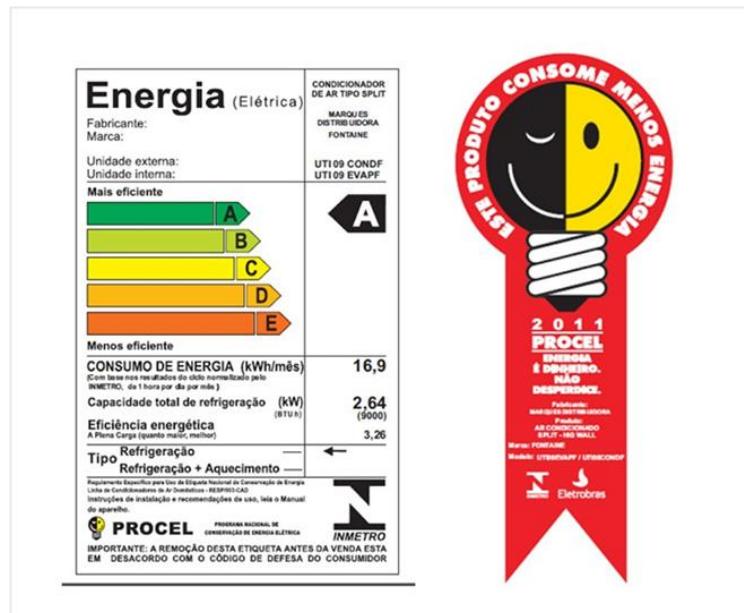
A preocupação com suprimento de energia tem se tornado algo cada vez mais notório e com isso novas soluções foram encontradas para lidar com a escassez de recursos. A primeira ideia é gerar mais energia, porém é uma solução custosa e que demanda muito tempo de desenvolvimento e construção de usinas. Em paralelo a essa solução, tem sido desenvolvidos equipamentos capazes de consumir menos e desempenhar melhor, surgindo assim o conceito de eficiência energética.

No quesito ambiental, o Brasil é referência no âmbito da geração limpa de energia, porém existe ainda uma grande quantidade de usinas termoelétricas instaladas e elas acabam sendo mais poluentes e mais custosas para a população. Existe um esforço grande no desenvolvimento das matrizes eólica, solar e de biomassa para aumentar a capacidade instalada total do país. O aumento do consumo energético nacional e global exige que novos meios de geração sejam sempre desenvolvidos e somar esse fator às melhorias dos equipamentos visando eficiência torna o objetivo final mais fácil de alcançar.

De acordo com (INMETRO, 2021), no brasil, começou uma discussão em 1984 sobre o tema de conservação da energia e assim surgiu o Programa Brasileiro de Etiquetagem (PBE) e ele deixa claro dados de desempenho dos produtos visando uma melhor eficiência e auxiliando o consumidor na escolha dos seus equipamentos. Após isso, surgiram o Selo Procel e o Selo CONPET, ambos classificam os equipamentos baseados numa nota e eles reconhecem equipamentos mais eficientes, no caso do Procel existe um foco em energia elétrica e para o CONPET o foco é nos derivados de petróleo e gás natural. Na Figura 9, pode-se ver o Selo Procel e suas categorias.

Ao utilizar equipamentos mais eficientes, o consumidor consegue economizar energia nos seus processos cotidianos, contribuindo para o meio ambiente e diminuindo os próprios custos ao longo do tempo. Esse freio de demanda que a eficiência energética possibilita, permite que o esforço em geração de energia seja mais controlado, já que ao longo do tempo, a energia economizada em processos equivale a uma usina elétrica virtual. Esse conceito consiste em fazer uma comparação entre o a quantidade de energia que seria demandada sem o uso de equipamentos eficientes num determinado tempo e a quantidade de energia demandada utilizando tais equipamentos, nota-se que com o passar do tempo, a distância entre as demandas se torna cada vez maior e esse diferencial pode ser usado em outras áreas.

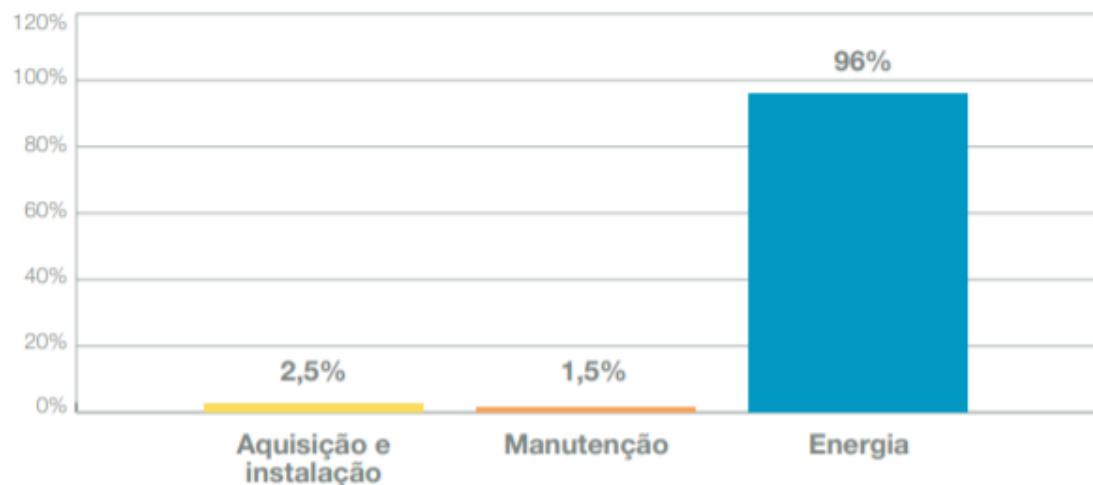
Figura 9 - Selo Procel



Fonte: (Litro de Luz, 2022)

Considerando o fator o financeiro, nota-se uma relação interessante entre os custos de um equipamento e o seu uso. No exemplo de um motor elétrico, o principal custo ao longo da vida útil está associado à energia consumida em operação, superando em muito o custo de aquisição e manutenção, como ilustrado na Figura 10. Essa tendência também se verifica em outros equipamentos como geladeiras, ar-condicionados, luminárias, motobombas etc.

Figura 10 – Custo de um motor elétrico após 10 anos



Fonte: (CanalEnergia, 2020)

2.7.1 Qualidade de energia

Um conceito muito associado à eficiência energética é o de qualidade de energia. A preocupação em saber se energia recebida é boa interfere diretamente nos custos de uma residência, indústria ou estabelecimento. Para isso, é importante definir bem a demanda do lugar e entender se a concessionária está fornecendo de forma ideal.

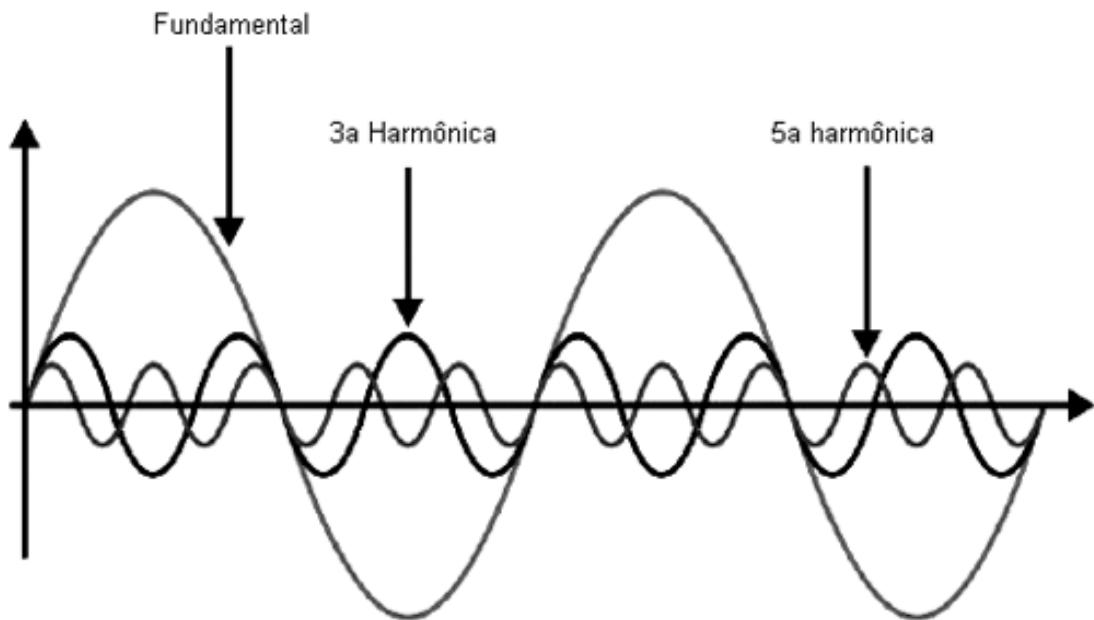
Portanto, algumas características são consideradas segundo (SILVA, IBAROLA, NÓBREGA, & MARTINS): o fator de potência, tensão em regime permanente, presença de harmônicos, desequilíbrio de tensão e variações de frequência. O conjunto de características negativas pode afetar o consumo, tornando necessário investir em soluções que contornem esses problemas.

