

INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DO TOCANTINS
CAMPUS PALMAS
CURSO SUPERIOR DE ENGENHARIA ELÉTRICA

DIEGO ARIEL BARBOSA SANTOS
MARCOS VINICIUS BORGES LOPES MESSIAS

**SISTEMA INTELIGENTE DE CONSULTA À NBR 5410: DESENVOLVIMENTO DE
CHATBOT PARA SUPORTE EM PROJETOS DE INSTALAÇÕES ELÉTRICAS**

PALMAS
2026

DIEGO ARIEL BARBOSA SANTOS
MARCOS VINICIUS BORGES LOPES MESSIAS

**SISTEMA INTELIGENTE DE CONSULTA À NBR 5410: DESENVOLVIMENTO DE
CHATBOT PARA SUPORTE EM PROJETOS DE INSTALAÇÕES ELÉTRICAS**

Trabalho de Conclusão de Curso apresentado à Coordenação do Curso de Bacharelado em Engenharia Elétrica da Unidade Campus Palmas, do Instituto Federal do Tocantins, como exigência à obtenção do título de Bacharel em Engenharia Elétrica.

Orientador: Prof. Me. Celivan Ferreira Vieira

PALMAS
2026

Dados Internacionais de Catalogação na Publicação (CIP)
Bibliotecas do Instituto Federal do Tocantins

S237s Santos, Diego Ariel Barbosa
SISTEMA INTELIGENTE DE CONSULTA À NBR 5410:
DESENVOLVIMENTO DE CHATBOT PARA SUPORTE EM
PROJETOS DE INSTALAÇÕES ELÉTRICAS / Diego Ariel Barbosa
Santos, Marcos Vinicius Borges Lopes Messias. – Palmas, TO, 2026.
65 p. : il. color.

Trabalho de Conclusão de Curso (Bacharelado em Engenharia
Elétrica) – Instituto Federal de Educação, Ciência e Tecnologia do
Tocantins, Campus Palmas, Palmas, TO, 2026.

Orientador: Me. Celivan Ferreira Vieira

1. Chatbot especializado. 2. NBR 5410. 3. Inteligência artificial. I.
Messias, Marcos Vinicius Borges Lopes. II. Vieira, Celivan Ferreira III.
, . IV. Título.

CDD 621

A reprodução total ou parcial, de qualquer forma ou por qualquer meio, deste documento é autorizada para fins
de estudo e pesquisa, desde que citada a fonte.

Elaborado pelo sistema de geração automática de ficha catalográfica do IFTO com os dados fornecidos
pelo(a) autor(a).

DIEGO ARIEL BARBOSA SANTOS
MARCOS VINICIUS BORGES LOPES MESSIAS

**SISTEMA INTELIGENTE DE CONSULTA À NBR 5410: DESENVOLVIMENTO DE
CHATBOT PARA SUPORTE EM PROJETOS DE INSTALAÇÕES ELÉTRICAS**

Trabalho de Conclusão de Curso apresentado
à Coordenação do Curso de Bacharelado em
Engenharia Elétrica da Unidade Campus Pal-
mas, do Instituto Federal do Tocantins, como
exigência à obtenção do título de Bacharel em
Engenharia Elétrica.

Aprovado em: 26/06/2026.

BANCA AVALIADORA



Documento assinado digitalmente
CELIVAN FERREIRA VIEIRA
Data: 15/07/2026 10:25:43-0300
Verifique em <https://validar.iti.gov.br>

Prof. Me. Celivan Ferreira Vieira (Orientador)
IFTO - Campus Palmas



Documento assinado digitalmente
WELLINGTON DIAS DA SILVA
Data: 15/07/2026 09:59:28-0300
Verifique em <https://validar.iti.gov.br>

Prof. Me. Wellington Dias da Silva
IFTO - Campus Palmas



Documento assinado digitalmente
HUMBERTO RODRIGUES MACEDO
Data: 15/07/2026 10:35:33-0300
Verifique em <https://validar.iti.gov.br>

Prof. Dr. Humberto Rodrigues Macedo
IFTO - Campus Palmas

RESUMO

A elaboração de projetos voltados para sistemas elétricos de baixa tensão exige consultas contínuas e rigorosas à NBR 5410, regulamentação técnica que define critérios essenciais para assegurar proteção operacional e adequação normativa. As metodologias convencionais de acesso a este documento normativo apresentam restrições significativas relacionadas ao período despendido na localização de informações específicas e à compreensão apropriada de exigências técnicas complexas. Este trabalho apresenta o desenvolvimento de um sistema inteligente para consulta à NBR 5410, implementado mediante um agente conversacional especializado que utiliza metodologias de processamento de linguagem natural para otimizar o acesso às informações regulamentares. O sistema foi construído por meio de uma estratégia híbrida que integra algoritmos proprietários de extração de entidades técnicas com tecnologias contemporâneas de inteligência artificial para processamento de consultas em linguagem coloquial. A estrutura de conhecimento foi organizada a partir do conteúdo integral da NBR 5410, sistematizando seções, tabelas especializadas, metodologias e especificações técnicas de modo a maximizar a recuperação de dados relevantes. O *chatbot* proposto demonstrou capacidade de interpretar questionamentos técnicos formulados em linguagem natural, reconhecer parâmetros específicos como potência elétrica, diferenças de potencial e categoria de instalação, fornecendo respostas contextualizadas que incorporam referências precisas às seções pertinentes da regulamentação. Os resultados indicam que o sistema proporciona acesso otimizado às informações da NBR 5410, diminuindo consideravelmente o período necessário para identificar requisitos técnicos específicos e disponibilizando apoio técnico especializado para engenheiros, técnicos eletricitas e estudantes durante a consulta e a elaboração de projetos de sistemas elétricos.

Palavras-chave: *Chatbot* especializado. NBR 5410. Apoio técnico. Inteligência artificial. Processamento de linguagem natural.

ABSTRACT

The development of projects aimed at low-voltage electrical systems requires continuous and rigorous consultation of NBR 5410, a technical regulation that defines essential criteria to ensure operational protection and normative compliance. Conventional methodologies for accessing this normative document present significant restrictions related to the time spent locating specific information and the appropriate understanding of complex technical requirements. This work presents the development of an intelligent system for consulting NBR 5410, implemented through a specialized conversational agent using natural language processing methodologies to optimize access to regulatory information. The system was built through a hybrid strategy that integrates proprietary algorithms for technical entity extraction with contemporary artificial intelligence technologies for processing queries in colloquial language. The knowledge structure was organized from the complete content of NBR 5410, systematizing sections, specialized tables, methodologies and technical specifications in order to maximize the retrieval of relevant data. The proposed chatbot demonstrated the ability to interpret technical inquiries formulated in natural language, recognize specific parameters such as electrical power, potential differences and installation category, providing contextualized responses that incorporate precise references to the pertinent sections of the regulation. The results indicate that the system provides optimized access to NBR 5410 information, considerably reducing the time necessary to identify specific technical requirements and providing specialized technical support for engineers, electrical technicians and students during the consultation and development of electrical system projects.

Keywords: Specialized chatbot. NBR 5410. Technical support. Artificial intelligence. Natural language processing.

LISTA DE ILUSTRAÇÕES

Figura 1 – Arquitetura do sistema de <i>chatbot</i> especializado	14
Figura 2 – Fluxo de processamento de <i>chatbot</i> baseado em regras	15
Figura 3 – Fluxo de processamento de <i>chatbot</i> baseado em IA	16
Figura 4 – Fluxo de processamento de <i>chatbot</i> híbrido	17
Figura 5 – Arquitetura RAG do sistema desenvolvido	28
Figura 6 – Interface do Assistente NBR 5410	31
Figura 7 – Interface do Assistente NBR 5410 — visão geral	32
Figura 8 – Consulta sobre cabos para tomadas de uso geral	32
Figura 9 – Consulta sobre seção mínima para circuito de tomadas de 20 A	33
Figura 10 – Consulta sobre condutor para ar-condicionado de 12.000 BTU/h	34
Figura 11 – Continuação: cálculo de corrente e seleção do condutor	35
Figura 12 – Consulta sobre bitola de cabo para circuito de 150 m e 15 A	36
Figura 13 – Resultado: seleção do condutor e alertas normativos	36
Figura 14 – Continuação: parâmetros assumidos e alertas finais	37
Figura 15 – Recusa do sistema a consulta fora do escopo técnico	37

LISTA DE TABELAS

Tabela 1	– Características dos tipos de <i>chatbots</i>	18
Tabela 2	– Componentes da base de conhecimento do sistema	20
Tabela 3	– Capacidade de condução de corrente — condutores de cobre, isolação PVC 70°C, método B1, 3 condutores carregados, 30°C	22

LISTA DE ABREVIATURAS E SIGLAS

ABNT	Associação Brasileira de Normas Técnicas
API	<i>Application Programming Interface</i> (Interface de Programação de Aplicações)
BERT	<i>Bidirectional Encoder Representations from Transformers</i>
BTU	<i>British Thermal Unit</i>
DR	Dispositivo Residual de Corrente
EER	<i>Energy Efficiency Ratio</i> (Coeficiente de Eficiência Energética)
GPT	<i>Generative Pre-trained Transformer</i>
IA	Inteligência Artificial
IFTO	Instituto Federal de Educação, Ciência e Tecnologia do Tocantins
IR	<i>Information Retrieval</i> (Recuperação de Informação)
LLM	<i>Large Language Model</i> (Modelo de Linguagem de Grande Escala)
NBR	Norma Brasileira
NER	<i>Named Entity Recognition</i> (Reconhecimento de Entidades Nomeadas)
PDF	<i>Portable Document Format</i>
PLN	Processamento de Linguagem Natural
RAG	<i>Retrieval-Augmented Generation</i> (Geração Aumentada por Recuperação)
WSD	<i>Word Sense Disambiguation</i> (Resolução de Ambiguidade Lexical)

LISTA DE SÍMBOLOS

I	Corrente de projeto do circuito (A)
P	Potência ativa da carga (W)
V	Tensão nominal (V)
f_p	Fator de potência da carga ($\cos \varphi$)
ΔV	Queda de tensão no condutor (V)
$\Delta V\%$	Queda de tensão percentual (%)
ρ	Resistividade elétrica do cobre ($\Omega \cdot \text{mm}^2/\text{m}$)
L	Comprimento do circuito (m)
S	Seção nominal do condutor (mm^2)
I_B	Corrente de projeto (de uso) do circuito (A)
I_n	Corrente nominal do dispositivo de proteção (A)
I_z	Capacidade de condução de corrente do condutor (A)
f_{temp}	Fator de correção por temperatura
f_{agrup}	Fator de correção por agrupamento de circuitos

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Problema de pesquisa	11
1.2	Justificativa	12
1.3	Objetivos	12
1.3.1	Objetivo Geral	12
1.3.2	Objetivos Específicos	12
2	REVISÃO DE LITERATURA	14
2.1	Chatbots aplicados à Engenharia	14
2.1.1	Chatbots Baseados em Regras	15
2.1.2	Chatbots Baseados em IA	16
2.1.3	Chatbots Híbridos	17
2.2	NBR 5410 e Literatura Técnica Especializada	18
2.2.1	Base de Conhecimento Técnico	18
2.2.2	Estrutura e Organização da NBR 5410	19
2.2.3	Delimitação da Base de Conhecimento do Sistema	19
2.2.4	Equações Fundamentais de Dimensionamento Elétrico	20
2.3	Inteligência Artificial Aplicada à Engenharia	23
2.3.1	Histórico da IA na Engenharia	23
2.3.2	Tecnologias de Processamento de Linguagem Natural	23
2.3.3	Conceitos Fundamentais do PLN	24
2.3.4	Sistemas de Recuperação de Informação	25
3	PROCEDIMENTOS METODOLÓGICOS	27
3.1	Pesquisa Bibliográfica	27
3.2	Desenvolvimento do Sistema	27
3.2.1	Visão Geral da Arquitetura	27
3.2.2	Construção da Base de Conhecimento	27
3.2.3	O Agente Inteligente e as Ferramentas de Cálculo	28
3.2.4	Interface de Usuário	29
3.2.5	Hospedagem e Código-fonte	29
3.3	Avaliação do Sistema	30
4	RESULTADOS E DISCUSSÃO	31
4.1	Interface do Sistema	31
4.2	Consultas Técnicas Realizadas	31
4.2.1	Cabos para Tomadas de Uso Geral	32

4.2.2	Seção Mínima para Circuito de Tomadas de 20 A	32
4.2.3	Condutor para Ar-condicionado de 12.000 BTU/h	33
4.2.4	Bitola de Cabo para Circuito de 150 m com Corrente de 15 A	34
4.3	Controle do Escopo da Ferramenta	35
4.4	Considerações Gerais	36
5	CONSIDERAÇÕES FINAIS	38
5.1	Trabalhos Futuros	38
	REFERÊNCIAS	40
	APÊNDICE A – CÓDIGO-FONTE DO AGENTE RAG	43
	APÊNDICE B – CÓDIGO-FONTE DO MÓDULO DE INGESTÃO	54
	APÊNDICE C – BASE DE CONHECIMENTO SUPLEMENTAR	58

1 INTRODUÇÃO

No cenário atual da engenharia elétrica brasileira, cresce a demanda por ferramentas que agilizem o acesso às informações normativas durante a elaboração de projetos. A norma NBR 5410, referência para instalações elétricas de baixa tensão, representa um documento extenso e dividido em múltiplas seções, desde conceitos básicos até exigências técnicas detalhadas.

Durante o desenvolvimento de projetos elétricos, questões como o dimensionamento de condutores, especificação de dispositivos de proteção e a aplicação de fatores de ajuste surgem constantemente, exigindo respostas rápidas e fundamentadas na norma. A demora na obtenção dessas respostas pode impactar prazos e comprometer a qualidade dos projetos.

Diante desse contexto, soluções baseadas em inteligência artificial, especialmente sistemas apoiados por processamento de linguagem natural, destacam-se como alternativas eficientes para otimizar o acesso à informação técnica. Tais tecnologias permitem que profissionais realizem consultas em linguagem comum, recebendo respostas contextualizadas e precisas.

Este trabalho propõe o desenvolvimento de um sistema inteligente para consulta à NBR 5410, implementado por meio de um *chatbot* especializado que integra o conteúdo normativo de forma estruturada. O objetivo é fornecer suporte técnico imediato a engenheiros, técnicos, estudantes e demais profissionais que atuam com projetos e instalações elétricas, aliando precisão normativa e facilidade de acesso por meio de interface conversacional eficiente.

1.1 Problema de pesquisa

Engenheiros, técnicos eletricitas, estudantes de engenharia elétrica e demais profissionais que atuam com instalações elétricas estão sujeitos a dificuldades significativas no acesso eficiente às informações específicas da NBR 5410, resultando em perda de produtividade e possíveis inconsistências na aplicação de requisitos normativos.

Diante deste cenário, surge a questão: como um sistema inteligente de consulta baseado em *chatbot* pode otimizar o acesso às informações da NBR 5410, fornecendo apoio técnico eficiente e preciso a engenheiros, técnicos e estudantes durante o desenvolvimento e a consulta de projetos de instalações elétricas de baixa tensão?

1.2 Justificativa

Este trabalho se justifica pela necessidade de otimizar o acesso às informações da NBR 5410, norma fundamental que regulamenta as instalações elétricas de baixa tensão no Brasil. A NBR 5410 constitui um documento técnico extenso e complexo, com múltiplas seções interdependentes que requerem consulta frequente durante projetos elétricos (ABNT, 2004).

A consulta tradicional apresenta limitações operacionais significativas que impactam a produtividade de engenheiros, técnicos e estudantes que precisam consultar a norma com frequência. O processo manual de localização de informações específicas em suas 217 páginas consome tempo considerável, especialmente para requisitos pontuais. Para estudantes e técnicos em formação, a dificuldade é ainda maior, pois a interpretação correta dos requisitos normativos exige familiaridade com uma estrutura extensa e altamente técnica.

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver um sistema inteligente de consulta à NBR 5410 por meio de um *chatbot* especializado que utilize processamento de linguagem natural para fornecer apoio técnico eficiente a engenheiros, técnicos eletricitas, estudantes e demais profissionais envolvidos com projetos e instalações elétricas de baixa tensão, democratizando o acesso às informações normativas de forma rápida, contextualizada e acessível.

1.3.2 Objetivos Específicos

- construir uma base de conhecimento a partir do conteúdo da ABNT NBR 5410, processando o documento oficial em formato PDF e elaborando material suplementar com tabelas de condutores e fórmulas de dimensionamento, de forma a viabilizar a recuperação semântica das informações normativas pelo sistema;
- desenvolver um sistema de busca inteligente capaz de localizar e apresentar informações relevantes da NBR 5410 com base em consultas técnicas dos usuários, incluindo referências precisas às seções normativas aplicáveis;
- desenvolver e publicar uma interface *web* acessível via navegador, utilizando o *framework* Streamlit, que permita a engenheiros, técnicos e estudantes formular consultas em linguagem natural e receber respostas contextualizadas com base nos requisitos da NBR 5410, sem necessidade de instalação de *software*;

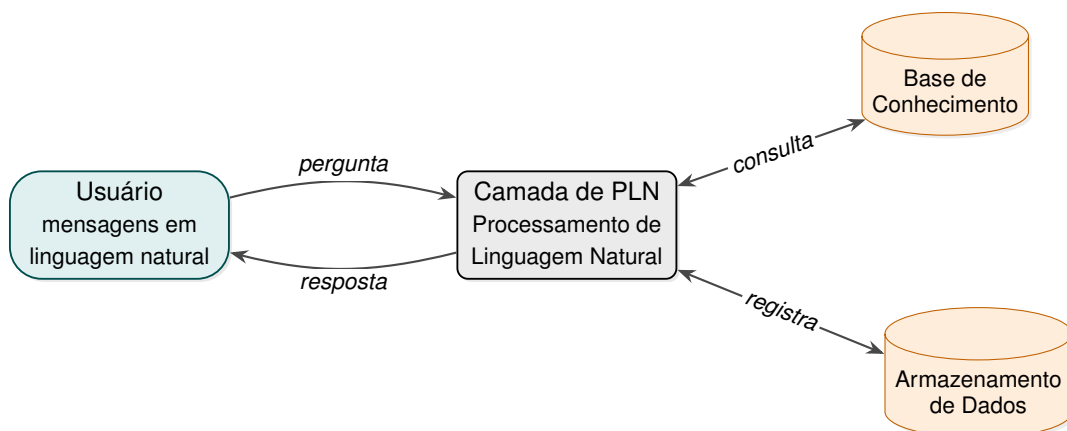
- validar o sistema por meio de testes com consultas técnicas representativas, avaliando a precisão das respostas e a adequação das referências normativas;

2 REVISÃO DE LITERATURA

2.1 Chatbots aplicados à Engenharia

Um *chatbot* pode ser definido, de forma simples, como um programa de computador capaz de simular uma conversa humana, respondendo perguntas e executando tarefas a partir de entradas em linguagem natural. Quando esse tipo de sistema é treinado ou ajustado para atuar dentro de uma área específica de conhecimento, ele passa a ser chamado de *chatbot* especializado. Diferentemente dos assistentes generalistas, que respondem sobre praticamente qualquer assunto, esses sistemas concentram-se em um domínio particular, o que lhes permite compreender melhor a terminologia própria daquela área e oferecer respostas mais precisas e contextualizadas (Chatcompose, 2023; Easycoder, 2024). De modo geral, esses sistemas organizam-se em torno de uma camada de processamento de linguagem natural que interpreta a pergunta do usuário, consulta uma base de conhecimento e registra as interações, conforme ilustra a Figura 1.

