CENTRO UNIVERSITÁRIO SANTO AGOSTINHO - UNIFSA
CURSO DE ENGENHARIA DE SOFTWARE – 4º PERÍODO
DISCIPLINA: PROJETO DE BANCO DE DADOS
PROFESSOR: ANDERSON SOARES

TRABALHO FINAL – BANCO DE DADOS VETORIAL

Classificador Inteligente de Similaridade Semântica para o Setor Imobiliário

ALUNO: ANTONIO NEVES AGUIAR NETO

Teresina - PI
24.11.2025

PROBLEMA ESCOLHIDO

O tema escolhido foi “Classificador inteligente por similaridade semântica”, fazendo a elaboração de um problema cotidiano do mercado, com o objetivo de dar maior complexidade e contexto ao projeto.

Problema: Empresas do setor imobiliário recebem diariamente diversas mensagens de clientes solicitando informações, fazendo reclamações, pedindo visitas ou enviando documentos. Esse processo é manual e lento, pois alguém precisa ler cada mensagem e encaminhar ao setor responsável.

Solução: *Classificar automaticamente mensagens de clientes, identificando o setor responsável rapidamente, usando Inteligência Artificial baseada em similaridade semântica. As categorias utilizadas para a classificação das mensagens foram: Interesse, Documentação, Sugestão, Dúvida e Reclamação.*

TECNOLOGIAS USADAS

1. **OpenAI (Embeddings)**
   Utilizada por meio da API para transformar cada texto em um vetor numérico, chamado de Embeddings.

2. **Supabase + pgvector (Banco de dados vetorial)**
   Utilizado para armazenamento dos exemplos de mensagens para comparação, coluna vetorial, Função para busca semântica e Postgres com o pgvector.

3. **N8N (Automatização)**
   Utilizado para automatizar todo o processo de: Receber mensagens (webhook), geração de embeddings, consulta ao banco vetorial, formatação das respostas e retorno para a interface.

4. **Lovable (Front-End)**
   Utilizado no front-end para simular um site de uma imobiliária onde o cliente envia uma mensagem, recebe o retorno em segundos de qual o departamento responsável entrará em contato.

O QUE SÃO EMBEDDINGS

De modo geral, embeddings são uma forma de transformar textos em números para que o computador consiga entender o significado ao invés de só palavras, são objetos do mundo real, frases, imagens, vídeos, mas de uma forma que os computadores podem processar. Essa representação, que é fundamental para IA, permite por exemplo, pesquisas de similaridade para medir o quão parecidas duas ou mais mensagens são, mesmo que elas usem palavras diferentes.

Dessa forma é possível o sistema saber, por exemplo, que:

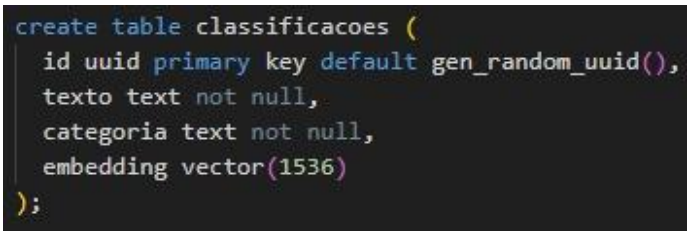
- “Gostaria de ver o apartamento segunda-feira” = **Interesse**
- “Preciso da segunda via dos documentos” = **Documentação**

“O corretor não compareceu ao horário combinado!” = **Reclamação**

- “Qual horário de atendimento no sábado?” = **Dúvida**
- **“**Seria bom poder acompanhar as atualizações pelo app/site.” = **Sugestão**

BANCO VETORIAL ESCOLHIDO E JUSTIFICATIVA

O banco vetorial escolhido foi o Supabase Postgres com a extensão pgvector, que permite armazenar e consultar vetores gerados por embeddings de forma eficiente.