O fator de potência pode ser definido como a razão entre potência ativa e reativa do sistema. A potência ativa é aquela que de fato se transforma em outras formas de energia e seu peso no fator de potência deve ser bem maior que o da potência reativa. Existe um controle que exige que o fator de potência se mantenha menor que 0,92 indutivo ou capacitivo, se não uma multa é aplicada no estabelecimento (ANEEL, 2010).

A presença de harmônicos também é um problema recorrente nas instalações, segundo (MARTENS & SILVA), e podem afetar gravemente a vida útil dos aparelhos. Uma harmônica pode ser definida como um sinal senoidal em que a frequência é um múltiplo inteiro da frequência fundamental, modificando a forma de onda que chega no equipamento. Um exemplo de harmônicas pode ser visto na Figura 11.

Se a taxa de distorção harmônica for muito alta, além de consumir mais energia e danificar os equipamentos, ela também pode gerar medições incorretas, dificultando o trabalho de correção pelo profissional.

Figura 11 – Harmônicas de 3^a e 5^a ordem



Fonte: (Braga, s.d.)

3 DESENVOLVIMENTO E METODOLOGIA

3.1 METODOLOGIA

Como a ideia é criar um agente de inteligência artificial capaz de auxiliar no controle de consumo energético de uma residência e gerar possíveis recomendações de melhorias, o projeto foi dividido em 8 etapas:

1. Definições iniciais: ferramentas, linguagens, bibliotecas, *frameworks* e modelos de *LLM* necessários para criação de um agente de inteligência artificial;
2. Pesquisa: documentos, artigos científicos, artigos na internet, normas técnicas, guia de boas práticas;
3. Treinamento nas ferramentas propostas;

4. Geração de dados personalizados para testes;
5. Montagem dos *workflows* no programa escolhido e criação do banco de dados;
6. Treinamento do agente por *fine-tuning*;
7. Testes com o *RAG*. Nessa etapa, foram realizados teste comparativos entre os modelos do agente apenas com acesso ao banco de dados e testes com o modelo com *fine-tuning*.
8. Análise financeira do projeto.

3.2 DEFINIÇÕES INICIAIS

O primeiro passo foi entender o problema e por quais meios o objetivo seria atingido. Assim, cada etapa do processo necessitou de ferramentas específicas, que fossem capazes de gerar os resultados necessários.

As linguagens de programação definidas foram o *Python* e *JavaScript*. A primeira possui uma vasta variedade de bibliotecas para tanto processamento de dados, quanto treinamento de modelos de inteligência artificial dos níveis mais primitivos aos mais avançados. A segunda, pois apresenta boa compatibilidade com outras ferramentas utilizadas no projeto.

As principais ferramentas utilizadas consistem no *n8n*, *Pinecone* e *Ollama*. O *n8n* é uma ferramenta de criação de *workflows* e possui características que permitem uma integração satisfatória entre diversos elementos do projeto, partindo de um *chat* até a busca de elementos num banco de dados e conexão direta com os modelos de IA. A escolha do *n8n* como ferramenta de orquestração se deu por sua possibilidade de uso de maneira gratuita sem grandes limitações e possuir uma variedade grande de integrações com diversos serviços. Para utilização do *n8n* na sua versão de comunidade *open-source*, foi necessário o uso da ferramenta de conteinerização *Docker*. O *Pinecone* serviu como o banco de dados vetorial, pois possui integração com o *n8n*, se mostrou prático na implementação com uma interface de uso simples e plano gratuito de uso abrangente. O *Ollama* é um programa que serve como ambiente propício para utilizar um modelo *open-source* de maneira local, possuindo uma interface de usuário e possibilitando seu uso no fluxo do *n8n*.

Ferramentas secundárias que foram importantes para algumas etapas do processo foram: o *Google Collab* para auxiliar no *fine-tuning* do modelo, pois disponibiliza um ambiente potente com *GPUs*, possibilitando um treinamento de modelo mais eficiente; e o *BEOpt* para criar dados de simulação de consumo energético de diversas residências com tamanhos diferentes, modelos de comportamento mais racionais ou não com a energia elétrica e equipamento variados.

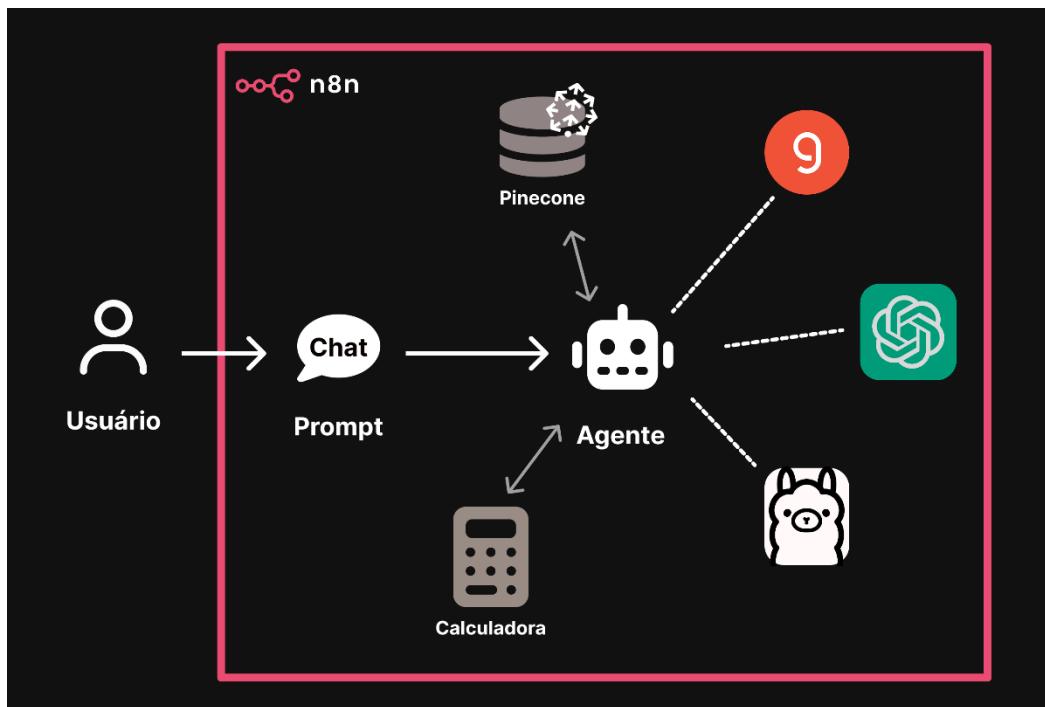
A *framework* de treinamento do modelo foi o *Unsloth*, especializado para *fine-tuning*.