Figura 1 – Arquitetura do sistema de *chatbot* especializado



Fonte: Adaptado pelos autores com base em Selzy (2024).

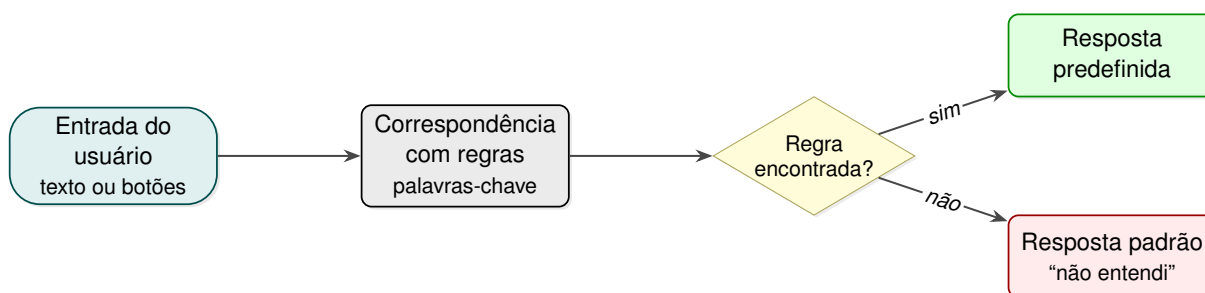
A literatura técnica indica que a adoção de *chatbots* em contextos de engenharia vem se mostrando promissora, sobretudo como apoio a tarefas repetitivas e à automação de consultas (Jack e Yan, 2024). Mirto (2023) destaca que *chatbots* baseados em inteligência artificial conseguem auxiliar profissionais a depurar *scripts*, apoiar etapas do projeto técnico e reduzir o tempo gasto em tarefas operacionais repetitivas, o que tende a aumentar a produtividade da equipe. Para a Easycoder (2024), esses sistemas especializados por domínio podem atuar como assistentes técnicos, consultores normativos ou ferramentas de apoio à tomada de decisão — papel que se aproxima bastante do que se pretende com o sistema desenvolvido neste trabalho. Em linhas gerais, o sucesso desse tipo de ferramenta depende diretamente da qualidade da base de conhecimento utilizada e da capacidade do sistema de interpretar corretamente a intenção por trás da pergunta do

usuário (Chatcompose, 2023).

2.1.1 Chatbots Baseados em Regras

Os *chatbots* baseados em regras são, historicamente, a forma mais simples de automatizar uma conversa. Eles funcionam a partir de um conjunto predefinido de padrões e respostas, seguindo uma lógica do tipo “se isso, então aquilo”: cada entrada do usuário é comparada a um conjunto de regras programadas, e a resposta correspondente é simplesmente recuperada (Robylon.AI, 2026). A Figura 2 representa esse fluxo de processamento, no qual a entrada do usuário é comparada às regras predefinidas até que uma correspondência seja encontrada e a resposta associada seja retornada. Por não dependerem de aprendizado ou de interpretação semântica, esses sistemas são especialmente indicados para domínios bem delimitados, em que as perguntas tendem a seguir estruturas recorrentes. Em engenharia elétrica, por exemplo, é comum que as consultas técnicas sigam padrões parecidos, como “qual a seção mínima para...”, “como calcular...” ou “quais os requisitos para...”.

Figura 2 – Fluxo de processamento de *chatbot* baseado em regras



Fonte: Adaptado pelos autores com base em Robylon.AI (2026).

Comparando as duas abordagens, a Robylon.AI (2026, [s. p.]) resume bem a diferença entre elas ao afirmar que um *chatbot* baseado em regras “segue *scripts* predefinidos, enquanto um *chatbot* de inteligência artificial usa processamento de linguagem natural e aprendizado de máquina para entender e responder dinamicamente”¹. Essa característica explica tanto a principal vantagem dos sistemas baseados em regras — a previsibilidade e o controle total sobre o conteúdo entregue — quanto sua maior limitação: a dificuldade de lidar com perguntas fora do escopo previsto pelo desenvolvedor (IBM, 2025).

Justamente por não compreender o contexto da linguagem, esse tipo de *chatbot* não aprende nem se adapta a situações novas: ele apenas reconhece palavras-chave ou opções de botões e devolve uma resposta já programada (Robylon.AI, 2026). Ainda assim, essa limitação não invalida sua utilidade. No contexto de instalações elétricas regidas pela NBR

¹ Tradução nossa. No original: “a rule-based chatbot follows predefined scripts, while an AI chatbot uses NLP and machine learning to understand and respond dynamically”.

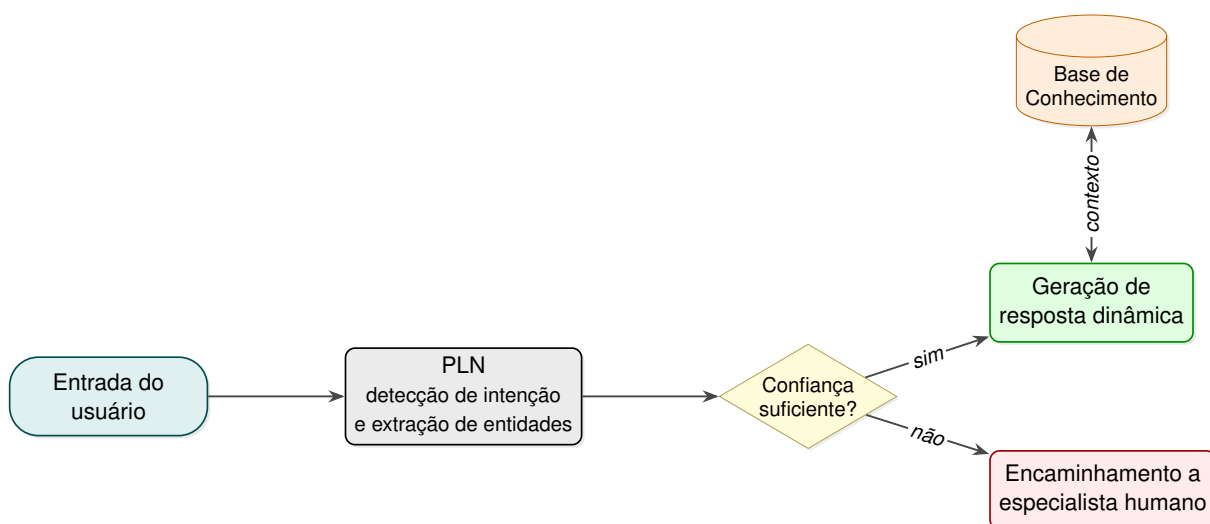
5410, esse modelo pode ser perfeitamente aplicado a demandas simples e recorrentes, como:

- responder perguntas frequentes sobre requisitos básicos da NBR 5410;
- apresentar tabelas fixas de dimensionamento de condutores para situações padronizadas;
- guiar o usuário por fluxos de verificação de conformidade com perguntas de sim ou não.

2.1.2 Chatbots Baseados em IA

Diferentemente dos sistemas baseados em regras, os *chatbots* construídos sobre inteligência artificial recorrem a técnicas de aprendizado de máquina e de processamento de linguagem natural para interpretar a pergunta do usuário, e não apenas reconhecê-la (Russell e Norvig, 2020). Essa diferença é o que permite a esses sistemas compreenderem variações de escrita, sinônimos e o contexto da conversa, produzindo respostas que se ajustam à situação específica de quem está perguntando (Jurafsky e Martin, 2021). A Figura 3 sintetiza esse fluxo, em que a etapa de processamento de linguagem natural avalia a confiança na interpretação da pergunta antes de gerar a resposta dinâmica ou encaminhar o atendimento a um especialista humano.

Figura 3 – Fluxo de processamento de *chatbot* baseado em IA



Fonte: Adaptado pelos autores com base em Robylon.AI (2026).

Aplicado à engenharia elétrica, esse tipo de *chatbot* tende a lidar melhor com a variação natural da terminologia técnica, já que consegue captar o sentido da pergunta mesmo quando ela não segue exatamente o formato esperado — por exemplo, ao identificar o tipo de instalação, o ambiente e a aplicação descritos pelo usuário antes de formular a

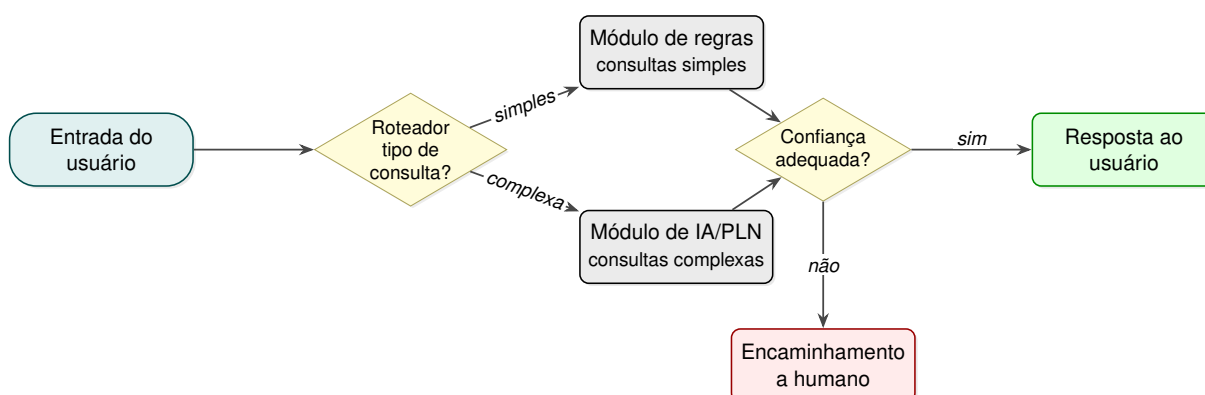
resposta (Mirto, 2023). Pesquisas sobre o uso de IA conversacional no ensino e na prática da engenharia também apontam que esse tipo de sistema amplia o acesso à informação técnica especializada, beneficiando tanto profissionais experientes quanto estudantes que ainda estão se familiarizando com as normas (Jack e Yan, 2024; University of Warwick, 2025).

Do ponto de vista técnico, esses sistemas dependem do pré-treinamento de modelos de linguagem em grandes volumes de texto, seguido de um ajuste fino com dados específicos da área de aplicação (Devlin et al., 2018). Radford et al. (2019) demonstraram que representações de linguagem treinadas de forma não supervisionada podem, posteriormente, ser adaptadas a tarefas especializadas com desempenho elevado — o que, na prática, viabiliza a construção de assistentes técnicos capazes de responder com precisão sobre normas e procedimentos de instalações elétricas, como é o caso do sistema desenvolvido neste trabalho.

2.1.3 Chatbots Híbridos

Entre as duas abordagens anteriores existe uma terceira possibilidade, que busca reunir o melhor dos dois modelos: o *chatbot* híbrido. A IBM (2025, [s. p.]) define esse tipo de sistema como “um sistema de inteligência artificial conversacional que combina lógica baseada em regras com capacidades de aprendizado de máquina”². Na prática, isso significa que os módulos baseados em regras cuidam da estrutura e da previsibilidade das respostas, enquanto a camada de IA assume as interações mais complexas ou ambíguas (IBM, 2025; Robylon.AI, 2026). A Figura 4 ilustra esse funcionamento, no qual um roteador direciona cada consulta ao módulo de regras ou ao módulo de IA conforme sua complexidade, antes de verificar a confiança na resposta e, quando necessário, encaminhá-la a um especialista humano.

Figura 4 – Fluxo de processamento de *chatbot* híbrido



Fonte: Adaptado pelos autores com base em IBM (2025) e Robylon.AI (2026).

² Tradução nossa. No original: “a conversational AI system that combines rule-based logic with machine learning capabilities”.

Para consultas técnicas em instalações elétricas de baixa tensão (ABNT, 2004), essa divisão de tarefas pode ser bastante vantajosa: regras fixas tratam de verificações simples, como a bitola mínima de um condutor ou a identificação de uma seção específica da norma, enquanto a IA assume perguntas mais elaboradas, que exigem interpretação de contexto. Outro ponto destacado pela IBM (2025) é que a arquitetura híbrida permite implementar verificações de consistência e redirecionar o usuário a um especialista humano sempre que o sistema não tiver confiança suficiente na resposta gerada — mecanismo alinhado às práticas de melhoria contínua de *chatbots* (IBM Research, 2025) e especialmente relevante em engenharia elétrica, área em que uma informação normativa imprecisa pode comprometer a segurança da instalação e de quem a utiliza.

A Tabela 1 sintetiza as principais características de cada um dos três tipos de *chatbot* apresentados, comparando vantagens, desvantagens e aplicações típicas no contexto da engenharia.

Tabela 1 – Características dos tipos de *chatbots*

Tipo	Vantagens	Desvantagens	Aplicação em Engenharia
Baseado em regras	Previsível e controlável	Inflexível e limitado	Consultas padronizadas
Baseado em IA	Flexível e natural	Imprevisível e complexo	Consultas complexas
Híbrido	Equilibrado e robusto	Maior complexidade de desenvolvimento	Aplicações profissionais

Fonte: Adaptado pelos autores com base em IBM (2025) e Robylon.AI (2026).

2.2 NBR 5410 e Literatura Técnica Especializada

2.2.1 Base de Conhecimento Técnico

Para que um *chatbot* especializado funcione bem, ele depende diretamente da qualidade da base de conhecimento sobre a qual foi construído (Chatcompose, 2023). A Easycoder (2024) reforça esse ponto ao destacar que a precisão e a atualização dos dados usados na indexação do sistema são fatores determinantes para a confiabilidade das respostas geradas. Considerando o caráter educacional deste trabalho e a utilização de recursos gratuitos em toda a sua composição, optou-se por adotar a própria ABNT NBR 5410 (ABNT, 2004), em formato PDF, como fonte normativa principal — evitando, assim, os riscos técnicos associados ao processamento simultâneo de múltiplos documentos extensos. A base de conhecimento do sistema é composta pelos seguintes elementos, extraídos diretamente da norma:

- texto completo da norma estruturado por seções, conforme organização original da ABNT NBR 5410;

- tabelas técnicas com dados de condutores de cobre, métodos de instalação, proteções e fatores de correção por temperatura e agrupamento;
- fórmulas e equações para cálculos elétricos de dimensionamento, organizadas em base de conhecimento suplementar elaborada pelos próprios autores;
- requisitos e especificações técnicas das principais seções normativas;
- anexos e referências normativas complementares presentes no documento.

2.2.2 Estrutura e Organização da NBR 5410

A NBR 5410:2004, intitulada “Instalações elétricas de baixa tensão”, é o principal documento normativo para projetos elétricos no Brasil (ABNT, 2004). Já em seu primeiro artigo, a norma deixa claro o seu propósito: “Esta Norma estabelece as condições a que devem satisfazer as instalações elétricas de baixa tensão, a fim de garantir a segurança de pessoas e animais” (ABNT, 2004, p. 1). A partir desse objetivo central, o documento se organiza em uma estrutura hierárquica que vai de conceitos fundamentais a especificações técnicas detalhadas, conforme resumido a seguir:

Seção 1 — Objetivo e campo de aplicação: estabelece o escopo normativo e suas limitações de aplicação.

Seção 2 — Referências normativas: relaciona documentos normativos complementares essenciais para a aplicação adequada da NBR 5410.

Seção 3 — Definições: a norma esclarece, já na abertura desta seção, que “para os efeitos desta Norma, aplicam-se as definições da ABNT NBR IEC 60050(826) e as seguintes” (ABNT, 2004, p. 7), reunindo o vocabulário técnico necessário para a leitura correta de todo o restante do documento.

Seções 4 a 6 — Requisitos fundamentais: abordam proteção contra choques elétricos, efeitos térmicos e sobrecorrentes, constituindo a base conceitual da norma.

Seções 7 e 8 — Critérios de projeto: estabelecem metodologias para seleção e instalação de componentes elétricos.

Seção 9 — Verificação: fecha o corpo da norma definindo os procedimentos obrigatórios de inspeção e ensaio que comprovam a conformidade da instalação executada.