Os modelos usados foram:

1. *Llama 3.2 1B*: esse modelo foi escolhido por ser *open-source*, o que permite o treinamento personalizado para o projeto e possui um requerimento de processamento não muito custoso para rodar na máquina local. Considerando que os testes finais ocorreram numa máquina de processador i7 e 16Gb de memória RAM sem placa de vídeo ou GPU dedicadas, que são especificações suficientes, mas não ideais. Testes realizados numa máquina com menor poder de processamento resultaram em demora na resposta e aquecimento da máquina;
2. *OpenAI 4.1-mini*: esse modelo foi utilizado numa etapa específica da busca de vetores do banco de dados, pois é recomendado o uso de um modelo mais potente nessa etapa. Além de ser usado como modelo comparativo;
3. *OpenAI text-embeddings-small*: esse modelo foi usado para criar os *embeddings* do banco de dados;
4. *Groq (llama 4)*: é um serviço de API que permite acessar diversos modelos de forma gratuita. Esse serviço foi usado para realizar testes no *workflow* e servir como comparação, pois é menos custoso computacionalmente e permite mais usos que os principais modelos do mercado de forma individual. O modelo escolhido para os testes foi à princípio o *llama 3.2*, mas optou-se pelo *llama 4* justamente por sua maior capacidade e tecnologia.

Na Figura 12, pode-se ver um diagrama geral do processo de funcionamento do agente.

Figura 12 – Diagrama geral do agente



Fonte: própria

3.3 PESQUISA

Como citado anteriormente, para desenvolver um *RAG* robusto e funcional, é necessário alimentar um banco de dados com diversos materiais específicos para a aplicação desejada. O principal foco do agente criado, além de auxiliar com mitigação de falhas, é sugerir formas de tornar a residência em questão mais eficiente do ponto de visto energético e financeiro.

Com tal objetivo em mente, os conteúdos escolhidos foram listados na Tabela 1 .

Tabela 1 - Tabela resumo de conteúdo inserido no RAG

Documento	Tipo	Tamanho (caracteres)	Fonte
Como economizar com ar-condicionado	Artigo internet	8748	https://blog.portoseguro.com.br/como-economizar-energia-com-ar-condicionado-7-dicas

Impacto troca de equipamento	Texto próprio	1658	-
Lâmpadas LED x fluorescentes	Artigo internet	5251	https://www.enel.com.br/pt-saopaulo/midia/news/d2025-4/Lampadas-eficientes-como-escolher-a-melhor-opcao-para-economizar-energia.html
Medidores de energia	Artigo internet	4528	https://loja.br.abb.com/blog/post/medidores-de-energia?srsltid=AfmBOoqd2L8U3OEB60P_EQ29P_jMy-2CHneIAOC2Ewsnu1O_mKhIZ8sVZ
Payback energia solar	Texto próprio	1053	-
Cálculos conta de luz	Artigo internet	3056	https://meufinanciamentosolar.com.br/dicas/para-voce/calcular-de-consumo-de-energia
Vantagens energia solar	Arquivo internet	7974	https://www.portalsolar.com.br/energia-solar-vantagens
Eficiência Energética no Brasil	Publicação	67389	https://www.climatepolicyinitiative.org/pt-br/publication/eficiencia-energetica-ee-no-brasil-e-no-mundo-mecanismos-das-politicas-de-ee-em-unidades-consumidoras-intensivas-de-eletricidade/
Lista de equipamentos	Texto próprio	1399	-
Eficiência Energética em Iluminação Pública	Artigo acadêmico	27461	(Sampaio, 2023)
Guia de manutenção	Artigo internet	20359	https://www.triider.com.br/blog/guia-sobre-manutencao-eletrica/?utm_source=chatgpt.com
Elementos projeto elétrico	Artigo internet	24464	https://www.aditivocad.com/blog/roteiro-projeto-elettrico-residencial/
Tipos de manutenção	Artigo internet	2930	https://ampher.com.br/qual-a-hora-de-fazer-uma-manutencao-no-quadro-de-energia/

Riscos projeto elétrico	Artigo internet	5080	https://www.orcafascio.com/papodeengenheiro/projeto-elettrico-mal-feito-obra-principais-riscos
Periodicidade de manutenção	Artigo internet	3639	https://prometalepis.com.br/o-que-e-manutencao-rede-eletrica/#:~:text=O%20que%20diz%20a%20NBR,e%20seguir%20os%20padr%C3%B5es%20exigidos.
Projeto elétrico	Artigo internet	3875	https://materiais.tomas.weg.net/guia-do-projeto-elettrico
NBR 5410	Norma técnica	254052	(ABNT, 2004)
Planilha de tempo de manutenção	Texto próprio	1064	-

Fonte: própria

Tais elementos foram introduzidos num banco de dados vetorial e convertidos em elementos mais “palatáveis” para uma máquina interpretá-los através de um *workflow* específico.

3.4 TREINAMENTO NAS FERRAMENTAS

Antes de implementar a metodologia de agentes e treinamento de modelos, foi necessário praticar o uso principalmente do *n8n* e *Ollama*. A fim de usar a ferramenta de criação de *workflows* de maneira gratuita, foi necessário o uso da ferramenta de conteinerização *Docker* e integrar o *Ollama* da máquina local.

O contêiner foi criado através de uma imagem do *n8n* disponível gratuitamente. Para abrir o *n8n*, sempre foi necessário utilizar o comando *docker-compose up -d* no terminal dentro da pasta onde a imagem foi criada. Tal versão de comunidade apresenta a limitação de não ser o modelo mais atualizado e não permite uma escalabilidade, porém é uma versão extremamente capaz nas etapas de treino e testes dos modelos.

O treinamento consistiu, primeiramente, em criar *workflows* de uso mais geral e aos poucos introduzir as ferramentas de: pesquisa de material proveniente de um

drive, banco de dados vetorial, módulo de agentes com modelos prontos e uso de ferramentas próprias para *RAG*. Por fim, os modelos prontos foram alternados com o modelo próprio que funcionou localmente graças ao *Ollama*.

3.5 GERAÇÃO DE DADOS

Houve a necessidade de gerar dados de residências diversas para auxiliar no treinamento do modelo. A ideia foi utilizar a ferramenta *BEOpt*, pois ela permite a criação de modelos energéticos específicos, como inserir eletrodomésticos, sistemas de geração *off-grid*, selecionar o tipo de iluminação ou isolamento térmico, entre outras funcionalidades.

O uso da ferramenta consistiu em sempre gerar pelo menos duas residências de tamanho igual com perfis energéticos e equipamentos parecidos, mas com mudanças pontuais. Por exemplo: criar uma residência pequena de 33 m² sem energia solar e outra com, ou uma residência de 45m² com uso de ar-condicionado central e outra que funcionava apenas com ventiladores. Foram gerados 40 cenários diferentes entre residências variando de tamanhos entre 33m² e 179m², possuindo dados de consumo diário de energia, horários de funcionamento de equipamentos e custo monetário de gastos energéticos.

Em posse dessas simulações, visando uma variedade de cenários diferentes, incluindo tamanhos de ambientes diferentes e perfis energéticos distintos, foi possível gerar dados suficientes para iniciar o processo de treinamento do modelo local.

3.6 FINE-TUNING

Como já explicado na seção 2.5.1, o *fine-tuning* consiste resumidamente em treinar um modelo de linguagem com dados específicos visando tornar o modelo mais especialista num determinado assunto. Os dados base foram os gerados pelo *BEOpt*, cerca de 40 exemplos, consistindo nos dados gerados associados a uma análise esperada como resposta.

Entretanto, por ser um processo computacionalmente custoso, foi necessário o uso do *Google Colab*, pois o ambiente permite o uso de *CPUs* e *GPUs* mais potentes

do que a máquina local do estudo, além de ser um ambiente usualmente utilizado para aplicações em *python*, especificamente de *machine learning*.

A *framework* de treinamento selecionada foi o *Unsloth*, ferramenta usual especializada em treinamento de modelos de linguagem. O processo consistiu basicamente na importação do modelo base, tokenização dos dados para torná-los mais legíveis para um computador e treinamento baseado nos dados fornecidos e resposta esperada em relação a esses dados. Alguns hiperparâmetros precisaram ser configurados para buscar um melhor treinamento do modelo:

- max_seq_length: 1024;
- LoRA rank: 64;
- lora_alpha: 128;
- gradient_accumulation_steps: 4 (indica batch_size de 8);
- num_train_epochs: 3;
- learning_rate: 2e-4;
- context_size: 200.

Após esse procedimento, o modelo pôde ser baixado e usado na aplicação.

3.7 WORKFLOWS E BANCO DE DADOS

Houve uma divisão do projeto em duas etapas principais: alimentação do banco de vetorial e criação do fluxo do agente. Assim, dois *workflows* separados foram desenvolvidos.

3.7.1 Workflow de alimentação do RAG

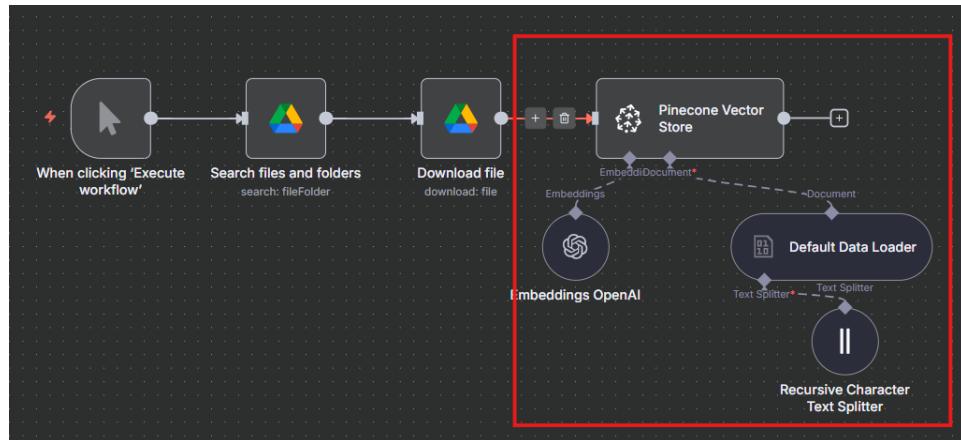
O fluxo mostrado na **Erro! Fonte de referência não encontrada.** é separado nas etapas de inserção de dados numa fonte compatível, no caso o *Google Drive*, e numa etapa de conversão dos dados em tokens legíveis para a máquina que são armazenados dentro do banco de dados vetorial do *Pinecone* com *batch size* dos embeddings configurado em 512.

Os conteúdos pesquisados para enriquecimento da fonte de dados do *RAG* foram todos inseridos no *drive* com uma nomenclatura que permitisse a sua inserção em lotes dentro do banco. A necessidade de lotes se deu por limitações do processo de envio de dados ao *Pinecone*.

A área marcada em vermelho na Figura 13Figura 13**Erro! Fonte de referência não encontrada.** consiste justamente na etapa crucial de envio. O bloco chave nomeado “*Pinecone Vector Store*”, que além das credenciais e destino dos dados, requisita dois elementos: o modelo de geração dos *embeddings* (vetores de dados), que no caso foi escolhido o modelo da *OpenAI*; e as regras de separação dos documentos em vetores.

Os vetores foram separados por documento específico e a cada 1000 caracteres, possuindo uma intercessão máxima de 250 caracteres entre dois vetores.

Figura 13 – *Workflow* de alimentação

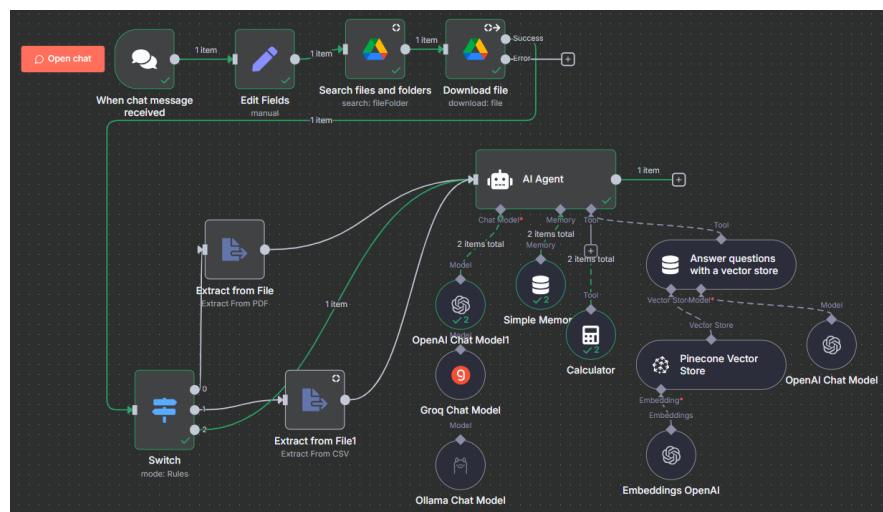


Fonte: própria

3.7.2 Workflow principal do agente

O *workflow* principal é mostrado na Figura 14, possui maior complexidade e se mostrou um desafio na construção de alguns de seus elementos. Pode ser separado em duas etapas: envio da mensagem com possível leitura de dados provenientes do *prompt* ou do *drive*; e o bloco do agente com seus blocos auxiliares.

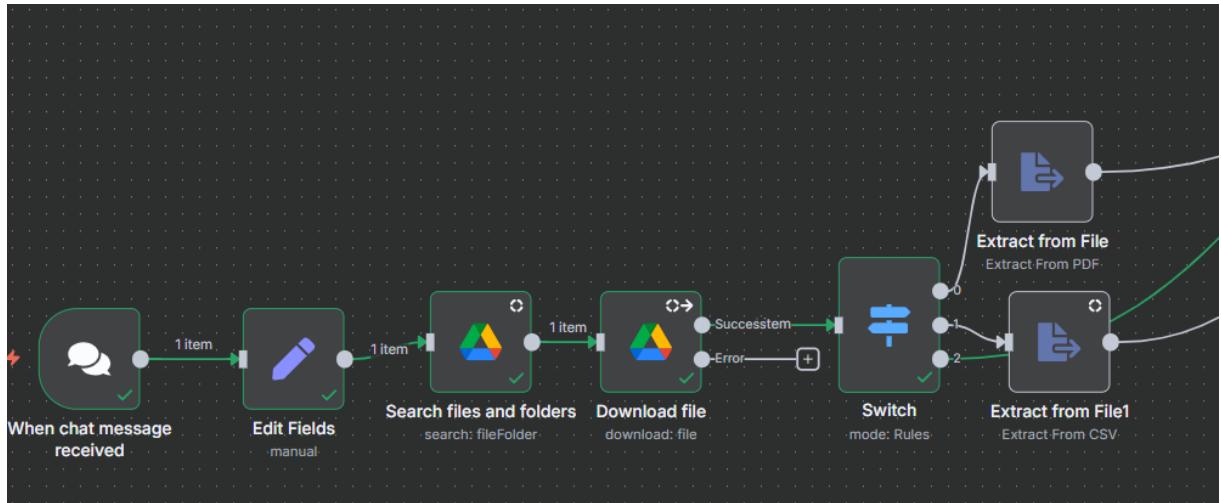
Figura 14 – Visão geral do *workflow* principal do agente



Fonte: própria

Na etapa nomeada de *prompt* presente na Figura 15, o modelo recebe uma requisição por meio de um *chatbox*, passa por uma etapa de formatação da mensagem, o fluxo avalia se possui dados no *drive* e a mensagem inteira é enviada ao agente. Os dados da mensagem podem estar tanto no corpo inicial do *prompt*, quanto num arquivo nomeado *prompt* dentro do *drive* nos formatos .pdf ou .csv.