2.2.3 Delimitação da Base de Conhecimento do Sistema

A literatura técnica sobre instalações elétricas é vasta, reunindo obras consolidadas como as de Cotrim (2009), Niskier e Macintyre (2013) e Creder (2016) além do Manual

Pirelli de Instalações Elétricas (Pirelli, 2018). Ainda assim, incorporar várias dessas fontes simultaneamente em formato PDF traria desafios técnicos consideráveis: maior tempo de processamento, mais consumo de recursos computacionais e risco de perda de qualidade na indexação semântica do conteúdo (Easycoder, 2024). Por isso, e levando em conta o caráter educacional deste trabalho e o uso exclusivo de recursos gratuitos em sua composição, optou-se por delimitar a base de conhecimento à própria ABNT NBR 5410, complementada por um documento suplementar elaborado pelos próprios autores em 2026. Manning, Raghavan e Schütze (2008) observam que delimitar bem o domínio de um sistema de recuperação de informação tende a aumentar a precisão das buscas, e foi exatamente esse ganho de precisão e consistência que motivou a escolha por uma fonte normativa única.

Tabela 2 – Componentes da base de conhecimento do sistema

Componente	Tipo	Conteúdo	Origem
PDF da NBR 5410	Norma oficial	Texto completo por seções, tabelas, fórmulas e anexos	ABNT (2004)
knowledge_extra.md	Documento suplementar	Fórmulas de dimensionamento, tabelas numéricas e diretrizes técnicas reestruturadas pelos autores	elaborado pelos autores

Fonte: elaborado pelos autores, 2026.

2.2.4 Equações Fundamentais de Dimensionamento Elétrico

O dimensionamento de circuitos elétricos em instalações de baixa tensão segue um conjunto de equações e dados tabulares que traduzem os requisitos normativos da NBR 5410 em grandezas calculáveis. Embora essas informações constem no texto original da norma, a sua apresentação em formato PDF — com tabelas complexas, notas de rodapé e formatação em múltiplas colunas — dificulta a fragmentação semântica exigida pelo *pipeline* RAG do sistema. Conforme Lewis et al. (2020), a qualidade da recuperação em sistemas RAG depende diretamente da granularidade e da clareza dos fragmentos indexados: dados numéricos embutidos em parágrafos extensos ou em tabelas rasterizadas tendem a ser ignorados ou mal interpretados pelo modelo de *embeddings*.

Por essa razão, as fórmulas e os valores numéricos mais utilizados no dimensionamento residencial foram reorganizados pelos autores em um documento suplementar em formato Markdown (`knowledge_extra.md`), com estrutura plana e seções curtas. Essa formatação permite que cada equação ou tabela seja recuperada como um fragmento autônomo durante a busca vetorial, aumentando a precisão das respostas geradas pelo *chatbot* (Manning, Raghavan e Schütze, 2008). São apresentadas, a seguir, as principais

equações de dimensionamento dessa base suplementar, com a descrição dos símbolos empregados.

A corrente de projeto de um circuito monofásico é obtida pela relação entre a potência ativa da carga, a tensão nominal e o fator de potência (Creder, 2016; ABNT, 2004):

$$I = \frac{P}{V \times f_p} \quad (1)$$

em que:

I = corrente de projeto do circuito (A);

P = potência ativa da carga (W);

V = tensão nominal (V);

f_p = fator de potência ($\cos \varphi$), adimensional.

Para circuitos trifásicos, o denominador inclui o fator $\sqrt{3}$ (Cotrim, 2009):

$$I = \frac{P}{\sqrt{3} \times V \times f_p} \quad (2)$$

A queda de tensão no condutor é calculada em função da resistividade do cobre, do comprimento e da seção do condutor (Creder, 2016):

$$\Delta V = \frac{2 \times \rho \times L \times I}{S} \quad (3)$$

em que:

ΔV = queda de tensão no condutor (V);

ρ = resistividade elétrica do cobre a 20 °C ($\approx 0,0172 \Omega \cdot \text{mm}^2/\text{m}$);

L = comprimento do circuito (m);

I = corrente de projeto do circuito (A);

S = seção nominal do condutor (mm^2).

O valor percentual da queda de tensão é obtido por:

$$\Delta V\% = \frac{\Delta V}{V} \times 100 \quad (4)$$

em que:

$\Delta V\%$ = queda de tensão percentual (%).

A NBR 5410 limita a queda de tensão a 4% em circuitos terminais e a 7% quando somados o circuito de alimentação e os circuitos terminais. O critério de coordenação entre a corrente de projeto, a corrente nominal do dispositivo de proteção e a capacidade de condução do condutor é:

$$I_B \leq I_n \leq I_z \quad (5)$$

em que:

I_B = corrente de projeto (de uso) do circuito (A);

I_n = corrente nominal do dispositivo de proteção (A);

I_z = capacidade de condução de corrente do condutor (A).

Quando as condições de instalação diferem das condições de referência, a capacidade de condução deve ser corrigida pelos fatores de temperatura e de agrupamento (Creder, 2016):

$$I_{z,cor} = I_z \times f_{temp} \times f_{agrup} \quad (6)$$

em que:

$I_{z,cor}$ = capacidade de condução corrigida (A);

I_z = capacidade de condução tabelada (A);

f_{temp} = fator de correção por temperatura;

f_{agrup} = fator de correção por agrupamento de circuitos.

Nas condições de referência (30 °C e circuito isolado), ambos os fatores de correção são iguais a 1,00.

Além das equações, a base suplementar inclui dados tabulares que o modelo precisa acessar de forma direta para concluir o dimensionamento. O exemplo mais recorrente é a capacidade de condução de condutores de cobre segundo o método de referência B1 (eletroduto embutido em alvenaria), apresentada de forma resumida na Tabela 3.

Tabela 3 – Capacidade de condução de corrente — condutores de cobre, isolamento PVC 70°C, método B1, 3 condutores carregados, 30 °C

Seção nominal (mm ²)	Capacidade I_z (A)
1,5	13
2,5	18
4,0	24
6,0	31
10	42
16	56
25	73
35	89
50	108
70	136
95	164
120	188

Fonte: Adaptado da NBR 5410, Tabela 36 (ABNT, 2004).

Na prática, essas equações e tabelas são aplicadas de forma encadeada. No exemplo resolvido para um ar-condicionado de 32.000 BTU/h, disponível na base suplementar, a

corrente de projeto resulta em $I = 3556 / (220 \times 0,92) \approx 17,6$ A. Consultando a Tabela 3, verifica-se que o condutor de $2,5 \text{ mm}^2$ suporta apenas 18 A — margem insuficiente —, levando à seleção de $4,0 \text{ mm}^2$ ($I_z = 24$ A). Com o critério da Equação 5, seleciona-se o disjuntor de 20 A curva C ($17,6 \leq 20 \leq 24$), e a Equação 3 confirma queda de apenas 1,4% para 20 m de circuito. Esse encadeamento ilustra como o *chatbot* articula fórmulas e dados tabulares da base suplementar para fornecer respostas completas e fundamentadas.

2.3 Inteligência Artificial Aplicada à Engenharia

2.3.1 Histórico da IA na Engenharia

A presença da inteligência artificial na engenharia elétrica não é recente. Já nas décadas de 1980 e 1990, sistemas especialistas baseados em regras eram amplamente utilizados para diagnosticar faltas em redes de distribuição, controlar tensão reativa e analisar sistemas de potência (PACW, 2024). Com o avanço do aprendizado de máquina nos anos 2000, esse uso se expandiu para a otimização de redes elétricas e para sistemas de controle inteligente (IndustriALL.AI, 2022), até chegar ao cenário atual, em que sistemas de apoio ao projeto e assistentes virtuais especializados já são capazes de interpretar normas técnicas e fornecer suporte em tempo real (Mirto, 2023; Jack e Yan, 2024).

Olhando esse percurso como um todo, é possível enxergar uma evolução contínua: dos sistemas especialistas simples da década de 1980 ao aprendizado de máquina e às redes neurais dos anos 1990, até as plataformas de processamento de linguagem natural e aprendizado profundo da atualidade (Iberdrola, 2023; Instituto de Engenharia, 2018). Essa trajetória, embora contínua, foi marcada por ciclos de entusiasmo e retração conhecidos como “invernos da IA” (Favaron, 2025). Russell e Norvig (2020) descrevem essa trajetória como um amadurecimento progressivo da IA enquanto disciplina aplicada, cada vez mais integrada aos processos reais de projeto e operação na engenharia — o que ajuda a situar historicamente a proposta deste trabalho.

2.3.2 Tecnologias de Processamento de Linguagem Natural

O Processamento de Linguagem Natural (PLN) é a área da inteligência artificial responsável por permitir que computadores compreendam e produzam linguagem humana de forma útil (Jurafsky e Martin, 2021). Bird, Klein e Loper (2009) descrevem seu objetivo central como capacitar sistemas a compreender, analisar e produzir comunicação humana de maneira eficiente e contextualizada — definição que se aplica diretamente ao tipo de sistema desenvolvido neste trabalho, voltado à interpretação de perguntas técnicas em português.

As raízes do PLN remontam à década de 1950, quando surgiram as primeiras tentativas de tradução automática entre idiomas (Jurafsky e Martin, 2021). É também dessa

época um dos marcos mais conhecidos da área: o Teste de Turing. No artigo em que o propôs, Turing (1950, p. 433) abre com a pergunta que viria a guiar décadas de pesquisa em IA: “proponho considerar a questão: ‘As máquinas podem pensar?’”³. Embora a pergunta de Turing fosse mais filosófica do que técnica, ela ajudou a estabelecer a linguagem natural como um dos principais critérios para avaliar a inteligência de uma máquina (Russell e Norvig, 2020). Décadas depois, o campo deu um salto de qualidade com a chegada do aprendizado profundo e das redes neurais (Devlin et al., 2018; Radford et al., 2019), que tornaram possível atingir um nível de precisão impensável nos primeiros sistemas baseados em regras.

Hoje, o PLN está presente em ferramentas usadas no dia a dia, da análise de sentimento em redes sociais à tradução automática de textos inteiros (Asimov Academy, 2024). Para este trabalho, no entanto, a aplicação mais relevante é outra: a análise e a recuperação de informações em documentos técnicos extensos, como normas e regulamentos (Manning, Raghavan e Schütze, 2008). Isso envolve técnicas como categorização de documentos, síntese automática e reconhecimento de entidades específicas dentro do texto (Bird, Klein e Loper, 2009) — recursos que, no sistema aqui desenvolvido, viabilizam a localização dos trechos mais relevantes da ABNT NBR 5410 a cada nova pergunta feita pelo usuário.

2.3.3 Conceitos Fundamentais do PLN

Alguns conceitos ajudam a entender, de forma mais concreta, como um sistema de PLN processa um texto antes de gerar uma resposta (Jurafsky e Martin, 2021).

Tokenização: é o processo de quebrar um texto em pedaços menores, chamados *tokens*, que podem ser palavras, expressões ou símbolos (Bird, Klein e Loper, 2009). É geralmente o primeiro passo de qualquer sistema de PLN. Por exemplo, a frase “Eu adoro aprender sobre PLN” seria dividida em: [“Eu”, “adoro”, “aprender”, “sobre”, “PLN”].

Stemming e lematização: são duas formas de reduzir uma palavra à sua forma mais básica (Jurafsky e Martin, 2021). O *stemming* corta sufixos de maneira mais mecânica, enquanto a lematização usa um dicionário para chegar à forma canônica da palavra (Bird, Klein e Loper, 2009). Assim, “correr”, “correndo” e “correu” acabam reduzidos à mesma raiz: “correr”.

Análise sintática e semântica: a análise sintática (*parsing*) organiza a frase segundo regras gramaticais, identificando a relação entre as palavras (Jurafsky e Martin, 2021), enquanto a análise semântica vai além e atribui significado a esse conteúdo. Na frase “O gato está no telhado”, o *parsing* identifica “gato” como sujeito e “telhado” como complemento, e a semântica é o que permite ao sistema entender que se trata de uma relação espacial entre os dois.

³ Tradução nossa. No original: “I propose to consider the question, ‘Can machines think?’”.

Reconhecimento de entidades nomeadas (NER): consiste em localizar e classificar nomes próprios, datas, lugares e organizações dentro de um texto (Bird, Klein e Loper, 2009). Na frase “Albert Einstein nasceu na Alemanha”, um sistema de NER reconheceria “Albert Einstein” como pessoa e “Alemanha” como local.

Desambiguação de sentido lexical (WSD): trata da tarefa de descobrir qual significado de uma palavra está sendo usado, com base no contexto (Jurafsky e Martin, 2021). A palavra “banco”, por exemplo, pode significar uma instituição financeira ou um assento, e é o contexto da frase que indica qual dos dois sentidos está em jogo.

Esses conceitos não ficam restritos à teoria: vários deles aparecem, de forma prática, no funcionamento do sistema desenvolvido neste trabalho. A tokenização e a redução morfológica entram no pré-processamento das perguntas dos usuários, o NER ajuda a identificar parâmetros técnicos como potência, tensão e tipo de instalação, e a análise semântica é o que sustenta a busca por similaridade vetorial, permitindo que uma pergunta em linguagem natural encontre os trechos certos da ABNT NBR 5410.

2.3.4 Sistemas de Recuperação de Informação

Se o PLN cuida de interpretar a pergunta, os sistemas de recuperação de informação são responsáveis pela outra metade do trabalho: localizar, dentro da base de conhecimento, os trechos que realmente respondem àquela pergunta (Manning, Raghavan e Schütze, 2008). Em um contexto de engenharia elétrica, isso significa que o sistema precisa ser capaz de:

- processar consultas técnicas complexas envolvendo múltiplos parâmetros (como potência, tipo de instalação e ambiente, entre outros) (Manning, Raghavan e Schütze, 2008);
- identificar rapidamente seções pertinentes da NBR 5410 e referências complementares à literatura técnica, mesmo em documentos extensos e variados;
- considerar o contexto e a aplicação específica apresentada pelo usuário, refinando a busca para maior relevância normativa e prática (Manning, Raghavan e Schütze, 2008).

Existem diferentes caminhos técnicos para se alcançar essa precisão. Entre os mais relevantes para este trabalho estão:

- indexação semântica de documentos, que organiza o conteúdo técnico de forma que os conceitos estejam acessíveis por significado, e não apenas por palavras-chave (Manning, Raghavan e Schütze, 2008);

- busca neural com modelos pré-treinados, capazes de compreender linguagem natural e contexto técnico, promovendo recomendações mais precisas (Devlin et al., 2018);
- busca baseada em *embeddings*, representando trechos textuais em vetores semânticos gerados por modelos de linguagem pré-treinados, o que permite correspondência mais fina e contextualizada com a consulta do usuário (Devlin et al., 2018; Vaswani et al., 2017);
- sistemas RAG (*Retrieval-Augmented Generation*), que integram mecanismos de geração de respostas a módulos de recuperação externa, possibilitando respostas fundamentadas e contextualizadas, especialmente em bases normativas extensas (Lewis et al., 2020).

É justamente essa última abordagem, a arquitetura RAG, que estrutura o funcionamento do sistema desenvolvido neste trabalho. Na prática, ela se concretiza por meio do ChromaDB (Chroma, 2025) como base vetorial, do modelo *gemini-embedding-001* (Google, 2025) para a geração dos *embeddings* semânticos e do LangGraph (LangGraph, 2025; LangChain, 2025) orquestrando o fluxo do agente — garantindo que cada consulta do usuário recupere os trechos normativos mais relevantes da NBR 5410 antes de qualquer resposta ser gerada (Lewis et al., 2020; Manning, Raghavan e Schütze, 2008).

3 PROCEDIMENTOS METODOLÓGICOS

O desenvolvimento do trabalho foi organizado em duas etapas principais: uma revisão bibliográfica, que serviu como base teórica para o projeto, e o desenvolvimento do sistema em si, que envolveu desde a escolha das tecnologias até a construção da interface e os testes com usuários.

3.1 Pesquisa Bibliográfica

A pesquisa bibliográfica foi o ponto de partida do trabalho. Foram estudados conceitos relacionados ao processamento de linguagem natural, à inteligência artificial aplicada à engenharia e ao funcionamento de sistemas de recuperação de informação. Além disso, foi feita uma análise detalhada da estrutura e do conteúdo da ABNT NBR 5410 (ABNT, 2004), norma que regulamenta as instalações elétricas de baixa tensão no Brasil e que constitui a base de conhecimento do sistema desenvolvido.

Essa etapa foi essencial para entender como organizar o conteúdo da norma de forma que o sistema pudesse recuperá-lo de maneira eficiente, e para fundamentar as escolhas técnicas feitas ao longo do desenvolvimento.

3.2 Desenvolvimento do Sistema

3.2.1 Visão Geral da Arquitetura

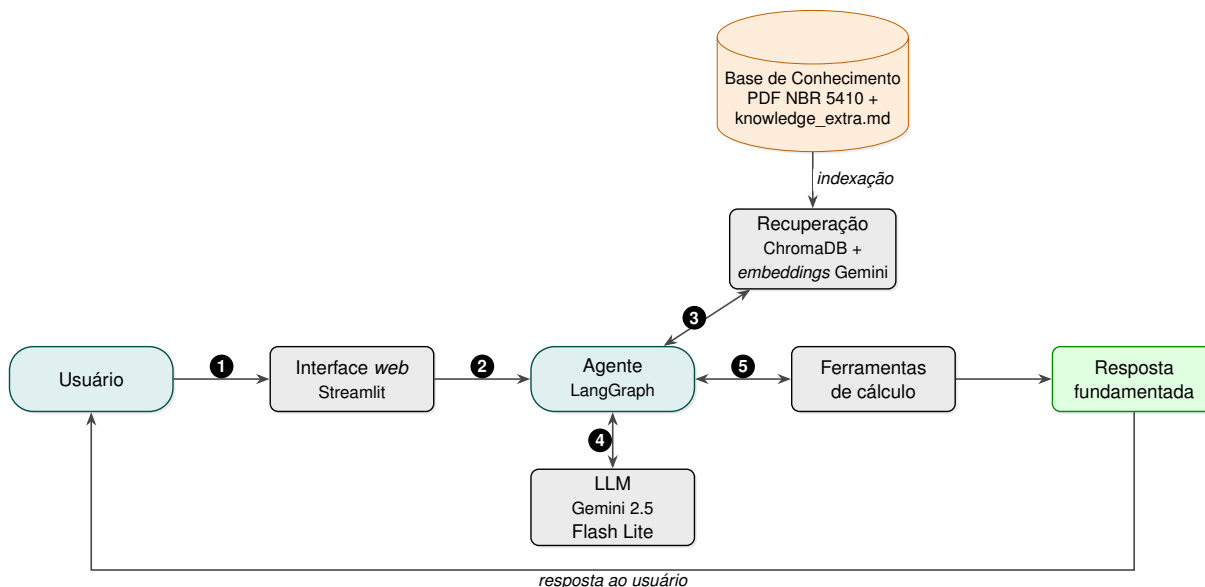
O sistema foi desenvolvido em linguagem Python e é composto por três partes principais que trabalham juntas: a interface com o usuário, o núcleo de processamento inteligente e a base de conhecimento. A comunicação entre essas partes foi construída com o auxílio do *framework* LangGraph (LangGraph, 2025), que permite definir o fluxo de execução do agente de forma organizada e controlada.