Figura 15 – Etapa de *prompt*



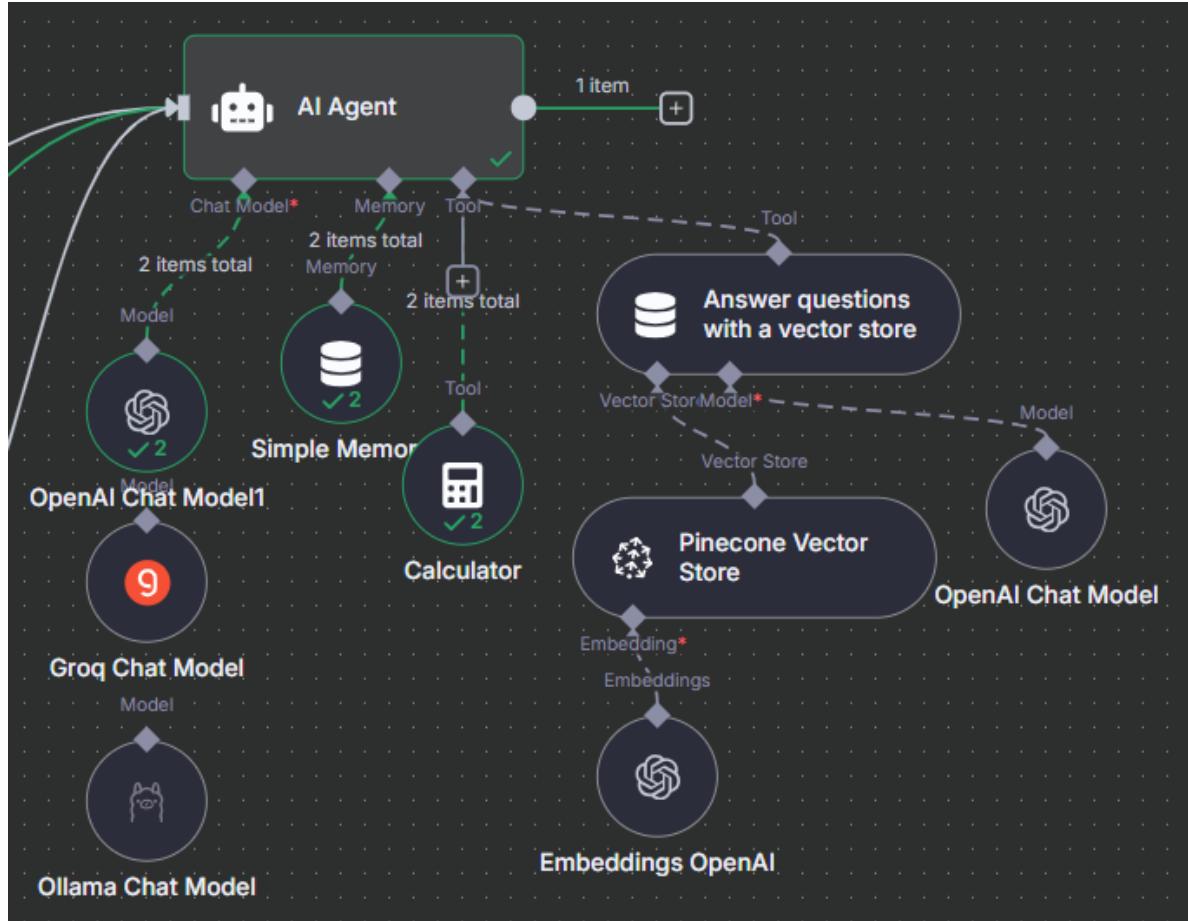
Fonte: própria

O bloco nomeado “*Switch*” faz a função de filtro do arquivo, possibilitando a conversão dos dados de acordo com seu tipo ou, se o *drive* estiver vazio, o envio direto da mensagem para o modelo.

Na etapa nomeada de modelo presente na Figura 16 pode-se ver o “cérebro” do agente de fato sendo ele o bloco mais importante de ambos os fluxos. O bloco “*AI Agent*” exige apenas 1 elemento acessório, mas para ter melhor aproveitamento, foi necessário inserir elementos auxiliares que possibilitam mais ações ao agente.

No componente de memória foi importante inserir um bloco que permitisse o agente ter acesso às mensagens passadas enviadas pelo usuário, permitindo que uma conversa seja criada e o agente “lembre” do contexto caso necessário. O bloco “*Simple Memory*” foi escolhido e configurado com uma janela de contexto de 5 mensagens, porém, se necessário, essa janela pode ser expandida.

Figura 16 – Etapa do modelo



Fonte: própria

No componente de ferramentas acessórias, foi vista uma necessidade de incluir dois elementos: a calculadora, visando diminuir a necessidade de processamento de cálculos matemáticos simples pelo modelo selecionado; e a conexão com o banco de dados do *Pinecone*. A segunda ferramenta age como uma ferramenta de busca e uma “estante de livros”, que permite o modelo consultar informações e direcionar o processamento da requisição. Como pode-se ver na Figura 16 **Figura 16: Exemplo de interface de usuário da aplicação**, existem dois modelos da *OpenAI* associados a esse processo, o bloco “*OpenAI Chat Model*” serve como a ferramenta de busca de fato, sendo importante que esse modelo seja potente o suficiente para fazer as buscas dentro do banco de dados, o modelo escolhido foi o *ChatGPT 4.1-mini* para todos os casos de teste. O bloco “*Embeddings OpenAI*” tem a função de interpretação dos vetores.

O componente de modelo age de fato como o “cérebro” do agente, é onde o processamento de informações é realizado. Associando o modelo com suas ferramentas, o bloco principal do agente ainda possui elementos que permitem somar ao *prompt* do usuário, como uma criação de persona específica.

3.8 TESTE COM O RAG

Nessa etapa, os 3 modelos selecionados (*llama 4*, *chatgpt 4.1-mini* e *llama 3.2 fine-tuned*) foram testados individualmente com as mesmas 17 perguntas, abrangendo temas de manutenção preventiva, análise de gastos energéticos, uso de geração solar, entre outros. Além disso, houve a necessidade de criar uma persona, uma espécie de inspiração para o agente, ou um roteiro a ser seguida. Tal persona foi a de um engenheiro eletricista especializado em gestão energética residencial.

O intuito de testar os modelos foi compreender as vantagens e desvantagens de cada um, avaliando se as perguntas conseguiam ser respondidas de maneira satisfatória e o tempo de processamento das requisições. Na Tabela 2, pode-se ver os critérios de avaliação do modelo. O julgamento foi realizado baseado em: a resposta estar tecnicamente correta, mesmo trazendo respostas curtas ou longas; coerência do texto, se o modelo se mantém respondendo ao tema sem divagar; clareza do texto, se o que é dito pelo modelo consegue ser bem compreendido; uso correto dos dados, se quando dados são fornecidos, o modelo consegue compreendê-los e utilizá-los para responder.

Tabela 2 - Critérios de avaliação

Tecnicamente correta	Coerência	Clareza	Uso adequado de dados (quando fornecido)
0-1	0-1	0-1	0-1

Fonte: própria

3.9 ANÁLISE TÉCNICA

Na Tabela 3, pode-se ver cada pergunta proposta, a avaliação subjetiva de cada resposta, o tempo de execução de cada etapa e a pontuação atribuída a cada pergunta segundo os critérios apresentados.

Tabela 3 – Tempo de geração e qualidade da resposta gerada por cada modelo

Tempo de geração e qualidade da resposta gerada por cada modelo			
Perguntas	llama-4-scout-17b-16e	ChatGPT 4.1-mini	llama 3.2 fine-tuned
Quais os principais equipamentos responsáveis pelos picos de consumo de uma residência?	Reposta satisfatória 9,572s	Reposta satisfatória 8,178s	Resposta incorreta 29min28s
Como identificar sobrecarga em um circuito a partir dos dados de corrente e tensão coletados?	1,00 Reposta ok com pequeno erro de português e leve erro técnico	0,87 Reposta satisfatória	0,00 Reposta correta em alguns pontos, mas com alucinações
Qual é a diferença entre consumo base e consumo sazonal, e como detectá-la nos dados históricos?	3,716s 0,83	7,56s 1,00	1min42s 0,10
	Reposta satisfatória 3,689s	Reposta satisfatória 8,076s	Resposta ok seguida de alucinações 1min57s