De forma simplificada, quando o usuário faz uma pergunta, o sistema busca os trechos mais relevantes da NBR 5410 armazenados em sua base de conhecimento e, em seguida, utiliza essas informações para gerar uma resposta fundamentada. Caso a pergunta não tenha relação com o tema da norma, o sistema informa ao usuário que não pode ajudar com aquele assunto. A Figura 5 ilustra essa arquitetura, baseada no padrão de Geração Aumentada por Recuperação (RAG).

3.2.2 Construção da Base de Conhecimento

A base de conhecimento do sistema foi construída a partir do documento oficial da ABNT NBR 5410 em formato PDF. O conteúdo da norma foi extraído e dividido em trechos menores, de forma a facilitar a busca por informações específicas durante as consultas.

Figura 5 – Arquitetura RAG do sistema desenvolvido



Fonte: elaborado pelos autores, 2026.

Para que o sistema fosse capaz de entender o significado das perguntas, e não apenas identificar palavras iguais, cada trecho da norma foi transformado em uma representação matemática — chamada de vetor semântico, ou *embedding* — utilizando um modelo de inteligência artificial fornecido pela Google (Google, 2025). Esses vetores ficam armazenados em um banco de dados vetorial, o ChromaDB (Chroma, 2025), e, quando o usuário faz uma pergunta, o sistema compara o significado da pergunta com o conteúdo armazenado e recupera os trechos mais relevantes.

Além do PDF da norma, foi elaborado pelos próprios autores um documento complementar contendo tabelas de condutores, fórmulas de dimensionamento e diretrizes técnicas extraídas da NBR 5410. Esse documento foi criado porque algumas informações da norma, como tabelas numéricas e equações, não são extraídas corretamente quando se lê um PDF de forma automatizada. Ao organizar esse conteúdo manualmente, garantiu-se que essas informações ficassem disponíveis com qualidade para o sistema.

3.2.3 O Agente Inteligente e as Ferramentas de Cálculo

O núcleo inteligente do sistema funciona como um agente que recebe a pergunta do usuário, decide quais informações buscar e, quando necessário, aciona ferramentas de cálculo para fornecer uma resposta mais precisa. Foram implementadas quatro ferramentas de cálculo integradas ao agente:

- cálculo de corrente elétrica: calcula a corrente de projeto a partir da potência, da tensão e do fator de potência informados pelo usuário;

- seleção de condutor: indica a bitola mínima de condutor de cobre conforme a Tabela 36 da NBR 5410, considerando o método de instalação e a temperatura ambiente;
- cálculo de queda de tensão: verifica se a queda de tensão do circuito está dentro dos limites estabelecidos pela NBR 5410;
- conversão de BTU/h para Watts: converte a capacidade de equipamentos de ar-condicionado de BTU/h para potência elétrica em Watts, facilitando o dimensionamento do circuito correspondente.

O modelo de linguagem utilizado foi o Gemini 2.5 Flash Lite, da Google (Google, 2025), configurado para gerar respostas objetivas e consistentes. O agente foi programado para sempre apresentar o resultado direto antes das explicações, utilizar as ferramentas de cálculo quando a pergunta envolver dimensionamento e recusar gentilmente perguntas fora do escopo da NBR 5410.

3.2.4 Interface de Usuário

A interface do sistema foi desenvolvida com o *framework* Streamlit, escolhido por sua praticidade e por permitir criar aplicações *web* de forma rápida e sem necessidade de conhecimentos avançados em desenvolvimento *web*. A aplicação foi publicada na plataforma Streamlit Cloud, tornando-a acessível por qualquer navegador, sem necessidade de instalação.

A tela principal apresenta um campo para o usuário digitar sua dúvida em linguagem natural, um painel lateral com os tópicos suportados pelo sistema e, após cada resposta, a indicação dos trechos da norma que foram utilizados como base. Essa transparência permite ao usuário verificar a origem das informações fornecidas.

3.2.5 Hospedagem e Código-fonte

O código-fonte do projeto foi versionado e armazenado em repositório público no GitHub, seguindo boas práticas de organização e controle de versão. A interface *web* foi publicada no Streamlit Cloud, plataforma gratuita que viabilizou o acesso ao sistema sem custos adicionais. A base de conhecimento vetorial foi armazenada no Hugging Face Hub, sendo carregada automaticamente pelo sistema na inicialização da aplicação.

O sistema encontra-se disponível publicamente para acesso e testes no seguinte endereço: <<https://nbr5410-agent-dtdsdlf8ecibhnnmbikfe.streamlit.app/>>. Os códigos-fonte que compõem a aplicação são reproduzidos nos Apêndices A, B e C.

3.3 Avaliação do Sistema

Para avaliar o funcionamento do sistema desenvolvido, a ferramenta foi disponibilizada, por meio de seu endereço *web*, a um grupo de 10 a 20 pessoas, entre estudantes e profissionais da área de engenharia elétrica. Os participantes puderam acessá-la livremente pelo navegador e submeter consultas técnicas de sua escolha, sem roteiro predefinido, de modo a reproduzir situações reais de uso.

A avaliação teve caráter qualitativo: foram coletados os comentários e as percepções dos usuários quanto à clareza, à objetividade e à utilidade das respostas. Em paralelo, registraram-se casos de consulta representativos, abrangendo diferentes níveis de complexidade de dimensionamento elétrico, para análise posterior. Esses casos e o retorno dos usuários são apresentados e discutidos no Capítulo 4.

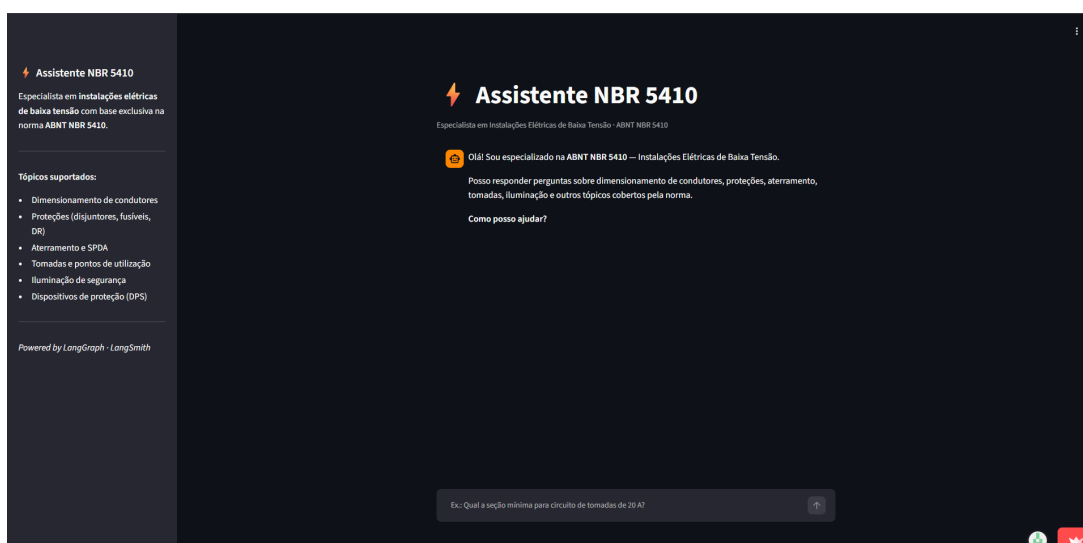
4 RESULTADOS E DISCUSSÃO

A partir da avaliação descrita na Seção 3.3, os *feedbacks* coletados foram positivos: os usuários descreveram o sistema como “bom” e “ótima ferramenta, bem simples e resumida”, o que indica que a interface e a objetividade das respostas atenderam bem ao público-alvo.

A seguir são apresentados os principais casos testados durante a validação do sistema, com as respectivas capturas de tela das interações.

4.1 Interface do Sistema

Figura 6 – Interface do Assistente NBR 5410



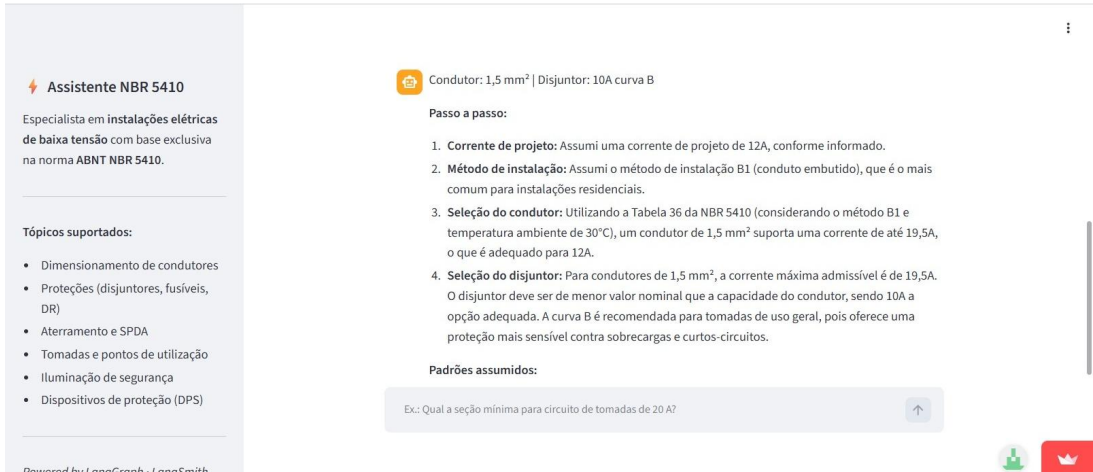
Fonte: elaborado pelos autores, 2026.

As Figuras 6 e 7 mostram a tela principal do sistema, na qual o usuário visualiza os tópicos suportados no painel lateral e interage com o assistente pelo campo de texto central. A interface foi desenvolvida para ser direta e acessível, sem exigir conhecimento técnico avançado para utilizá-la.

4.2 Consultas Técnicas Realizadas

Durante os testes, os usuários submeteram perguntas de diferentes níveis de complexidade, todas relacionadas a instalações elétricas de baixa tensão. O sistema demonstrou capacidade de interpretar as consultas em linguagem natural e fornecer respostas fundamentadas nos requisitos da NBR 5410, utilizando as ferramentas de cálculo quando necessário.

Figura 7 – Interface do Assistente NBR 5410 — visão geral

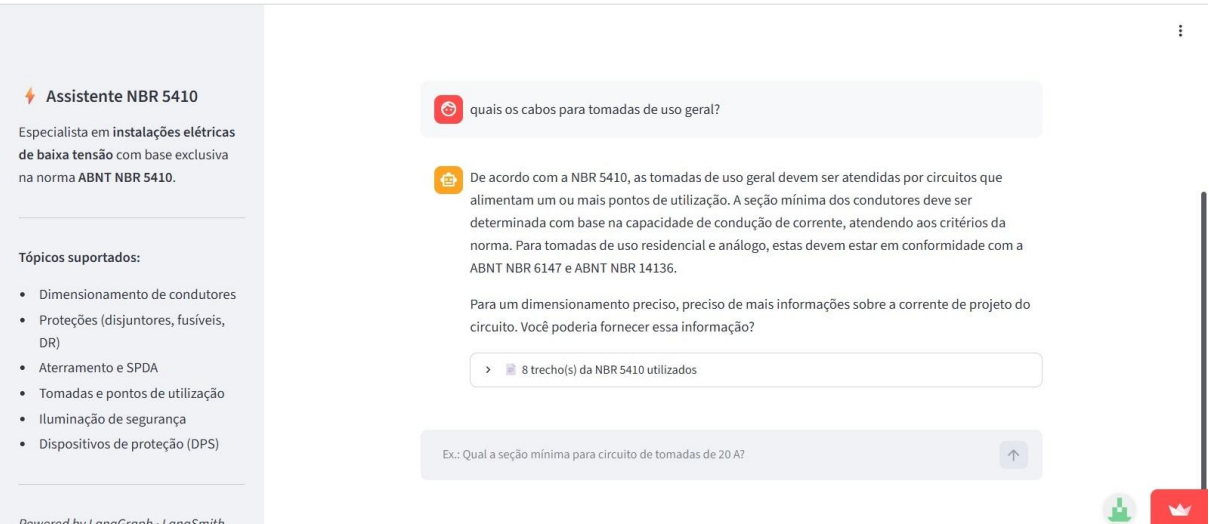


Fonte: elaborado pelos autores, 2026.

4.2.1 Cabos para Tomadas de Uso Geral

Em uma das primeiras interações, um usuário perguntou sobre os cabos adequados para tomadas de uso geral. O sistema respondeu referenciando os critérios da NBR 5410 e solicitou ao usuário a corrente de projeto do circuito para que pudesse fornecer um dimensionamento mais preciso — comportamento correto, pois a norma exige esses parâmetros para o dimensionamento adequado. A Figura 8 mostra essa interação.

Figura 8 – Consulta sobre cabos para tomadas de uso geral



Fonte: elaborado pelos autores, 2026.

4.2.2 Seção Mínima para Circuito de Tomadas de 20 A

Quando questionado sobre a seção mínima de condutor para um circuito de tomadas com corrente de 20 A, o sistema apresentou uma resposta organizada em etapas, identi-

ficando a norma aplicável, consultando os dados da NBR 5410 e concluindo pela seção mínima de 2,5 mm² para condutores de cobre. A resposta também incluiu os parâmetros assumidos e uma observação sobre a necessidade de considerar o método de instalação e a temperatura para o dimensionamento definitivo, conforme mostra a Figura 9.

Figura 9 – Consulta sobre seção mínima para circuito de tomadas de 20 A

2. Calcular a corrente:

- Assumi uma tensão de 220V (padrão comum no Brasil para ar condicionado) e fator de potência de 0,92 (típico para ar condicionado).
- `corrente_a = 12000 W / (220 V * 0.92) = 5.9 A`

3. Selecionar o condutor:

- A corrente de projeto é de 5.9 A.
- Assumi o método de instalação B1 (conduto embutido), que é o mais comum.
- Assumi 1 circuito agrupado.
- Consultando as tabelas da NBR 5410 (ou usando a ferramenta de cálculo), para uma corrente de 5.9A em método B1, a seção mínima do condutor é de 2,5 mm².

Padrões assumidos:

- Tensão: 220V
- Fator de potência: 0,92
- Método de instalação: B1 (conduto embutido)
- EER: 10

Observação: Este dimensionamento considera apenas o cabo de alimentação. Para o disjuntor, seria necessário um de 10A (valor comercial imediatamente superior à corrente calculada, conforme NBR 5410). Recomenda-se sempre consultar um profissional qualificado para um projeto elétrico completo e seguro.

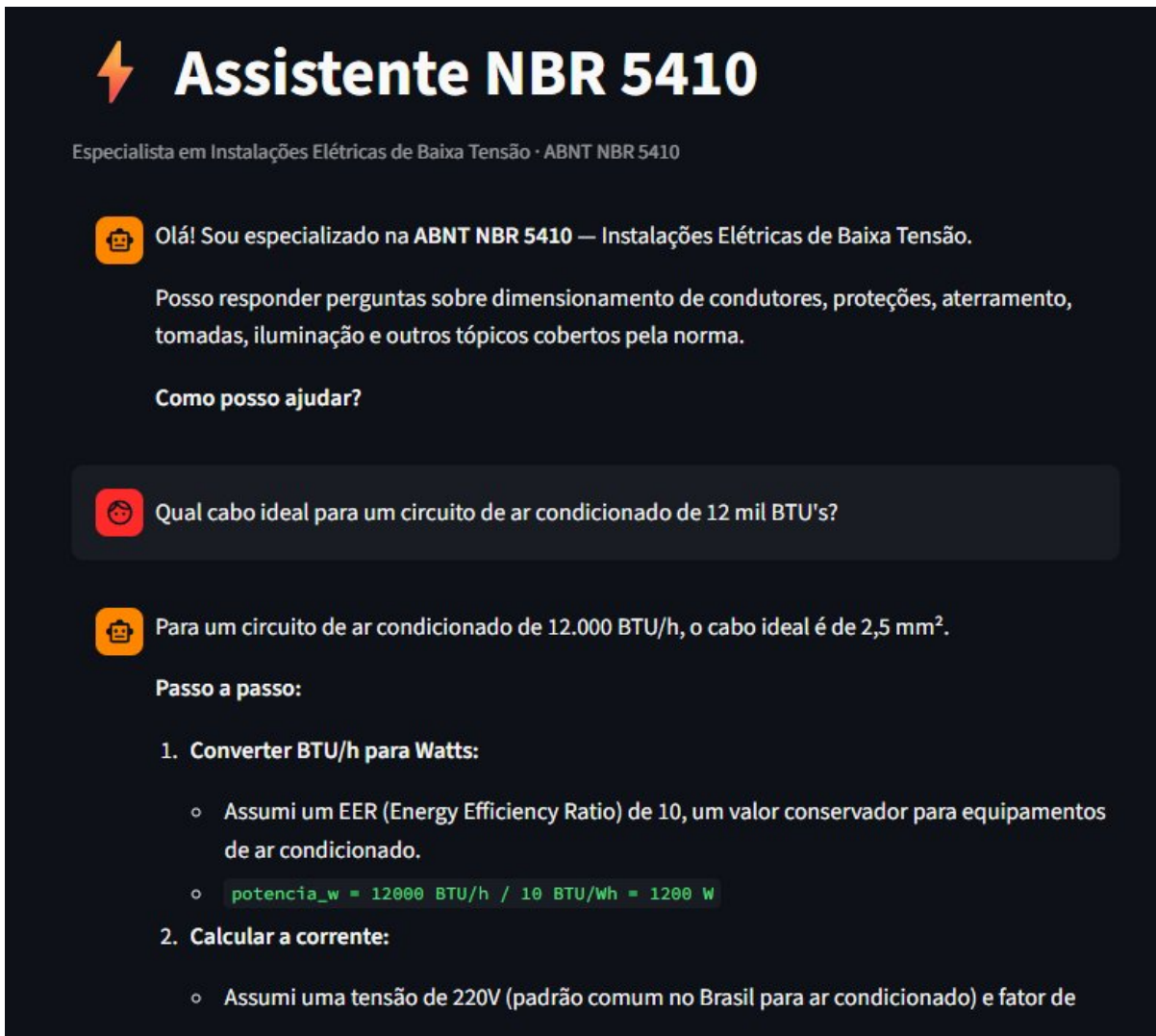
> 8 trecho(s) da NBR 5410 utilizados

Fonte: elaborado pelos autores, 2026.

4.2.3 Condutor para Ar-condicionado de 12.000 BTU/h

Uma das consultas de maior complexidade envolveu o dimensionamento do condutor para um circuito de ar-condicionado de 12.000 BTU/h. O sistema converteu automaticamente a capacidade do equipamento para potência em Watts, calculou a corrente elétrica do circuito e selecionou a bitola mínima de 2,5 mm², apresentando cada etapa do raciocínio de forma clara e explicando os valores assumidos. As Figuras 10 e 11 mostram essa interação.

Figura 10 – Consulta sobre condutor para ar-condicionado de 12.000 BTU/h



⚡ Assistente NBR 5410

Especialista em Instalações Elétricas de Baixa Tensão · ABNT NBR 5410

Olá! Sou especializado na ABNT NBR 5410 — Instalações Elétricas de Baixa Tensão.

Posso responder perguntas sobre dimensionamento de condutores, proteções, aterramento, tomadas, iluminação e outros tópicos cobertos pela norma.

Como posso ajudar?

Qual cabo ideal para um circuito de ar condicionado de 12 mil BTU's?

Para um circuito de ar condicionado de 12.000 BTU/h, o cabo ideal é de 2,5 mm².

Passo a passo:

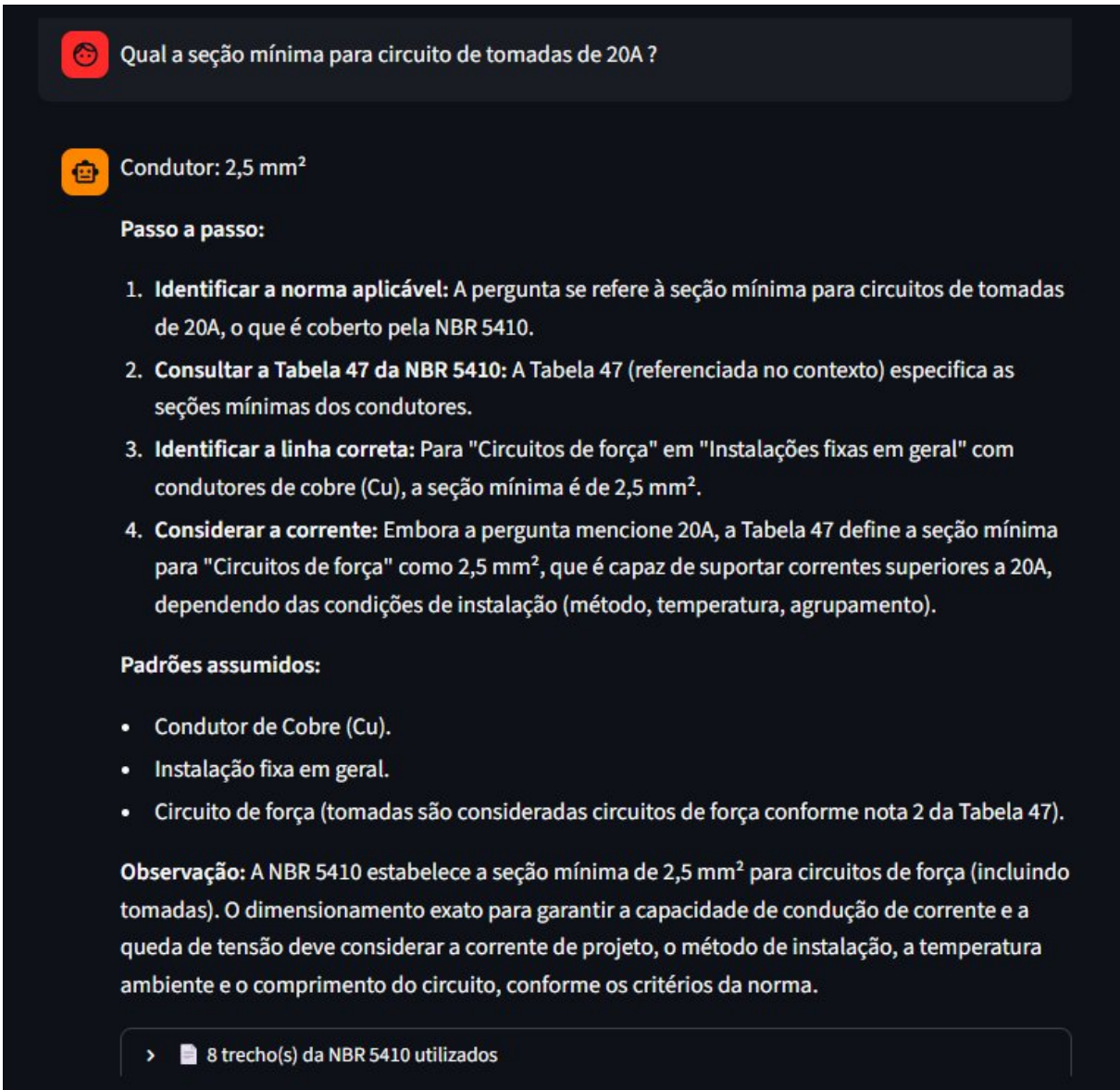
- 1. Converter BTU/h para Watts:**
 - Assumi um EER (Energy Efficiency Ratio) de 10, um valor conservador para equipamentos de ar condicionado.
 - `potencia_w = 12000 BTU/h / 10 BTU/Wh = 1200 W`
- 2. Calcular a corrente:**
 - Assumi uma tensão de 220V (padrão comum no Brasil para ar condicionado) e fator de

Fonte: elaborado pelos autores, 2026.

4.2.4 Bitola de Cabo para Circuito de 150 m com Corrente de 15 A

Outro caso testado foi a definição da bitola de cabo para um circuito de 150 metros com corrente de 15 A. O sistema identificou os dados informados, assumiu os parâmetros padrão de instalação e retornou a bitola de 4 mm² como resultado. Além disso, emitiu alertas importantes: informou que a NBR 5410 recomenda queda de tensão máxima de 4% em circuitos terminais e que seria necessário o comprimento exato do circuito para verificar esse requisito. Esse comportamento mostra que o sistema entende seus limites e não apresenta resultados incompletos como se fossem definitivos. As Figuras 12, 13 e 14 ilustram essa interação.

Figura 11 – Continuação: cálculo de corrente e seleção do condutor



Qual a seção mínima para circuito de tomadas de 20A ?

Condutor: 2,5 mm²

Passo a passo:

1. **Identificar a norma aplicável:** A pergunta se refere à seção mínima para circuitos de tomadas de 20A, o que é coberto pela NBR 5410.
2. **Consultar a Tabela 47 da NBR 5410:** A Tabela 47 (referenciada no contexto) especifica as seções mínimas dos condutores.
3. **Identificar a linha correta:** Para "Circuitos de força" em "Instalações fixas em geral" com condutores de cobre (Cu), a seção mínima é de 2,5 mm².
4. **Considerar a corrente:** Embora a pergunta mencione 20A, a Tabela 47 define a seção mínima para "Circuitos de força" como 2,5 mm², que é capaz de suportar correntes superiores a 20A, dependendo das condições de instalação (método, temperatura, agrupamento).

Padrões assumidos:

- Condutor de Cobre (Cu).
- Instalação fixa em geral.
- Circuito de força (tomadas são consideradas circuitos de força conforme nota 2 da Tabela 47).

Observação: A NBR 5410 estabelece a seção mínima de 2,5 mm² para circuitos de força (incluindo tomadas). O dimensionamento exato para garantir a capacidade de condução de corrente e a queda de tensão deve considerar a corrente de projeto, o método de instalação, a temperatura ambiente e o comprimento do circuito, conforme os critérios da norma.

> 8 trecho(s) da NBR 5410 utilizados

Fonte: elaborado pelos autores, 2026.

4.3 Controle do Escopo da Ferramenta

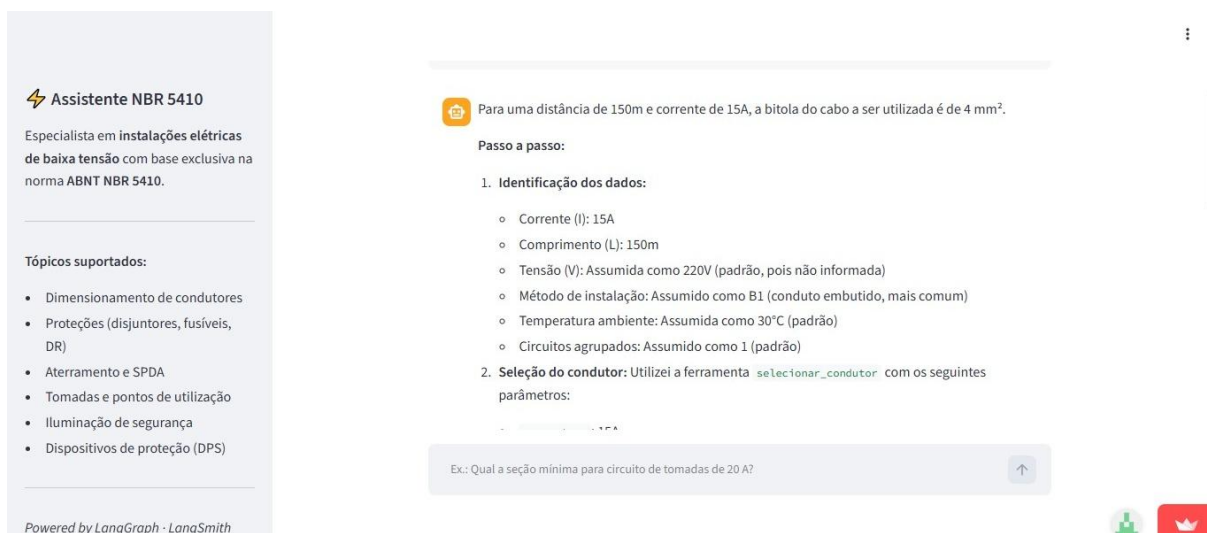
Um aspecto importante observado nos testes foi o comportamento do sistema diante de perguntas fora do escopo técnico. Quando um usuário perguntou sobre os números da Mega-Sena, o sistema recusou-se a responder de forma adequada, informando que só pode auxiliar com questões relacionadas à NBR 5410 e ao dimensionamento elétrico. Esse comportamento, mostrado na Figura 15, é fundamental para garantir a confiabilidade da ferramenta, evitando que ela seja utilizada fora de seu propósito técnico.

Figura 12 – Consulta sobre bitola de cabo para circuito de 150 m e 15 A



Fonte: elaborado pelos autores, 2026.

Figura 13 – Resultado: seleção do condutor e alertas normativos



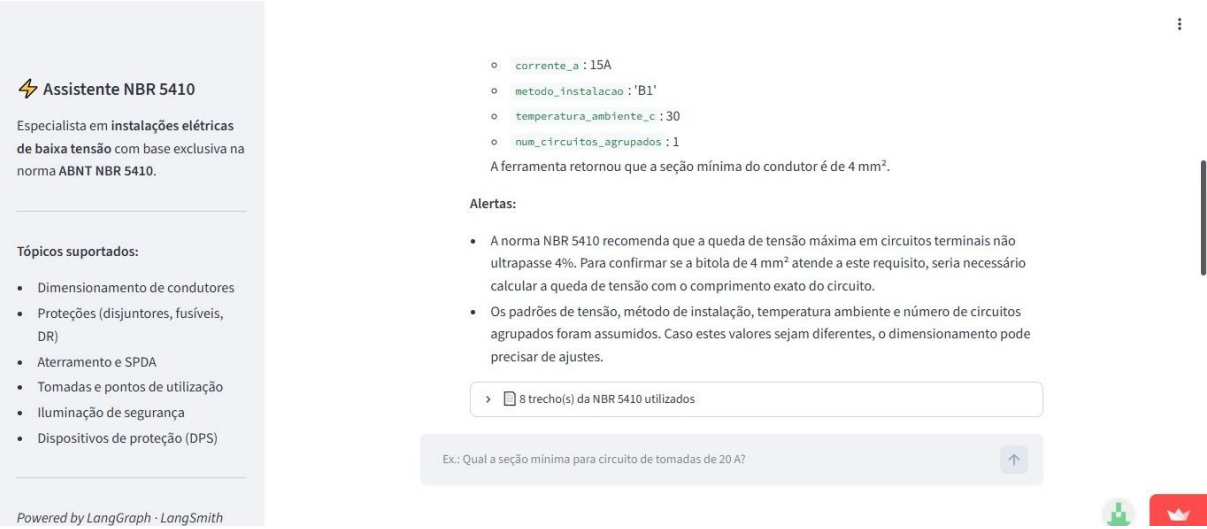
Fonte: elaborado pelos autores, 2026.

4.4 Considerações Gerais

De forma geral, os testes realizados demonstraram que o sistema atende satisfatoriamente aos objetivos propostos. As respostas foram tecnicamente corretas, fundamentadas na NBR 5410, e apresentadas de forma clara e acessível tanto para estudantes quanto para profissionais. A interface simples e objetiva foi bem recebida pelos usuários, que destacaram a facilidade de uso como um ponto positivo da ferramenta.

A principal limitação identificada está relacionada ao escopo da base de conhecimento, que foi construída exclusivamente a partir da NBR 5410. Em consultas que envolvem normas complementares — como as que tratam de tomadas, dispositivos de proteção contra

Figura 14 – Continuação: parâmetros assumidos e alertas finais



Fonte: elaborado pelos autores, 2026.

Figura 15 – Recusa do sistema a consulta fora do escopo técnico



Fonte: elaborado pelos autores, 2026.

surtos ou sistemas de aterramento — o sistema pode solicitar informações adicionais ao usuário ou fornecer respostas mais genéricas. Essa limitação não compromete a utilidade da ferramenta, mas aponta um caminho natural para melhorias futuras, como a inclusão de outras normas técnicas relacionadas ao tema.

5 CONSIDERAÇÕES FINAIS

Este trabalho apresentou o desenvolvimento de um sistema inteligente de consulta à ABNT NBR 5410, implementado por meio de um agente conversacional especializado em instalações elétricas de baixa tensão. Trata-se de uma contribuição prática e relevante para a área de engenharia elétrica, pois oferece uma solução ágil, acessível e fundamentada diretamente no conteúdo normativo vigente, permitindo que engenheiros, técnicos e estudantes realizem consultas técnicas em linguagem natural, sem a necessidade de percorrer manualmente as 217 páginas da norma.

Em um cenário no qual a precisão técnica e a conformidade normativa são exigências constantes no desenvolvimento de projetos elétricos, a ferramenta desenvolvida se destaca por democratizar o acesso ao conhecimento da NBR 5410, reduzindo o tempo de consulta e apoiando tanto a formação acadêmica quanto o exercício profissional.

O impacto deste projeto vai além do contexto acadêmico: o sistema desenvolvido inaugura uma nova forma de acessar o conhecimento normativo em instalações elétricas, promovendo autonomia técnica, agilidade na tomada de decisões e maior segurança na elaboração de projetos. Para os profissionais, representa um ganho de produtividade ao permitir consultas rápidas e fundamentadas. Para estudantes e futuros engenheiros, apresenta-se como instrumento pedagógico, auxiliando na compreensão prática dos requisitos normativos e na aplicação das diretrizes técnicas no dia a dia.

A principal limitação identificada foi o escopo restrito da base de conhecimento, construída exclusivamente a partir da NBR 5410. Consultas que envolvem normas complementares podem resultar em respostas mais genéricas ou na solicitação de informações adicionais ao usuário. Esse aspecto aponta caminhos naturais para trabalhos futuros, como a ampliação da base com outras normas técnicas da área elétrica, a incorporação de novos tópicos ao documento suplementar e a realização de testes com grupos maiores e mais diversificados de usuários.

Ao unir inteligência artificial e engenharia elétrica com propósito educacional e técnico, este trabalho contribui para a modernização da prática profissional e promove uma cultura de acesso facilitado à informação normativa — algo ainda raro e muitas vezes restrito por barreiras de tempo ou de familiaridade com os textos das normas. O sistema desenvolvido surge, assim, como um passo significativo rumo a uma engenharia mais conectada, acessível e eficiente.

5.1 Trabalhos Futuros

Este trabalho também abre espaço para algumas possibilidades simples de continuidade. Entre as sugestões estão:

- ampliar a base de conhecimento com outras normas relacionadas à NBR 5410, como a NBR 5419, a NBR 6147 e a NBR 14136;
- incluir mais livros e manuais técnicos da área no documento suplementar elaborado pelos autores;
- adicionar novas ferramentas de cálculo ao agente, como o dimensionamento de disjuntores e a queda de tensão em circuitos com múltiplas cargas.

Esperamos que esse projeto sirva de ponto de partida para outras iniciativas parecidas, que continuem aproximando a tecnologia da prática da engenharia.