	0,93	1,00	0,30
	Reposta satisfatória	Reposta satisfatória	Resposta ok seguida de alucinações
Que tipo de sensor devo usar para medir o consumo individual de cada circuito do quadro?	3,659s	26,48s	1min20s
	0,97	1,00	0,20
	Reposta satisfatória	Reposta satisfatória	Reposta satisfatória
Qual é o potencial de redução de consumo se eu automatizar a iluminação com sensores de presença e dimerização?	3,276s	17,451s	59,483s
	1,00	0,93	0,93
	Reposta tecnicamente errada para elétrica, mas coerente para TI	Reposta tecnicamente insuficiente, mas com clareza	Respostas voltada para TI e não satisfatória
Como o modelo pode balancear carga entre circuitos para reduzir perdas em cabos?	4,875s	4,791s	1min27s
	0,50	0,73	0,10
	Reposta satisfatória	Reposta satisfatória	Não trouxe resposta
Como prever o melhor momento para ligar o ar-condicionado com base na temperatura e na tarifa de energia?	8,356s	8,602s	-

	Reposta satisfatória	Reposta satisfatória	Não trouxe resposta
Qual seria o impacto energético de trocar toda a iluminação por LED de alta eficiência?	4,398s	5,118s	-
	0,93	1,00	0,00
	Reposta satisfatória	Reposta correta, porém não explicitou os cálculos	Resposta com cálculos errados
Como o aumento da tarifa de energia em 8% ao ano impacta o custo total de operação em 10 anos?	3,936s	16,593s	10s
	0,93	0,77	0,00
	Reposta satisfatória	Reposta satisfatória	Reposta repleta de alucinações
Como o modelo pode prever o impacto da troca de um equipamento específico (por exemplo, geladeira) no consumo total anual?	5,071s	13,345s	1min28s
	0,87	0,93	0,00
O consumo mensal da residência é de 420 kWh. O ar-condicionado e o chuveiro elétrico representam 55% desse valor. Que estratégias técnicas e	Reposta boa com pequeno erro de português	Reposta satisfatória	Não trouxe resposta

comportamentais podem reduzir o consumo total em 20% sem perda de conforto?

3,954s

10,205s

-

0,90

0,93

0,00

Não trouxe resposta

Resposta ok, porém incompleta

Resposta repetitiva com alucinações

O medidor inteligente fornece dados de consumo em tempo real, e a concessionária adota tarifa branca. Como o agente pode controlar automaticamente equipamentos para minimizar a conta de energia?

-

16,562s

1min14s

0,00

0,77

0,00

Um sistema fotovoltaico de 5 kWp custa R\$ 25.000 e gera 600 kWh/mês. Considerando tarifa média de R\$ 1,00/kWh e aumento anual de 7%, qual é o payback estimado?

Resposta satisfatória

Resposta satisfatória

Resposta incorreta com alucinações

4,863s

21,144s

30,13s

0,95

1,00

0,00

Reposta satisfatória

Reposta satisfatória

Reposta incorreta com alucinações

Histórico de consumo (kWh/mês): 380, 400, 420, 450, 500, 520, 510, 480, 460, 430, 410, 390.

Como prever o consumo dos próximos três meses e explicar os fatores que influenciam a variação?

4,498s

38,782s

20,98s

1,00

1,00

0,00

Não trouxe resposta

Reposta satisfatória, mas poderia explicitar melhor o cálculo

Reposta incorreta com alucinações

O fator de potência médio é 0,82 e o consumo ativo é 10.000 kWh/mês. Qual a potência reativa necessária para elevar o FP para 0,95?

-

15,63s

1min02s

0,00

0,90

0,00

O consumo noturno médio é 0,6 kWh/h, mas subiu para 1,2 kWh/h sem alteração de hábitos. Como o agente pode

Reposta satisfatória

Reposta satisfatória

Reposta incorreta com alucinações

identificar a origem desse aumento?

20,203s

25,38s

2min

0,93

1,00

0,00

Resposta satisfatória

Resposta satisfatória

Resposta inicia corretamente, mas não consegue concluir o raciocínio

A residência pode escolher entre:

- Tarifa convencional: R\$ 0,92/kWh
 - Tarifa branca: ponta R\$ 1,30, intermediário R\$ 1,00, fora de ponta R\$ 0,70
- Qual opção é mais vantajosa e por quê?
- Consumo estimado: 20% ponta, 30% intermediário, 50% fora de ponta.

3,373s

6,478s

2min13s

0,95

0,93

0,48

Fonte: própria

Na Tabela 4/Tabela 4, é possível ter uma análise comparativa mais resumida da performance de cada modelo.

Tabela 4 – Comparativo resumido de performance do tempo de execução e performance média

Índices	llama-4-scout-17b-16e	ChatGPT 4.1-mini	llama 3.2 fine-tuned
Tempo médio de execução (s)	5,25627s	14,72206s	75,69231s
Taxa de respostas satisfatórias (%)	88,24%	100%	5,88%
Taxa de alucinações (%)	0%	0%	64,71%
Média de pontuação	0,80	0,93	0,12

Fonte: própria

Vale salientar que as respostas dadas pelo último modelo foram paradas depois de um tempo de alucinações muito extensas e o primeiro resultado foi desconsiderado no cálculo e tratado como uma alucinação

3.10 ANÁLISE FINANCEIRA

Além dos testes de desempenho, é valido salientar o custo das operações das ferramentas. Maioria das ferramentas são gratuitas e *open-source*, porém o *ChatGPT* possui cobrança e o modelo usado no *Groq* permite visualizar o quanto seria cobrado num plano pago, pensando em escalabilidade futura, são dados coerentes para análise. Os custos apresentados na Tabela 5 são referentes aos modelos do *ChatGPT* e uso do *Groq*.

Tabela 5 – Custos financeiros dos modelos

Groq	ChatGPT	
llama-4-scout-17b-16e	chatgpt 4.1-mini	text-embedding-3-small
R\$0,21	R\$0,69	< R\$ 0,01
91.800 tokens	1.689.900 tokens	11.863.000 tokens

Fonte: própria

Os testes realizados são referentes a cerca de 50 requisições por cada modelo. Pensando numa sugestão de modelo de negócio, existem 2 possibilidades de cobrança de uma ferramenta paga com os modelos do ChatGPT e Groq:

- Cobrança 100% por demanda com pequeno adicional de valor sobre o uso, ou seja, se o agente custou R\$1,00 com 100 requisições no mês, cobra-se R\$2,00 e esse valor varia com a demanda, podendo baixar com o tamanho da demanda. Tal modelo considera o objetivo final de ganho financeiro.
- Cobrança fixa por pacote, calcula-se um custo médio por cada valor de requisições mensais (100, 200, 500, etc) e soma-se o valor a ser lucrado no pacote.

Em ambos os casos sugeridos, ainda é considerada a acessibilidade, não visando ganhos substanciais. Particularmente no caso por demanda, é possível cobrar apenas o preço de custo, sem objetivo de lucro final.

Se houver o desejo pelo uso do modelo 100% gratuito, é válido sempre avaliar o poder de processamento que o usuário possui à disposição, pois afeta diretamente o tempo de resposta do modelo. Vale salientar que, pelos resultados apresentados, mesmo que o modelo fine-tunado seja aprimorado, os modelos pagos devem possuir melhor desempenho técnico e de tempo de processamento.

4 CONCLUSÕES E PROPOSTAS DE CONTINUIDADE

4.1 CONCLUSÕES TÉCNICAS

Os modelos de linguagem mostraram desempenhos e respostas bastante diversas entre si, no APÊNDICE A pode-se ver a respostas exatas dadas por cada modelo.

Pode-se concluir que o modelo do *llama 4* possui, uma alta capacidade de responder os *prompts* de maneira satisfatória e com tempo de execução mais baixo. O modelo do *chatgpt 4.1-mini* tem um tempo de execução maior, porém com uma taxa maior de respostas positivas. Finalmente, o modelo mais fraco *llama 3.2 fine-tuned* apresentou grandes dificuldades em trazer respostas satisfatórias num tempo coerente e apresentou alta taxa de alucinações.

A justificativa para tais resultados, no caso positivo, se dá pela alta capacidade de modelos novos e com poder de processamento maior. Nota-se, inclusive uma melhora no tempo de execução quando se tratou de perguntas com elementos mais próximos ao conteúdo do banco de dados.