REFERÊNCIAS

ABNT. *NBR 5410: instalações elétricas de baixa tensão*. Rio de Janeiro: ABNT, 2004. Citado 6 vezes nas páginas 12, 18, 19, 21, 22 e 27.

ASIMOV ACADEMY. *Processamento de linguagem natural (PLN)*. 2024. Disponível em: <<https://hub.asimov.academy/blog/processamento-linguagem-natural-pln/>>. Acesso em: 18 jun. 2025. Citado na página 24.

BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Processing with Python: analyzing text with the natural language toolkit*. Sebastopol: O'Reilly Media, 2009. Citado 3 vezes nas páginas 23, 24 e 25.

CHATCOMPOSE. *Entendendo a arquitetura dos chatbots*. 2023. Disponível em: <<https://www.chatcompose.com/PT/arquitetura.html>>. Acesso em: 25 jul. 2025. Citado 3 vezes nas páginas 14, 15 e 18.

CHROMA. *Chroma: the open-source embedding database*. 2025. Disponível em: <<https://www.trychroma.com>>. Acesso em: 10 set. 2025. Citado 2 vezes nas páginas 26 e 28.

COTRIM, A. A. M. B. *Instalações Elétricas*. 5. ed. São Paulo: Pearson, 2009. Citado 2 vezes nas páginas 19 e 21.

CREDER, H. *Instalações Elétricas*. 16. ed. Rio de Janeiro: LTC, 2016. Citado 3 vezes nas páginas 19, 21 e 22.

DEVLIN, J. et al. *BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding*. 2018. ArXiv preprint arXiv:1810.04805. Disponível em: <<https://arxiv.org/abs/1810.04805>>. Acesso em: 27 ago. 2025. Citado 3 vezes nas páginas 17, 24 e 26.

EASYCODER. *Mastering domain-specific chatbots*. 2024. Disponível em: <<https://www.easycoder.co/blog/mastering-domain-specific-chatbots>>. Acesso em: 25 jul. 2025. Citado 3 vezes nas páginas 14, 18 e 20.

FAVARON, G. *Os invernos da IA: ciclos de ascensão e queda na história da inteligência artificial*. 2025. Disponível em: <<https://www.guilhermefavaron.com.br/post/os-invernos-da-ia-ciclos-de-ascensao-e-queda-na-historia-da-inteligencia-artificial>>. Acesso em: 18 jun. 2025. Citado na página 23.

GOOGLE. *Google Generative AI: Gemini API*. 2025. Disponível em: <<https://ai.google.dev>>. Acesso em: 10 set. 2025. Citado 3 vezes nas páginas 26, 28 e 29.

IBERDROLA. *História da inteligência artificial*. 2023. Disponível em: <<https://www.iberdrola.com/quem-somos/nosso-modelo-inovacao/historia-inteligencia-artificial>>. Acesso em: 20 jul. 2025. Citado na página 23.

IBM. *Chatbot types*. 2025. Disponível em: <<https://www.ibm.com/think/topics/chatbot-types>>. Acesso em: 10 jun. 2025. Citado 3 vezes nas páginas 15, 17 e 18.

IBM RESEARCH. *Continuous Improvement of Chatbots*. 2025. Disponível em: <<https://research.ibm.com/projects/continuous-improvement-of-chatbots>>. Acesso em: 10 jun. 2025. Citado na página 18.

INDUSTRIALL.AI. *Entenda a história da IA: afinal, desde quando ela existe*. 2022. Disponível em: <<https://industrial.ai/blog/historia-inteligencia-artificial>>. Acesso em: 18 jun. 2025. Citado na página 23.

INSTITUTO DE ENGENHARIA. *A história da inteligência artificial*. 2018. Disponível em: <<https://www.institutodeengenharia.org.br/site/2018/10/29/a-historia-da-inteligencia-artificial/>>. Acesso em: 12 jun. 2025. Citado na página 23.

JACK, H.; YAN, Y. The use of chatbots in engineering including critical thinking and problem definition. In: *American Society for Engineering Education Annual Conference*. Washington: ASEE, 2024. Disponível em: <<https://peer.asee.org/the-use-of-chatbots-in-engineering-including-critical-thinking-and-problem-definition>>. Acesso em: 25 jul. 2025. Citado 3 vezes nas páginas 14, 17 e 23.

JURAFSKY, D.; MARTIN, J. H. *Speech and Language Processing: an introduction to natural language processing, computational linguistics, and speech recognition*. 3. ed. Stanford: Stanford University, 2021. Citado 4 vezes nas páginas 16, 23, 24 e 25.

LANGCHAIN. *LangChain: Build context-aware reasoning applications*. 2025. Disponível em: <<https://www.langchain.com>>. Acesso em: 10 set. 2025. Citado na página 26.

LANGGRAPH. *LangGraph: Build stateful, multi-step applications*. 2025. Disponível em: <<https://langchain-ai.github.io/langgraph>>. Acesso em: 10 set. 2025. Citado 2 vezes nas páginas 26 e 27.

LEWIS, P. et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, v. 33, p. 9459–9474, 2020. Citado 2 vezes nas páginas 20 e 26.

MANNING, C. D.; RAGHAVAN, P.; SCHÜTZE, H. *Introduction to Information Retrieval*. Cambridge: Cambridge University Press, 2008. Citado 4 vezes nas páginas 20, 24, 25 e 26.

MIRTO, L. *Leveraging AI chatbots for engineering*. 2023. Zuken. Disponível em: <<https://www.zuken.com/us/blog/leveraging-ai-chatbots-for-engineering/>>. Acesso em: 20 jul. 2025. Citado 3 vezes nas páginas 14, 17 e 23.

NISKIER, J.; MACINTYRE, A. J. *Instalações Elétricas*. 6. ed. Rio de Janeiro: LTC, 2013. Citado na página 19.

PACW. *AI in Electric Power Systems Protection and Control*. 2024. PAC World Magazine. Disponível em: <<https://www.pacw.org/ai-in-electric-power-systems-protection-and-control>>. Acesso em: 10 set. 2025. Citado na página 23.

PIRELLI. *Manual Pirelli de Instalações Elétricas*. São Paulo: Pirelli, 2018. Citado na página 20.

RADFORD, A. et al. *Language Models are Unsupervised Multitask Learners*. 2019. OpenAI Blog, v. 1, n. 8, p. 9. Disponível em: <https://cdn.openai.com/better-language-models/language_models_are_unsupervised_multitask_learners.pdf>. Acesso em: 27 ago. 2025. Citado 2 vezes nas páginas 17 e 24.

ROBYLON.AI. *Rule-based vs AI Chatbots: 4 Hidden Trade-Offs Nobody Talks About*. 2026. Disponível em: <<https://www.robylon.ai/blog/rule-based-vs-ai-chatbots-2026>>. Acesso em: 10 jan. 2026. Citado 4 vezes nas páginas 15, 16, 17 e 18.

RUSSELL, S.; NORVIG, P. *Artificial Intelligence: A Modern Approach*. 4. ed. Upper Saddle River: Prentice Hall, 2020. Citado 3 vezes nas páginas 16, 23 e 24.

SELZY. *O que é chatbot e como funciona?* 2024. Disponível em: <<https://www.selzy.com/br/blog/o-que-e-chatbot-e-como-funciona/>>. Acesso em: 10 jun. 2025. Citado na página 14.

TURING, A. M. Computing machinery and intelligence. *Mind*, v. 59, n. 236, p. 433–460, 1950. Citado na página 24.

UNIVERSITY OF WARWICK. *Can AI chatbots pass the Fundamentals of Engineering (FE) and Principles and Practice of Engineering (PE) exams?* 2025. Disponível em: <https://warwick.ac.uk/fac/cross_fac/eduport/edufund/projects/yang/projects/can-ai-chatbots-pass-the-fundamentals-of-engineering-fe-and-principles-and-practice-of-3f716a>. Acesso em: 20 jun. 2025. Citado na página 17.

VASWANI, A. et al. Attention is all you need. *Advances in Neural Information Processing Systems*, v. 30, 2017. Citado na página 26.

APÊNDICE A – Código-fonte do agente RAG

Os códigos-fonte reproduzidos neste e nos apêndices seguintes correspondem ao sistema em funcionamento, disponível publicamente para acesso e testes em <<https://nbr5410-agent-dtdsdf8ecibhnnmbikfe.streamlit.app/>>.

O arquivo `agent.py` contém o núcleo inteligente do sistema. Nele são definidas as quatro ferramentas de cálculo elétrico (cálculo de corrente, seleção de condutor, queda de tensão e conversão BTU/h), o grafo de execução LangGraph com os nós de recuperação (*retrieve*), geração de resposta (*generate*) e resposta de escopo negativo (*not_found*), além da interface pública `ask()` utilizada pela camada *web* Streamlit.

Código A.1 – `agent.py` — Agente RAG com ferramentas de cálculo baseadas na NBR 5410

```

1  """
2  Agente RAG + Ferramentas de Calculo baseado na NBR 5410 usando LangGraph
3
4  Fluxo do grafo:
5      START -> retrieve -> [generate / not_found] -> END
6
7  - retrieve      : busca os trechos mais relevantes no vectorstore
8  - generate      : gera resposta com base nos trechos e pode acionar
9                    ferramentas
10                    de calculo (corrente, condutor, queda de tensao,
11                    conversao BTU)
12 - not_found     : retorna mensagem de "sem informacao" quando nenhum trecho
13                    passa pelo limiar de relevancia
14 """
15
16 from __future__ import annotations
17
18 import os
19 from typing import Annotated, List, TypedDict
20
21 from dotenv import load_dotenv
22 from langchain_chroma import Chroma
23 from langchain_core.documents import Document
24 from langchain_core.messages import AIMessage, HumanMessage,
25     SystemMessage, ToolMessage
26 from langchain_core.tools import tool
27 from langchain_google_genai import ChatGoogleGenerativeAI,
28     GoogleGenerativeAIEmbeddings
29 from langgraph.graph import END, START, StateGraph
30 from langgraph.graph.message import add_messages
31
32 load_dotenv()

```

```

29
30 VECTORSTORE_DIR = "./vectorstore"
31 RELEVANCE_THRESHOLD = 0.40
32 TOP_K = 8
33 TOP_K_GENERATE = 6
34
35 # --- Tabelas NBR 5410 (Tabela 36)
    -----
36
37 _CONDUCTOR_TABLE: dict[str, dict[float, int]] = {
38     "B1": {1.5: 13, 2.5: 18, 4.0: 24, 6.0: 31, 10: 42, 16: 56, 25: 73,
39           35: 89, 50: 108, 70: 136, 95: 164, 120: 188, 150: 213, 185:
40           246, 240: 278},
41     "B2": {1.5: 15, 2.5: 20, 4.0: 27, 6.0: 34, 10: 46, 16: 62, 25: 80,
42           35: 99, 50: 118, 70: 149, 95: 179, 120: 206, 150: 233, 185:
43           268, 240: 304},
44     "C": {1.5: 18, 2.5: 24, 4.0: 32, 6.0: 41, 10: 57, 16: 76, 25: 99,
45           35: 121, 50: 145, 70: 183, 95: 220, 120: 253, 150: 290, 185:
46           334, 240: 380},
47     "E": {1.5: 22, 2.5: 30, 4.0: 40, 6.0: 51, 10: 70, 16: 94, 25: 119,
48           35: 148, 50: 180, 70: 232, 95: 282, 120: 328, 150: 379, 185:
49           437, 240: 514},
50 }
51
52 _TEMP_CORRECTION: dict[int, float] = {
53     10: 1.22, 15: 1.17, 20: 1.12, 25: 1.06, 30: 1.00,
54     35: 0.94, 40: 0.87, 45: 0.79, 50: 0.71, 55: 0.61,
55 }
56
57 _GROUPING_CORRECTION: dict[str, float] = {
58     "1": 1.00, "2": 0.80, "3": 0.70, "4": 0.65, "5": 0.65,
59     "6": 0.57, "7": 0.57, "8": 0.52, "9": 0.52, "10": 0.47,
60 }
61
62 def _temp_factor(t: float) -> float:
63     temps = sorted(_TEMP_CORRECTION.keys())
64     closest = min(temps, key=lambda x: abs(x - t))
65     return _TEMP_CORRECTION[closest]
66
67 def _grouping_factor(n: int) -> float:
68     if n <= 0:
69         return 1.0
70     key = str(min(n, 10))
71     return _GROUPING_CORRECTION.get(key, 0.47)

```

```
71
72 # --- Ferramentas de Calculo
    -----
73
74 @tool
75 def calcular_corrente(
76     potencia_w: float,
77     tensao_v: float = 220.0,
78     fator_potencia: float = 0.92,
79 ) -> str:
80     """
81     Calcula a corrente eletrica de uma carga ( $I = P / V \times fp$ ).
82     Use sempre que precisar da corrente para dimensionar condutores ou
83     disjuntores.
84     tensao_v: tensao em Volts (127 ou 220 para monofasico; 220 ou 380
85     para trifasico).
86     fator_potencia: 1.0 para cargas resistivas (chuveiro, forno); 0.92
87     para ar-cond.; 0.85 para motores.
88     """
89     if tensao_v <= 0 or fator_potencia <= 0:
90         return "Erro: tensao e fator de potencia devem ser positivos."
91     corrente = potencia_w / (tensao_v * fator_potencia)
92     return (
93         f"Corrente de projeto: {corrente:.2f} A\n"
94         f" Potencia: {potencia_w:.0f} W\n"
95         f" Tensao: {tensao_v:.0f} V\n"
96         f" Fator de potencia: {fator_potencia}"
97     )
98
99 @tool
100 def btu_para_watts(
101     capacidade_btu_h: float,
102     eer: float = 9.0,
103 ) -> str:
104     """
105     Converte capacidade frigorifrica de ar-condicionado (BTU/h) para
106     potencia eletrica consumida (W).
107     IMPORTANTE: BTU/h e capacidade de resfriamento, NAO e consumo
108     eletrico diretamente.
109     eer: Energy Efficiency Ratio (BTU/Wh). Use 9 para calculo
110     conservador; splits inverter modernos: 11-14.
111     """
112     if eer <= 0:
113         return "Erro: EER deve ser positivo."
114     potencia_w = capacidade_btu_h / eer
115     return (
```

```

111         f"Potencia eletrica consumida: {potencia_w:.0f} W\n"
112         f" Capacidade frigorifica: {capacidade_btu_h:.0f} BTU/h\n"
113         f" EER utilizado: {eer} BTU/Wh\n"
114         f" Nota: use EER=9 para dimensionamento conservador"
115     )
116
117
118 @tool
119 def selecionar_condutor(
120     corrente_a: float,
121     metodo_instalacao: str = "B1",
122     temperatura_ambiente_c: float = 30.0,
123     num_circuitos_agrupados: int = 1,
124 ) -> str:
125     """
126     Seleciona a secao minima de condutor de cobre conforme NBR 5410
127     Tabela 36.
128     metodo_instalacao: B1=conduto embutido (mais comum), B2=conduto
129     sobreposto, C=cabo sobre parede, E=livre no ar.
130     temperatura_ambiente_c: temperatura do local de instalacao (tipico:
131     30-40C em tetos, 25-30C em paredes).
132     num_circuitos_agrupados: quantos circuitos passam no mesmo conduto (
133     afeta capacidade por agrupamento).
134     """
135
136     metodo = metodo_instalacao.upper().strip()
137     tabela = _CONDUCTOR_TABLE.get(metodo, _CONDUCTOR_TABLE["B1"])
138     if metodo not in _CONDUCTOR_TABLE:
139         metodo = "B1"
140
141     f_temp = _temp_factor(temperatura_ambiente_c)
142     f_group = _grouping_factor(num_circuitos_agrupados)
143     fator_total = f_temp * f_group
144     corrente_necessaria = corrente_a / fator_total
145
146     secao_selecionada = None
147     capacidade_tabelada = None
148     for secao, cap in sorted(tabela.items()):
149         if cap >= corrente_necessaria:
150             secao_selecionada = secao
151             capacidade_tabelada = cap
152             break
153
154     if secao_selecionada is None:
155         return (
156             f"Corrente {corrente_a:.1f} A (corrigida: {
157                 corrente_necessaria:.1f} A) "
158             f"excede as secoes tabeladas para metodo {metodo}. "