O modelo *llama 4* apresentou dificuldades em duas perguntas específicas, mostrando-se necessário incluir mais conteúdos no banco de dados referentes a tais assuntos para que o agente seja mais robusto no tocante ao modelo selecionado e a taxa de respostas satisfatórias aumente. Apesar de tais dificuldades, o modelo performou melhor em questões mais matemáticas, trazendo boas respostas num tempo de execução menor.

O modelo *chatgpt 4.1-mini* apresentou uma taxa satisfatória de 100%, porém performou com tempo de execução superior, aparentando dificuldade para responder questões com cálculos matemáticos. O fator mais interessante nesse caso foi a qualidade da escrita das respostas do modelo, pois foi interpretado como apresentando um toque mais próximo do humano nas respostas.

O modelo *llama 3.2 fine-tuned* apresentou uma performance bem abaixo. A capacidade da máquina pode justificar os tempos de execução maiores, porém a alta taxa de alucinações pode se dar por dois fatores: o processo de *fine-tuning* foi insuficiente, pois o modelo deveria necessitar de mais exemplos de dados para

funcionar melhor; e houve uma particularidade no uso desse modelo, ele não permite o uso de ferramentas, justamente por ser um modelo menos capaz, assim, ele não teve acesso às ferramentas de calculadora e busca de dados no banco.

No geral, o agente é capaz de responder perguntas específicas com cada modelo selecionado e possui áreas de melhoria. O limitador da máquina usada nos testes foi considerável, limitando a escolha de um modelo mais potente para o modelo local do *Ollama*.

4.2 CONCLUSÕES FINANCEIRAS

Considerando que o agente desenvolvido é um protótipo, o custo de testes foi considerado bem baixo, não ultrapassando R\$1,00, o que permite concluir que uma solução de maior porte tem potencial de desenvolvimento sem um custo muito maior, mantendo um custo baixo do usuário e com potencial de lucro. O treinamento de um modelo local não é custoso financeiramente, sendo os únicos agravantes o tempo de treinamento e capacidade de processamento da máquina disponível.

4.3 LIMITAÇÕES DO PROJETO

Houve três pontos de limitações do projeto que justificam alguns resultados de performance dos modelos, listados por ordem de dificuldade do menor para o maior:

1. Base de dados do RAG restrita a temas específicos: Nota-se uma diferença em tempo e qualidade das respostas de assuntos abordados por conteúdo dentro do banco de dados e assuntos não abordados, sendo uma solução o aumento do conteúdo de pesquisa do agente, incluindo mais temas pertinentes;
2. Modelo fine-tunado com baixa performance: O processo de fine-tuning se mostrou um desafio. O processo em si não é complexo, porém apresentou desafios por necessitar de um tempo de refino muito extenso;
3. Limitações de hardware: Máquinas melhores com GPUs dedicadas e voltadas para performance de modelos de linguagem permitiriam o treinamento de

modelos mais potentes e até mesmo multimodais. Entretanto, o custo é inviável para a escala atual do projeto.

4.4 PROPOSTAS DE CONTINUIDADE

Um agente de IA *RAG* tem sempre potencial de melhoria do seu banco de dados acessório, pois as informações se atualizam e cada vez mais dados podem ser adicionados ao banco, enriquecendo as fontes de pesquisa do agente, possibilitando respostas mais coerentes e uma ação mais rápida dos modelos.

Um modelo treinado localmente também apresenta potencial de melhoria, pois também pode sempre ser mais bem treinado, sendo apresentado mais dados, tornando-o mais especializado na sua tarefa. Vale salientar que é possível a perda de capacidades gerais ao treinar um modelo muito específico.

Visando escalabilidade do produto, faz sentido buscar uma solução personalizada de agente, porém tal situação deve incidir também um aumento dos custos do produto. Pensando como uma ferramenta de pesquisa e visando o baixo custo do usuário, manter o uso da versão de comunidade do *n8n* segue sendo uma solução coerente.

Um ponto interessante de melhoria se dá em aumentar a gama de ferramentas disponíveis ao agente, possibilitando também diminuir processamento com ganho de performance, já que o modelo não deverá se preocupar com algumas questões e irá focar mais em responder a requisição.

Uma possibilidade de extensão das funcionalidades do agente é propor pessoas diferentes, especializadas em áreas diferentes da engenharia e possuindo outros materiais de pesquisa no banco de dados, estendendo sua aplicação para outras áreas como mecânica, eletrônica e civil.

Por fim, independente do uso final, seja pesquisa ou desenvolvimento de produto, é coerente considerar o desenvolvimento de uma interface mais amigável ao usuário não técnico, pois também torna o agente mais acessível no quesito usabilidade.

REFERÊNCIAS

- ABNT. (2004). NBR 5410:2004. *Instalações elétricas de baixa tensão*.
- Academy, D. S. (2023). *DEEP LEARNING BOOK: Capítulo 14: Algoritmo Back-Propagation*.
Fonte: DEEP LEARNING BOOK: https://www.deeplearningbook.com.br/algoritmo-backpropagation-parte1-grafos-computacionais-e-chain-rule/?utm_source=chatgpt.com
- ANEEL. (09 de 09 de 2010). Resolução Normativa nº414.
- Anisuzzaman. (2025). Fine-Tuning Large Language Models for Specialized Use Cases. *Mayo Clinic Proceedings: Digital Health*.
- Braga, N. C. (s.d.). <https://newtoncbraga.com.br/usando-os-instrumentos/3103-ins144.html>.
Fonte: Instituto NCB.
- Breiman, L. (2001). Random Forests. *Machine Learning*.
- CanalEnergia. (2020). <https://www.canalenergia.com.br/noticias/53155142/bonus-motor-saida-ideal-para-a-eficiencia-energetica-nas-empresas>. Fonte: CanlEnergia.
- Data Science Academy. (2025). Fonte: deeplearningbook:
<https://www.deeplearningbook.com.br/algoritmo-de-agente-baseado-em-ia-com-reinforcement-learning-q-learning/>
- FENEK, O. (2025). <https://python.plainenglish.io/fine-tune-langs-a-comprehensive-guide-between-full-partial-fine-tuning-an-end-to-end-python-3fa7223f5519>. Fonte: Medium.
- IBM. (s.d.). *O que é machine learning*. Acesso em 23 de Março de 2024, disponível em ibm.com: <https://www.ibm.com/br-pt/topics/machine-learning>
- INMETRO. (2021). *Conheça o programa brasileiro de etiquetagem*. Acesso em 25 de março de 2024, disponível em gov: [https://www.gov.br/inmetro/pt-br/assuntos/avaliacao-da-conformidade/programa-brasileiro-de-etiquetagem/conheca-o-programa#:~:text=O%20Programa%20Brasileiro%20de%20Etiquetagem%20\(PBE\)%20coordenado%20pelo%20Inmetro,decis%C3%B5es%20de%20compra%20mais%20co](https://www.gov.br/inmetro/pt-br/assuntos/avaliacao-da-conformidade/programa-brasileiro-de-etiquetagem/conheca-o-programa#:~:text=O%20Programa%20Brasileiro%20de%20Etiquetagem%20(PBE)%20coordenado%20pelo%20Inmetro,decis%C3%B5es%20de%20compra%20mais%20co)
- Lake, B. M. (2016). Fonte: arxiv: <https://arxiv.org/abs/1604.00289>
- Litro de Luz. (2022). <https://www.litrodeluz.com/post/selo-procel-o-que-%C3%A9-para-que-serve>. Fonte: Litro de Luz.
- MACHADO, R. d. (2023). APLICAÇÃO DE REDES NEURAIS ARTIFICIAIS PARA PARAMETRIZAÇÃO DA CONDUTIVIDADE HIDRÁULICA EM SOLOS NÃO SATURADOS. UFRRJ.
- MARTENS, A. d., & SILVA. (s.d.). N. da. Atribuição de responsabilidades por conteúdo harmônico em redes de distribuição utilizando-se das definições de fator de consumo e fator de geração de harmônicos. *Eletrônica de Potência*.
- MARTINS, E. D. (2020). *Teorias da Aprendizagem*. Acesso em 23 de Março de 2024, disponível em <https://ppp.esp.ce.gov.br/pressupostos-teorico-metodologicos/teorias-da-aprendizagem/>
- Matos, D. (2023). Fonte: Ciência e Dados: <https://www.cienciaedados.com/normalizacao-em-machine-learning/>
- MetaQuotes. (2023). <https://www.mql5.com/pt/articles/12627>. Fonte: MQL5.