```

```
153         f"Consultar engenheiro eletrico para solucao (paralelo de
           condutores ou cabo especial)."
```

```
154     )
155
156     capacidade_real = capacidade_tabelada * fator_total
157
158     return (
159         f"Condutor selecionado: {secao_selecionada} mm2 (cobre, PVC 70C)
           \n"
160         f" Metodo de instalacao: {metodo}\n"
161         f" Capacidade tabelada (30C, 1 circuito): {capacidade_tabelada}
           A\n"
162         f" Fator temperatura ({temperatura_ambiente_c}C): {f_temp:.2f}\n"
163         f" Fator agrupamento ({num_circuitos_agrupados} circuito(s)): {
           f_group:.2f}\n"
164         f" Capacidade real apos correcoes: {capacidade_real:.1f} A\n"
165         f" Corrente de projeto: {corrente_a:.2f} A -- ATENDE"
166     )
167
168
169 @tool
170 def calcular_queda_tensao(
171     corrente_a: float,
172     comprimento_m: float,
173     secao_mm2: float,
174     tensao_v: float = 220.0,
175 ) -> str:
176     """
177     Calcula a queda de tensao em circuito monofasico de cobre.
178     NBR 5410: maximo recomendado 4% para circuitos terminais.
179     corrente_a: corrente de projeto em Amperes.
180     comprimento_m: comprimento do circuito em metros (distancia do QD ao
           equipamento).
181     secao_mm2: secao do condutor em mm2.
182     tensao_v: tensao nominal do circuito (127 ou 220 V).
183     """
184     if secao_mm2 <= 0 or comprimento_m <= 0:
185         return "Erro: secao e comprimento devem ser positivos."
186
187     rho_cobre = 0.0172 # ohm.mm2/m a 20C
188     queda_v = 2 * rho_cobre * comprimento_m * corrente_a / secao_mm2
189     queda_pct = (queda_v / tensao_v) * 100
190
191     if queda_pct <= 4.0:
192         status = "APROVADO (dentro do limite de 4% recomendado pela NBR
           5410)"
```

```
193     elif queda_pct <= 7.0:
194         status = "ATENCAO: acima de 4% -- aceitavel para circuitos de
                forca, mas aumentar secao e recomendado"
195     else:
196         status = "REPROVADO: acima de 7% -- aumentar secao do condutor
                obrigatoriamente"
197
198     return (
199         f"Queda de tensao: {queda_v:.2f} V ({queda_pct:.2f}%)\\n"
200         f"   Status: {status}\\n"
201         f"   Parametros: I={corrente_a}A, L={comprimento_m}m, S={
                secao_mm2}mm2, V={tensao_v}V\\n"
202         f"   Limite NBR 5410: 4% recomendado, 7% maximo absoluto"
203     )
204
205
206 _TOOLS = [calcular_corrente, btu_para_watts, selecionar_condutor,
            calcular_queda_tensao]
207 _TOOL_MAP = {t.name: t for t in _TOOLS}
208
209 # --- Embeddings / Vectorstore singletons
                -----
210
211 _embeddings: GoogleGenerativeAIEmbeddings | None = None
212
213
214 def _get_api_key() -> str:
215     key = os.environ.get("GOOGLE_API_KEY", "")
216     if not key:
217         try:
218             import streamlit as st
219             key = st.secrets.get("GOOGLE_API_KEY", "")
220         except Exception:
221             pass
222     return key
223
224
225 def _get_embeddings() -> GoogleGenerativeAIEmbeddings:
226     global _embeddings
227     if _embeddings is None:
228         _embeddings = GoogleGenerativeAIEmbeddings(
229             model="models/gemini-embedding-001",
230             google_api_key=_get_api_key(),
231         )
232     return _embeddings
233
234
```

```

235 _vectorstore: Chroma | None = None
236
237
238 def _get_vectorstore() -> Chroma:
239     global _vectorstore
240     if _vectorstore is None:
241         _vectorstore = Chroma(
242             persist_directory=VECTORSTORE_DIR,
243             embedding_function=_get_embeddings(),
244             collection_metadata={"hnsw:space": "cosine"},
245         )
246     return _vectorstore
247
248
249 # --- State
250 -----
251 class RAGState(TypedDict):
252     question: str
253     retrieved_docs: List[Document]
254     answer: str
255     history: List # mensagens anteriores da conversa
256
257
258 # --- Nodes
259 -----
260 def retrieve_node(state: RAGState) -> dict:
261     vs = _get_vectorstore()
262     docs = vs.similarity_search(state["question"], k=TOP_K)
263     return {"retrieved_docs": docs}
264
265
266 _SYSTEM = """\
267 Voce e um assistente tecnico especializado na ABNT NBR 5410 (Instalacoes
268     Eletricas \
269 de Baixa Tensao), com ferramentas de calculo para dimensionamento
270     eletrico.
271
272 REGRAS -- siga sem excecao:
273
274 1. RESPOSTA OBJETIVA PRIMEIRO: Comece sempre com o resultado direto \
275 (ex.: "Condutor: 4 mm^2 | Disjuntor: 20A curva C | DR: 30 mA"). \
276 Depois apresente o passo a passo e alertas.
277
278 2. USE AS FERRAMENTAS: Sempre calcule antes de responder quando houver \
279 dimensionamento (condutor, corrente, BTU, queda de tensao).

```

```
278
279 3. ASSUMA PADROES quando um parametro nao for informado -- nao paralise
    pedindo dados:
280     - Tensao: 220V (se nao dito)
281     - Fator de potencia: 1,0 para resistivos; 0,92 para AC; 0,85 para
        motores
282     - Metodo de instalacao: B1 (conduto embutido)
283     - Temperatura: 30C
284     - Circuitos agrupados: 1
285     Mencione brevemente quais padroes assumiu ao final da resposta.
286
287 4. UMA PERGUNTA DE CADA VEZ: Se precisar esclarecer algo, faca no maximo
    \
288 UMA pergunta por mensagem. Prefira assumir o padrao e avisar.
289
290 5. USE O HISTORICO: Nao peca informacoes que ja foram dadas na conversa
    anterior.
291
292 6. Nunca invente valores de tabelas -- use apenas dados calculados ou do
    contexto.
293
294 7. Se a pergunta nao for sobre instalacoes eletricas / NBR 5410, recuse
    gentilmente.\
295 """"
296
297
298 def generate_node(state: RAGState) -> dict:
299     docs = state["retrieved_docs"][:TOP_K_GENERATE]
300     context = "\n\n--\n\n".join(
301         f"[Fonte: {doc.metadata.get('source', 'NBR 5410')}] ",
302         f"[Pagina/Secao: {doc.metadata.get('page', doc.metadata.get('secao', '??'))}] \n"
303         f"{doc.page_content}"
304         for doc in docs
305     )
306
307     llm = ChatGoogleGenerativeAI(
308         model="gemini-2.5-flash-lite",
309         temperature=0,
310         google_api_key=_get_api_key(),
311     )
312     llm_with_tools = llm.bind_tools(_TOOLS)
313
314     messages: list = [SystemMessage(content=_SYSTEM)]
315
316     # Historico da conversa (max. 6 mensagens = 3 trocas)
317     for msg in state.get("history", [])[-6:]:
```

```

318         messages.append(msg)
319
320     messages.append(HumanMessage(
321         content=f"CONTEXTO NBR 5410:\n{context}\n\nPERGUNTA: {state['question']}"
322     ))
323
324     # Loop de tool-calling (maximo 6 rodadas)
325     for _ in range(6):
326         response = llm_with_tools.invoke(messages)
327         if not response.tool_calls:
328             break
329         messages.append(response)
330         for tc in response.tool_calls:
331             tool_fn = _TOOL_MAP.get(tc["name"])
332             if tool_fn:
333                 resultado = tool_fn.invoke(tc["args"])
334             else:
335                 resultado = f"Ferramenta '{tc['name']}' nao encontrada."
336             messages.append(
337                 ToolMessage(content=str(resultado), tool_call_id=tc["id"]
338                     ])
339
340         )
341
342     return {"answer": response.content}
343
344 def not_found_node(state: RAGState) -> dict:
345     return {
346         "answer": (
347             "Nao encontrei informacoes relevantes na base de conhecimento da "
348             "NBR 5410 para esta pergunta.\n\n"
349             "***Sugestoes:**\n"
350             "- Verifique se o tema e coberto pela NBR 5410\n"
351             "- Tente usar termos tecnicos da norma (ex.: SPDA, DPS, disjuntor, "
352             "aterramento, secao minima)\n"
353             "- Reformule a pergunta de forma mais especifica, incluindo tensao, "
354             "potencia ou tipo de carga"
355         )
356     }
357
358 # --- Routing

```

```

359
360 def _route(state: RAGState) -> str:
361     return "generate" if state["retrieved_docs"] else "not_found"
362
363
364 # --- RAG graph (usado pelo Streamlit)
365     -----
366
367 def _build_rag_graph():
368     g = StateGraph(RAGState)
369     g.add_node("retrieve", retrieve_node)
370     g.add_node("generate", generate_node)
371     g.add_node("not_found", not_found_node)
372     g.add_edge(START, "retrieve")
373     g.add_conditional_edges("retrieve", _route)
374     g.add_edge("generate", END)
375     g.add_edge("not_found", END)
376     return g.compile()
377
378 _rag_graph = _build_rag_graph()
379
380
381 def ask(question: str, history: list | None = None) -> dict:
382     """Interface publica para o Streamlit."""
383     result = _rag_graph.invoke(
384         {"question": question, "retrieved_docs": [], "answer": "", "
385             history": history or []}
386     )
387     return {"answer": result["answer"], "sources": result["
388         retrieved_docs"]}
389
390 # --- Chat graph (compatível com LangGraph Studio / LangSmith)
391     -----
392
393 class ChatState(TypedDict):
394     messages: Annotated[list, add_messages]
395
396
397 def _chat_node(state: ChatState) -> dict:
398     last_human = next(
399         (m for m in reversed(state["messages"]) if isinstance(m,
400             HumanMessage)),
401         None,
402     )
403     if last_human is None:

```

```
401         return {"messages": [AIMessage(content="Nenhuma pergunta
           recebida.")]}
402
403     result = ask(last_human.content)
404     return {"messages": [AIMessage(content=result["answer"])]}
405
406
407 def _build_chat_graph():
408     g = StateGraph(ChatState)
409     g.add_node("agent", _chat_node)
410     g.add_edge(START, "agent")
411     g.add_edge("agent", END)
412     return g.compile()
413
414
415 # Exposto para o LangGraph Studio via langgraph.json
416 graph = _build_chat_graph()
```

APÊNDICE B – Código-fonte do módulo de ingestão

O arquivo `ingest.py` é responsável pelo pré-processamento da base de conhecimento. Ele carrega o PDF da NBR 5410, divide o conteúdo em fragmentos (*chunks*) de até 800 caracteres com sobreposição de 150, integra a base suplementar `knowledge_extra.md` — dividida por cabeçalhos Markdown para preservar tabelas intactas — e gera os vetores semânticos (*embeddings*) via API do Google, persistindo-os no banco de dados vetorial ChromaDB. O script precisa ser executado uma única vez antes de iniciar a aplicação, ou sempre que a base de conhecimento for atualizada.

Código B.1 – `ingest.py` — Processamento e indexação da base de conhecimento

```

1  """
2  Processa o PDF da NBR 5410 + base de conhecimento suplementar e cria o
    vectorstore local (Chroma).
3  Execute sempre que atualizar o PDF ou o knowledge_extra.md.
4
5  Uso:
6      python ingest.py                                # usa ./data/NBR-5410.pdf +
    ./knowledge_extra.md
7      python ingest.py <caminho_para_o.pdf>           # PDF personalizado
8  """
9
10 import os
11 import sys
12 import shutil
13 import time
14
15 from dotenv import load_dotenv
16 from langchain_community.document_loaders import PyPDFLoader, TextLoader
17 from langchain_text_splitters import RecursiveCharacterTextSplitter,
    MarkdownHeaderTextSplitter
18 from langchain_google_genai import GoogleGenerativeAIEmbeddings
19 from langchain_chroma import Chroma
20
21 load_dotenv()
22
23 VECTORSTORE_DIR = "./vectorstore"
24 CHUNK_SIZE = 800
25 CHUNK_OVERLAP = 150
26 EXTRA_KNOWLEDGE = "./knowledge_extra.md"
27
28
29 def load_markdown_knowledge(md_path: str) -> list:
30     """Divide o markdown por cabeçalhos para preservar tabelas intactas.
    """

```

```
31     with open(md_path, "r", encoding="utf-8") as f:
32         content = f.read()
33
34     headers = [("#", "secao"), ("##", "subsecao"), ("###", "topico")]
35     md_splitter = MarkdownHeaderTextSplitter(
36         headers_to_split_on=headers, strip_headers=False
37     )
38     docs = md_splitter.split_text(content)
39
40     # Garantir que secoes muito longas sejam subdivididas
41     char_splitter = RecursiveCharacterTextSplitter(
42         chunk_size=1200, chunk_overlap=100
43     )
44     final_docs = []
45     for doc in docs:
46         if len(doc.page_content) > 1400:
47             sub = char_splitter.create_documents(
48                 [doc.page_content], metadatas=[doc.metadata]
49             )
50             final_docs.extend(sub)
51         else:
52             final_docs.append(doc)
53
54     for doc in final_docs:
55         doc.metadata.setdefault("source", md_path)
56         doc.metadata["type"] = "conhecimento_suplementar"
57
58     return final_docs
59
60
61 def ingest(pdf_path: str) -> None:
62     if not os.path.exists(pdf_path):
63         print(f"[ERRO] Arquivo nao encontrado: '{pdf_path}'")
64         print("Coloque o NBR-5410.pdf na pasta ./data/ ou informe o
65             caminho completo.")
66         sys.exit(1)
67
68     # Limpar vectorstore anterior
69     if os.path.exists(VECTORSTORE_DIR):
70         shutil.rmtree(VECTORSTORE_DIR)
71         print("[!] Vectorstore anterior removido para recriacao completa")
72
73     # 1) Carregar PDF
74     print(f"[1/5] Carregando PDF: {pdf_path}")
75     loader = PyPDFLoader(pdf_path)
76     pages = loader.load()
```

```
76     print(f"          {len(pages)} paginas carregadas")
77
78     # 2) Dividir PDF em chunks
79     print("[2/5] Dividindo PDF em chunks...")
80     splitter = RecursiveCharacterTextSplitter(
81         chunk_size=CHUNK_SIZE,
82         chunk_overlap=CHUNK_OVERLAP,
83         add_start_index=True,
84     )
85     pdf_chunks = splitter.split_documents(pages)
86     print(f"          {len(pdf_chunks)} chunks do PDF")
87
88     # 3) Carregar base de conhecimento suplementar
89     all_chunks = list(pdf_chunks)
90     if os.path.exists(EXTRA_KNOWLEDGE):
91         print(f"[3/5] Carregando base suplementar: {EXTRA_KNOWLEDGE}")
92         extra_chunks = load_markdown_knowledge(EXTRA_KNOWLEDGE)
93         all_chunks.extend(extra_chunks)
94         print(f"          {len(extra_chunks)} chunks do knowledge_extra.md")
95     else:
96         print(f"[3/5] knowledge_extra.md nao encontrado -- ignorando")
97
98     print(f"          Total de chunks: {len(all_chunks)}")
99
100    # 4) Carregar modelo de embeddings
101    print("[4/5] Configurando Google Generative AI Embeddings...")
102    embeddings = GoogleGenerativeAIEmbeddings(
103        model="models/gemini-embedding-001",
104        google_api_key=os.environ.get("GOOGLE_API_KEY"),
105    )
106
107    # 5) Criar vectorstore em batches para respeitar rate limit (100 req
        /min)
108    print("[5/5] Gerando embeddings e salvando vectorstore...")
109    BATCH_SIZE = 80
110    vectorstore = None
111    for i in range(0, len(all_chunks), BATCH_SIZE):
112        batch = all_chunks[i:i + BATCH_SIZE]
113        print(f"          Batch {i // BATCH_SIZE + 1}/{(len(all_chunks) - 1)
            // BATCH_SIZE + 1} ({len(batch)} chunks)...")
114        if vectorstore is None:
115            vectorstore = Chroma.from_documents(
116                documents=batch,
117                embedding=embeddings,
118                persist_directory=VECTORSTORE_DIR,
119                collection_metadata={"hnsw:space": "cosine"},
120            )
```

```
121         else:
122             vectorstore.add_documents(batch)
123             if i + BATCH_SIZE < len(all_chunks):
124                 time.sleep(65)
125
126         print(f"\n[OK] Vectorstore salvo em '{VECTORSTORE_DIR}/'")
127         print("[OK] Ingestion concluída! Agora execute: py -m streamlit run
128             app.py")
129
130 if __name__ == "__main__":
131     pdf = sys.argv[1] if len(sys.argv) > 1 else "./data/NBR-5410.pdf"
132     ingest(pdf)
```

APÊNDICE C – Base de conhecimento suplementar

O arquivo `knowledge_extra.md` é um documento em formato Markdown elaborado pelos próprios autores para complementar o PDF da NBR 5410. Ele organiza, em texto plano com seções curtas, as fórmulas de dimensionamento, as tabelas da Tabela 36 da norma (capacidade de condução por método de instalação), os fatores de correção por temperatura e agrupamento, os perfis de cargas residenciais típicas, o procedimento de dimensionamento em 10 etapas e três exemplos resolvidos. Essa estrutura plana garante que cada fragmento seja recuperado de forma autônoma pelo *pipeline* RAG, aumentando a precisão das respostas geradas pelo *chatbot*.

```

1 # Base de Conhecimento Suplementar - Instalações Elétricas Prediais e Residenciais
2
3 ## Fórmulas Fundamentais de Dimensionamento
4
5 ### Corrente de Projeto
6 Corrente elétrica:  $I = P / (V \times fp)$ 
7 Corrente trifásica:  $I = P / (\sqrt{3} \times V \times fp)$ 
8 Potência aparente:  $S = V \times I$  (VA)
9 Potência ativa:  $P = S \times fp$  (W)
10 Fator de potência:  $fp = \cos(\phi)$ 
11
12 Tensões nominais no Brasil: 127V ou 220V (monofásico); 220V ou 380V (trifásico)
13 Tensão padrão para dimensionamento residencial: 220V entre fases, 127V fase-neutro
14
15 ### Queda de Tensão em Circuitos de Cobre (monofásico)
16 Fórmula:  $\Delta V = 2 \times \rho \times L \times I / S$ 
17  $\rho$  cobre = 0,0172 Ohm.mm2/m (a 20grausC)
18  $\Delta V\% = (\Delta V / V_{nominal}) \times 100$ 
19
20 Limites NBR 5410 seção 9.4.2:
21 - Circuitos de iluminação: máximo 4% (recomendado <= 3%)
22 - Demais circuitos (tomadas, motores, ar-cond.): máximo 7% (recomendado <= 4%)
23 - Medido desde o ponto de entrega até o ponto mais distante
24
25 ### Queda de Tensão Trifásico
26  $\Delta V = \sqrt{3} \times \rho \times L \times I / S$ 
27
28 ## Conversão de Unidades Comuns
29
30 ### BTU para Watts Elétricos (Ar-Condicionado)
31 IMPORTANTE: BTU/h é capacidade frigorífica, NÃO é consumo elétrico.
32 Consumo elétrico = Capacidade BTU/h / EER (Energy Efficiency Ratio)
33 EER típico splits residenciais inverter: 11 a 14 BTU/Wh

```

34 EER típico splits convencionais: 8 a 10 BTU/Wh

35 Use EER = 9 para cálculo conservador (dimensionamento a favor da segurança)

36

37 ### HP para Watts (Motores)

38 1 HP = 745,7 W (mecânico)