- Monteiro, S. T. (2004). Desempenho de algoritmos de aprendizagem por reforço sob condições de ambiguidade sensorial em robótica móvel.
- Nascimento, R. F. (2009). O algoritmo Support Vector Machines (SVM): avaliação da separação ótima de classes.
- Nielsen, M. (2019). <https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applicationsundefined>. Fonte: Neural Networks and Deep Learning.
- Oliveira, A. F. (2018). Favorecendo o Desempenho do k-Means via.
- Otten, N. V. (2023). <https://spotintelligence.com/2023/11/20/gpt-prompt-engineering/>. Fonte: Spot Intelligence.
- PETUKHOVA, E., & al. e. (2024). Large language models (LLMs): survey, technical frameworks, and future challenges. *Artificial Intelligence Review*.
- RDD10+. (s.d.). *Agentes de IA: Evolução, Componentes e Aplicações*. Fonte: RDD10+: <https://www.robertodiasduarte.com.br/agentes-de-ia-evolucao-componentes-e-aplicacoes/>
- SAMPAIO, F. P. (2023). EFICIÊNCIA ENERGÉTICA EM ILUMINAÇÃO PÚBLICA: ESTUDO DE CASO DO BAIRRO JARDIM SANTA FÉ. *Revista Científica Semana Acadêmica. Fortaleza*.
- Santo, R. d. (2012). Utilização da Análise de Componentes Principais.
- Santos, V. V. (2023). INTELIGÊNCIA ARTIFICIAL E APRENDIZADO DE MÁQUINA: UMA VISÃO GERAL.
- Savage T, P. M. (2025). Fine-Tuning Methods for Large Language Models in Clinical Applications. *JMIR (Journal of Medical Internet Research)*.
- SHABANI, N. (2024). A Historical Survey of Advances in Transformer Architectures. *Universidade de Uppsala*.
- Shaibu, S. (2024). Fonte: Datacamp: <https://www.datacamp.com/tutorial/normalization-vs-standardization>
- Sharma, P. (2025). Fonte: Octal IT solution LLP: <https://www.octalsoftware.com/blog/types-of-l1m>
- Shewan, D. (2023). <https://www.wordstream.com/blog/ws/2017/07/28/machine-learning-applicationsundefined>. Fonte: WordStream.
- SILVA, A. L., IBAROLA, D., NÓBREGA, M. J., & MARTINS, F. B. (s.d.). Qualidade e eficiência energética aplicada a consumidor do grupo A. *Revista Tecnológica da Universidade Santa Úrsula*.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., N. Gomez, A., . . . Polosukhin, I. (2017). Attention Is All You Need. *Advances in neural information processing systems*.
- WANG, X., WEI, J., SCHUURMANS, D., LE, Q. V., CHI, E. H., NARANG, S., . . . ZHOU, D. (2022). *Self-consistency improves chain of thought reasoning in language models*. Fonte: arXiv: <https://arxiv.org/abs/2203.11171>
- WEI, K. (2023). <https://python.plainenglish.io/fine-tune-l1ms-a-comprehensive-guide-between-full-partial-fine-tuning-an-end-to-end-python-3fa7223f5519>. Fonte: Writer's Room.

- WHITE, J., FU, Q., HAYS, S., SANDBORN, M., OLEA, C., GILBERT, H., . . . SCHMIDT, D. C. (2023). A Prompt Pattern Catalog to Enhance Prompt Engineering with ChatGPT. *arXiv*.
- YAO, S., ZHAO, J., YU, D., DU, N., SHAFRAN, I., NARASIMHAN, K., & CAO, Y. (2022). ReAct: Synergizing Reasoning and Acting in Language Models. *arXiv*.
- ZHAO, W. X., ZHOU, K., LI, J., TANG, T., WANG, X., HOU, Y., . . . LIU, Z. (2023). *A Survey of Large Language Models*. Fonte: arXiv: <https://arxiv.org/abs/2303.18223>

APÊNDICE A – EXEMPLO DE COMPARAÇÃO DE RESULTADOS DAS PERGUNTAS AOS MODELOS

1. Groq (llama-4-scout-17b-16e)

- Qual é o potencial de redução de consumo se eu automatizar a iluminação com sensores de presença e dimerização?

A automação da iluminação com sensores de presença e dimerização pode ter um potencial significativo de redução de consumo de energia. Aqui estão algumas estimativas:

Sensores de Presença:

- **Redução de consumo:** 20-40% - Os sensores de presença podem reduzir o consumo de energia ao desligar as luzes quando não há pessoas presentes em uma área.
- **Economia de energia:** 10-20 kWh/m²/ano - Isso pode variar dependendo do tipo de iluminação, do uso do espaço e da eficiência dos sensores.

Dimerização:

- **Redução de consumo:** 10-30% - A dimerização pode reduzir o consumo de energia ao ajustar o nível de iluminação às necessidades específicas de uma área.
- **Economia de energia:** 5-15 kWh/m²/ano - Isso pode variar dependendo do tipo de iluminação, do uso do espaço e da eficiência da dimerização.

Combinação de Sensores de Presença e Dimerização:

- **Redução de consumo:** 30-50% - A combinação de sensores de presença e dimerização pode ter um efeito sinérgico, levando a uma redução ainda maior no consumo de energia.
- **Economia de energia:** 15-30 kWh/m²/ano - Isso pode variar dependendo do tipo de iluminação, do uso do espaço e da eficiência dos sensores e da dimerização.

É importante notar que essas são estimativas gerais e que o potencial de redução de consumo pode variar dependendo de muitos fatores, incluindo:

- Tipo de iluminação (incandescente, fluorescente, LED, etc.)
- Uso do espaço (residencial, comercial, industrial, etc.)
- Eficiência dos sensores e da dimerização

- Configuração do sistema de iluminação

No entanto, em geral, a automação da iluminação com sensores de presença e dimerização pode ser uma estratégia eficaz para reduzir o consumo de energia e melhorar a eficiência energética.

Tempo: 3,276s

2. ChatGPT 4.1-mini

- **Qual é o potencial de redução de consumo se eu automatizar a iluminação com sensores de presença e dimerização?**

A automação da iluminação com sensores de presença e dimerização pode reduzir o consumo de energia da iluminação entre 20% a 60%, dependendo do perfil de uso e da qualidade da automação. Essa tecnologia evita que as luzes fiquem acesas desnecessariamente e ajusta o nível de luz conforme a necessidade, resultando em economia significativa. Além disso, contribui para maior vida útil das luminárias, redução de custos de manutenção e melhor conforto para os usuários.

Tempo: 17,451s

3. Ollama (llama-3.2 fine tune)

- **Qual é o potencial de redução de consumo se eu automatizar a iluminação com sensores de presença e dimerização?**

Um sistema de iluminação inteligente que automatiza a iluminação em relação à presença de pessoas e regula a intensidade da luz pode oferecer várias vantagens. Aqui estão algumas estimativas do potencial de redução de consumo:

- Desligamento automático: 10% - 20% de redução no consumo de energia elétrica
- Dimerização (redução da intensidade da luz): 5% - 15% de redução no consumo de energia elétrica
- Combinado com sensores de presença: 20% - 30% de redução no consumo de energia elétrica

É importante notar que essas estimativas são aproximadas e dependem do uso real da iluminação, das condições de iluminação desejada e do tipo de equipamento utilizados.

No entanto, é fácil ver como um sistema de iluminação inteligente pode ter um impacto positivo significativo no consumo de energia elétrica. Além disso, com o uso crescente de tecnologias de iluminação LED, que são mais eficientes em termos de energia, um sistema de automação e dimerização pode oferecer uma melhoria significativa na eficiência energética da iluminação.

Tempo: 59,483s

Todas as respostas podem ser encontradas no link:

<https://github.com/jricardocv/TCC-Jos-Ricardo-Vianna>