39 1 CV = 736 W

40 Potência elétrica do motor = HP x 745,7 / rendimento

41 Rendimento típico motores: 85% a 93%

42

43 ### kVA para kW

44 $P(kW) = S(kVA) \times fp$

45 Exemplo: motor 5 kVA, $fp=0,80 \rightarrow P = 5 \times 0,80 = 4 \text{ kW}$

46

47 ## Tabela de Condutores de Cobre - NBR 5410 Tabela 36

48

49 ### Método B1 - Conduto embutido em parede ou teto, 3 condutores carregados, isolamento PVC 70grausC

50

51 Seção 1,5 mm² -> capacidade 13 A

52 Seção 2,5 mm² -> capacidade 18 A

53 Seção 4,0 mm² -> capacidade 24 A

54 Seção 6,0 mm² -> capacidade 31 A

55 Seção 10 mm² -> capacidade 42 A

56 Seção 16 mm² -> capacidade 56 A

57 Seção 25 mm² -> capacidade 73 A

58 Seção 35 mm² -> capacidade 89 A

59 Seção 50 mm² -> capacidade 108 A

60 Seção 70 mm² -> capacidade 136 A

61 Seção 95 mm² -> capacidade 164 A

62 Seção 120 mm² -> capacidade 188 A

63

64 Método B1 é o mais comum em instalações residenciais (eletrodutos embutidos em alvenaria).

65

66 ### Método B2 - Conduto sobreposto ou em parede, 3 condutores carregados, PVC 70grausC, 30grausC

67

68 Seção 1,5 mm² -> capacidade 15 A

69 Seção 2,5 mm² -> capacidade 20 A

70 Seção 4,0 mm² -> capacidade 27 A

71 Seção 6,0 mm² -> capacidade 34 A

72 Seção 10 mm² -> capacidade 46 A

73 Seção 16 mm² -> capacidade 62 A

74 Seção 25 mm² -> capacidade 80 A

75 Seção 35 mm² -> capacidade 99 A

76 Seção 50 mm² -> capacidade 118 A

77 Seção 70 mm² -> capacidade 149 A

78 Seção 95 mm² -> capacidade 179 A

79 Seção 120 mm² -> capacidade 206 A

80

81 ### Método C - Cabo fixado diretamente sobre parede ou teto, 3 condutores carregados, PVC 70grausC
82
83 Seção 1,5 mm² -> capacidade 18 A
84 Seção 2,5 mm² -> capacidade 24 A
85 Seção 4,0 mm² -> capacidade 32 A
86 Seção 6,0 mm² -> capacidade 41 A
87 Seção 10 mm² -> capacidade 57 A
88 Seção 16 mm² -> capacidade 76 A
89 Seção 25 mm² -> capacidade 99 A
90 Seção 35 mm² -> capacidade 121 A
91 Seção 50 mm² -> capacidade 145 A
92 Seção 70 mm² -> capacidade 183 A
93 Seção 95 mm² -> capacidade 220 A
94 Seção 120 mm² -> capacidade 253 A
95
96 ### Método E - Livre no ar (bandejamento aéreo), 3 condutores, PVC 70grausC, 30grausC
97
98 Seção 1,5 mm² -> capacidade 22 A
99 Seção 2,5 mm² -> capacidade 30 A
100 Seção 4,0 mm² -> capacidade 40 A
101 Seção 6,0 mm² -> capacidade 51 A
102 Seção 10 mm² -> capacidade 70 A
103 Seção 16 mm² -> capacidade 94 A
104 Seção 25 mm² -> capacidade 119 A
105 Seção 35 mm² -> capacidade 148 A
106 Seção 50 mm² -> capacidade 180 A
107 Seção 70 mm² -> capacidade 232 A
108 Seção 95 mm² -> capacidade 282 A
109 Seção 120 mm² -> capacidade 328 A
110
111 ### Para 2 condutores carregados (monofásico sem neutro como condutor de carga)
112 Multiplicar as capacidades acima por 1,15 (aproximadamente).
113
114 ## Fatores de Correção de Temperatura - Isolação PVC 70grausC
115
116 Temperatura 20grausC -> fator 1,12
117 Temperatura 25grausC -> fator 1,06
118 Temperatura 30grausC -> fator 1,00 (referência da tabela)
119 Temperatura 35grausC -> fator 0,94
120 Temperatura 40grausC -> fator 0,87
121 Temperatura 45grausC -> fator 0,79
122 Temperatura 50grausC -> fator 0,71
123 Temperatura 55grausC -> fator 0,61
124
125 Capacidade corrigida = Iz_tabela x fator_temperatura
126 Corrente admissível real = Iz_tabela x fator_temperatura x fator_agrupamento
127

128 ## Fatores de Correção por Agrupamento de Condutos ou Cabos
129
130 1 circuito -> fator 1,00
131 2 circuitos -> fator 0,80
132 3 circuitos -> fator 0,70
133 4 a 5 circuitos -> fator 0,65
134 6 a 7 circuitos -> fator 0,57
135 8 a 9 circuitos -> fator 0,52
136 10 a 11 circuitos -> fator 0,47
137 12 a 13 circuitos -> fator 0,45
138 14 a 16 circuitos -> fator 0,41
139 17 a 19 circuitos -> fator 0,38
140 20 ou mais -> fator 0,35
141
142 Aplicar quando circuitos embutidos no mesmo conduto ou cabos agrupados juntos.
143
144 ## Perfis de Cargas Típicas Residenciais
145
146 ### Ar-Condicionado Split Residencial
147 Fator de potência: 0,92
148 EER típico convencional: 9 BTU/Wh (conservador para dimensionamento)
149 EER típico inverter eficiente: 12 BTU/Wh
150 Tensão de alimentação: 220V (padrão para splits)
151 Disjuntor recomendado: curva C (pico de partida do compressor)
152 Proteção obrigatória: DR 30mA (NBR 5410 exige em circuitos de ar-condicionado)
153
154 Consumo elétrico típico por capacidade frigorífica (EER=9):
155 - 9.000 BTU/h -> potência ~1.000 W -> corrente ~4,9 A @ 220V
156 - 12.000 BTU/h -> potência ~1.333 W -> corrente ~6,6 A @ 220V
157 - 18.000 BTU/h -> potência ~2.000 W -> corrente ~9,9 A @ 220V
158 - 24.000 BTU/h -> potência ~2.667 W -> corrente ~13,2 A @ 220V
159 - 30.000 BTU/h -> potência ~3.333 W -> corrente ~16,5 A @ 220V
160 - 32.000 BTU/h -> potência ~3.556 W -> corrente ~17,6 A @ 220V
161 - 36.000 BTU/h -> potência ~4.000 W -> corrente ~19,8 A @ 220V
162 - 48.000 BTU/h -> potência ~5.333 W -> corrente ~26,4 A @ 220V
163 - 60.000 BTU/h -> potência ~6.667 W -> corrente ~33,0 A @ 220V
164
165 Seção de condutor recomendada por capacidade (método B1, 30grausC, EER=9):
166 - Até 18.000 BTU -> 2,5 mm² (Iz=18A, suficiente para até ~10A)
167 - 24.000 a 36.000 BTU -> 4,0 mm² (Iz=24A, suficiente para até ~20A)
168 - 48.000 BTU -> 6,0 mm² (Iz=31A)
169 - 60.000 BTU -> 10 mm² (Iz=42A)
170
171 ### Chuveiro Elétrico
172 Fator de potência: 1,00 (carga puramente resistiva)
173 Potência e correntes típicas:
174 - 5.500 W @ 220V -> corrente 25,0 A

175 - 6.800 W @ 220V -> corrente 30,9 A
176 - 7.500 W @ 220V -> corrente 34,1 A
177 - 5.500 W @ 127V -> corrente 43,3 A
178 - 6.800 W @ 127V -> corrente 53,5 A
179 Seção mínima NBR 5410: 4,0 mm² para chuveiro @ 220V; 6,0 mm² @ 127V
180 Disjuntor: curva B ou C, In imediatamente acima da corrente (32A ou 40A @ 220V)
181 DR 30mA obrigatório por NBR 5410 (local molhado)
182
183 ### Fogão e Forno Elétrico de Embutir
184 Fator de potência: 1,00 (resistivo)
185 Potência típica forno de embutir: 2.000 a 3.500 W
186 Corrente @ 220V: 9 a 16 A
187 Seção mínima: 2,5 mm² (até 18A); 4,0 mm² para fornos acima de 3.500 W
188 Disjuntor: curva B, 16A ou 20A
189
190 ### Máquina de Lavar Roupa
191 Fator de potência: 0,85
192 Potência típica: 1.000 a 2.500 W (excluindo aquecimento de água)
193 Corrente @ 220V: ~5 a 13 A
194 Seção mínima: 2,5 mm²
195 Disjuntor: 16A curva C (pico de partida do motor)
196
197 ### Motor Elétrico Monofásico (bomba, compressor)
198 Fator de potência: 0,75 a 0,85
199 Corrente de partida: 4 a 7 x corrente nominal
200 Corrente nominal = HP x 745,7 / (V x fp x rendimento)
201 Disjuntor: curva D ou relé térmico adequado
202
203 ### Iluminação LED
204 Fator de potência: 0,90 a 0,95
205 Corrente muito baixa - dimensionamento pelo limite de 15A por circuito
206 Seção mínima: 1,5 mm²
207
208 ### Tomadas de Uso Geral (TUG)
209 Carga estimada por tomada: 100 VA (NBR 5410)
210 Máximo: 6 pontos por circuito (padrão de projeto)
211 Seção mínima: 2,5 mm²
212 Disjuntor: 20A
213
214 ### Tomadas de Uso Específico (TUE)
215 Circuito exclusivo para um equipamento
216 Seção dimensionada pela corrente do equipamento
217 Exemplos: ar-condicionado, chuveiro, forno, lava-louça
218
219 ## Seções Mínimas por Tipo de Circuito (NBR 5410)
220
221 Iluminação residencial -> mínimo 1,5 mm² cobre

222 Tomadas de uso geral (TUG) -> mínimo 2,5 mm² cobre
223 Tomadas de uso específico até 25A -> mínimo 2,5 mm² cobre
224 TUE acima de 25A -> mínimo 4,0 mm² cobre
225 Chuveiro elétrico @ 220V -> mínimo 4,0 mm² cobre
226 Chuveiro elétrico @ 127V -> mínimo 6,0 mm² cobre
227 Ar-condicionado até 18.000 BTU -> mínimo 2,5 mm² cobre
228 Ar-condicionado 24.000 a 36.000 BTU -> mínimo 4,0 mm² cobre
229 Ar-condicionado 48.000 BTU -> mínimo 6,0 mm² cobre
230 Condutor de proteção (PE) -> igual ao fase até 16 mm²; metade acima de 35 mm²
231
232 ## Dimensionamento de Disjuntores e Proteções
233
234 ### Critério de Seleção do Disjuntor (NBR 5410 seção 9.4)
235 Condição: $I_B \leq I_n \leq I_z$
236 I_B = corrente de projeto (carga)
237 I_n = corrente nominal do disjuntor
238 I_z = capacidade de condução do condutor (após correções)
239
240 Curvas de atuação:
241 - Curva B (2 a 3x I_n): cargas resistivas, iluminação, aquecimento
242 - Curva C (5 a 10x I_n): uso geral, motores pequenos, ar-condicionado residencial
243 - Curva D (10 a 20x I_n): transformadores, grandes motores, altas correntes de partida
244
245 Valores nominais padronizados (A): 6, 10, 16, 20, 25, 32, 40, 50, 63, 80, 100
246
247 ### DR (Dispositivo Diferencial Residual)
248 30 mA: obrigatório em banheiros, áreas molhadas, chuveiros, piscinas, ar-condicionado
249 300 mA: proteção contra incêndio em circuitos de tomadas gerais
250 Deve ser instalado após o disjuntor
251
252 ### DPS (Dispositivo de Proteção contra Surtos)
253 Obrigatório em instalações novas (NBR 5410 revisão 2020)
254 Tipo 2: instalado no quadro de distribuição
255 Corrente nominal de descarga: mínimo 5 kA
256
257 ## Procedimento Padrão de Dimensionamento de Circuito
258
259 Passo 1: Identificar a carga - potência (W), tensão (V), fator de potência, regime de operação
260 Passo 2: Calcular corrente de projeto: $I = P / (V \times fp)$
261 Passo 3: Aplicar fator de demanda se houver múltiplas cargas no mesmo circuito
262 Passo 4: Definir método de instalação (B1 = embutido, B2 = sobreposto, C = cabo sobre parede, E =
263 Passo 5: Definir temperatura ambiente e número de circuitos agrupados
264 Passo 6: Calcular corrente corrigida = $I / (\text{fator_temperatura} \times \text{fator_agrupamento})$
265 Passo 7: Selecionar seção do condutor pela Tabela 36: $I_z \geq$ corrente corrigida
266 Passo 8: Verificar queda de tensão para o comprimento do circuito (máx. 4% recomendado)
267 Passo 9: Selecionar disjuntor: $I_B \leq I_n \leq I_z$ (valores padronizados)
268 Passo 10: Verificar necessidade de DR 30mA e DPS

269

270 ## Exemplo Resolvido: Ar-Condicionado 32.000 BTU

271

272 Pergunta: Qual o condutor e disjuntor para um split de 32.000 BTU alimentado a 220V?

273

274 Resolução passo a passo:

275 1. Capacidade: 32.000 BTU/h

276 2. Consumo elétrico (EER=9 conservador): $P = 32.000 / 9 = 3.556 \text{ W}$

277 3. Tensão: 220V, $\text{fp} = 0,92$

278 4. Corrente de projeto: $I = 3.556 / (220 \times 0,92) = 17,6 \text{ A}$

279 5. Método B1 (embutido em alvenaria), temperatura 30grausC, 1 circuito

280 6. Fator temperatura = 1,00; fator agrupamento = 1,00

281 7. Corrente corrigida = 17,6 A

282 8. Tabela 36 método B1: 2,5 mm² suporta 18 A -> margem muito pequena

283 9. Selecionar 4,0 mm² (suporta 24 A) - folga adequada e para queda de tensão

284 10. Queda de tensão @ 20 m: $\Delta V = 2 \times 0,0172 \times 20 \times 17,6 / 4,0 = 3,0 \text{ V} = 1,4\% - \text{OK}$

285 11. Disjuntor: 20 A curva C ($I_B=17,6\text{A} \leq 20\text{A} \leq I_Z=24\text{A}$)

286 12. DR 30 mA obrigatório por NBR 5410

287

288 Resposta: Condutor 4,0 mm² cobre, disjuntor 20A curva C, DR 30mA.

289

290 ## Exemplo Resolvido: Chuveiro Elétrico 7.500W @ 220V

291

292 1. Potência: 7.500 W, $\text{fp} = 1,00$ (resistivo)

293 2. Corrente: $I = 7.500 / (220 \times 1,00) = 34,1 \text{ A}$

294 3. Método B1, 30grausC

295 4. Tabela 36 B1: 10 mm² suporta 42 A -> OK

296 5. Seção mínima NBR 5410 para chuveiro @ 220V: 4,0 mm²

297 6. Usar 6,0 mm² ou 10 mm² dependendo do comprimento

298 7. Disjuntor: 40 A curva B ($34,1\text{A} \leq 40\text{A} \leq 42\text{A}$ com 10mm²)

299 8. DR 30 mA obrigatório

300

301 ## Exemplo Resolvido: Quadro de Distribuição Residencial (100 m²)

302

303 Cargas típicas e dimensionamento:

304 - C1 Iluminação: 1.500 W / 220V / $\text{fp}=0,95$ -> $I=7,2\text{A}$ -> 1,5 mm² -> DJ 10A curva B

305 - C2 Tomadas sala/quartos: 1.200 VA -> $I=5,5\text{A}$ -> 2,5 mm² -> DJ 16A curva C

306 - C3 Tomadas sala/quartos: 1.200 VA -> $I=5,5\text{A}$ -> 2,5 mm² -> DJ 16A curva C

307 - C4 Tomadas cozinha/área: 1.200 VA -> $I=5,5\text{A}$ -> 2,5 mm² -> DJ 16A curva C

308 - C5 Chuveiro: 6.800W / 220V -> $I=30,9\text{A}$ -> 6,0 mm² -> DJ 32A curva B

309 - C6 Ar-condicionado 12.000 BTU: 1.333W / 220V / $\text{fp}=0,92$ -> $I=6,6\text{A}$ -> 2,5 mm² -> DJ 10A curva C

310 - C7 Ar-condicionado 18.000 BTU: 2.000W / 220V / $\text{fp}=0,92$ -> $I=9,9\text{A}$ -> 2,5 mm² -> DJ 16A curva C

311 - C8 Forno embutir: 3.000W / 220V -> $I=13,6\text{A}$ -> 2,5 mm² -> DJ 16A curva B

312

313 Disjuntor geral: soma das correntes com fator de demanda ($\sim 0,7$) -> DJ 63A ou 80A

314

315 ## Carga Mínima por Cômodo (NBR 5410 referência para projeto)

316

317 Sala de estar até 6 m²: 1 TUG por 5 m², mínimo 1

318 Sala de estar acima de 6 m²: 1 TUG por 5 m²

319 Dormitório até 6 m²: 1 TUG por 5 m², mínimo 1

320 Cozinha: mínimo 2 TUG (1 exclusivo por equipamento fixo)

321 Banheiro: 1 TUG por lavatório (fora da zona de risco)

322 Área de serviço: 1 TUG para máquina de lavar + 1 TUG geral

323 Garagem: 1 TUG

324 Quartos e salas: 1 ponto de iluminação por ambiente

325

326 ## Aterramento e Proteção (NBR 5410 seção 5)

327

328 Sistema TN-S (recomendado): neutro e PE separados em toda a instalação

329 Sistema TT: terra independente para a instalação

330 Resistência de aterramento: máximo 10 Ohm (para DR de 30mA funcionar corretamente)

331 Condutor PE: deve acompanhar todos os circuitos

332 PE não deve ser interrompido por dispositivos de proteção