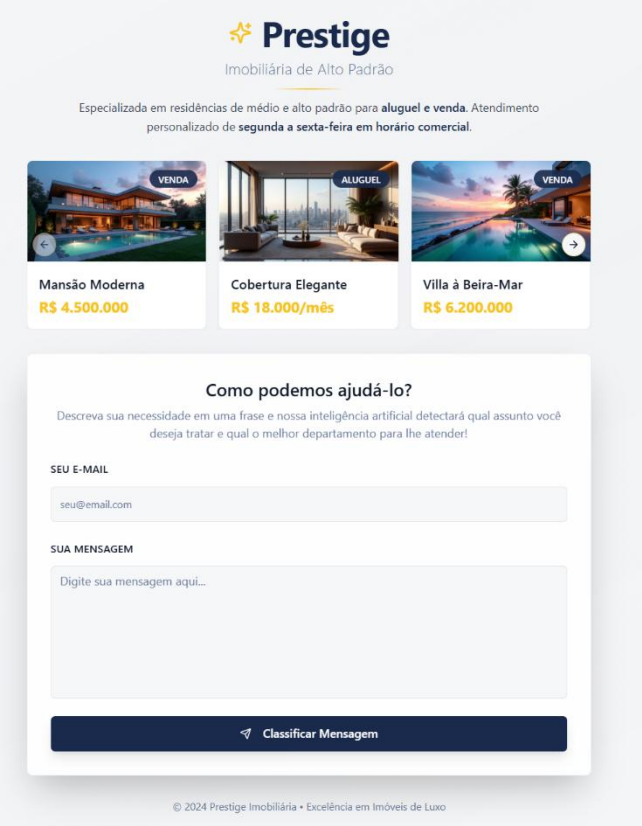
```
create table classificacoes (
  id uuid primary key default gen_random_uuid(),
  texto text not null,
  categoria text not null,
  embedding vector(1536)
);
```

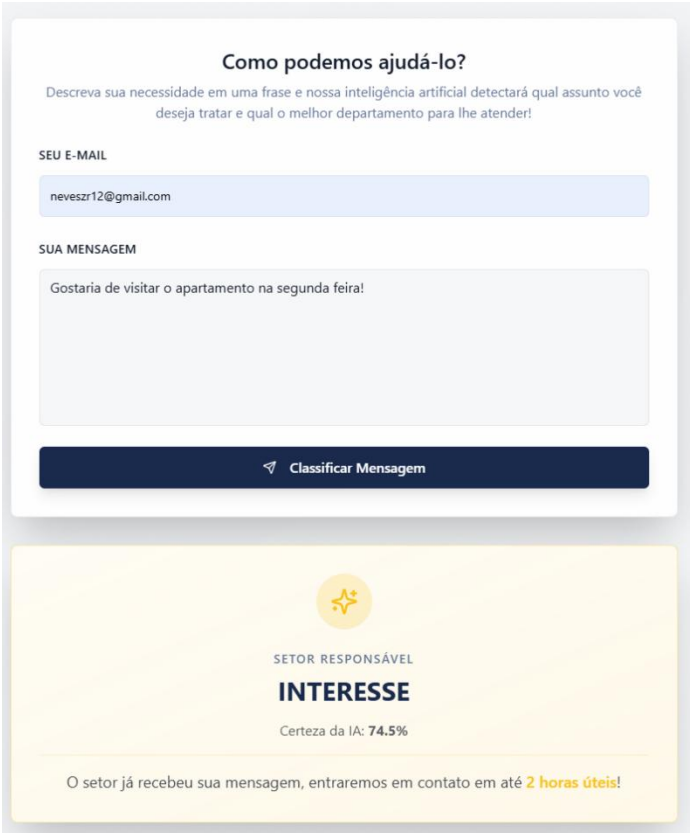
- Alta performance e cálculos por similaridade, pois o pgvector é otimizado para cálculos de similaridade;
- Facilidade para integrar com o N8N, com nodes prontos;
- Integração nativa com o Postgres;
- Permite criação de funções RPC de busca semântica diretamente no banco.

ARQUITETURA DA SOLUÇÃO

A) Interface (Front-End feito com Lovable)

O que o usuário vê, permite que a pessoa envie uma mensagem livre, com o assunto escolhido e receba como resposta o setor responsável pela demanda que entrará em contato, além da % de confiança dessa classificação.





B) Banco de dados vetorial (Supabase + Postgres SQL + Pgvector)

O Supabase funciona como repositório do sistema, onde as frases definidas previamente para comparação estão.

Foi utilizada uma tabela vetorial no projeto, ela é responsável por armazenar os exemplos usados para comparação semântica.

Tabela: Classificacoes

Coluna	Tipo	Descrição
id	uuid	Identificador único
categoria	text	Setor responsável
exemplo	text	Mensagem exemplo
embedding	vector(1536)	Vetor Pgvector gerado pela OpenAI

C) Automação do N8N (Pipeline de Classificação)

Pipeline principal, é responsável por receber a mensagem, gerar embedding, consultar o banco vetorial, comparar e devolver o resultado.

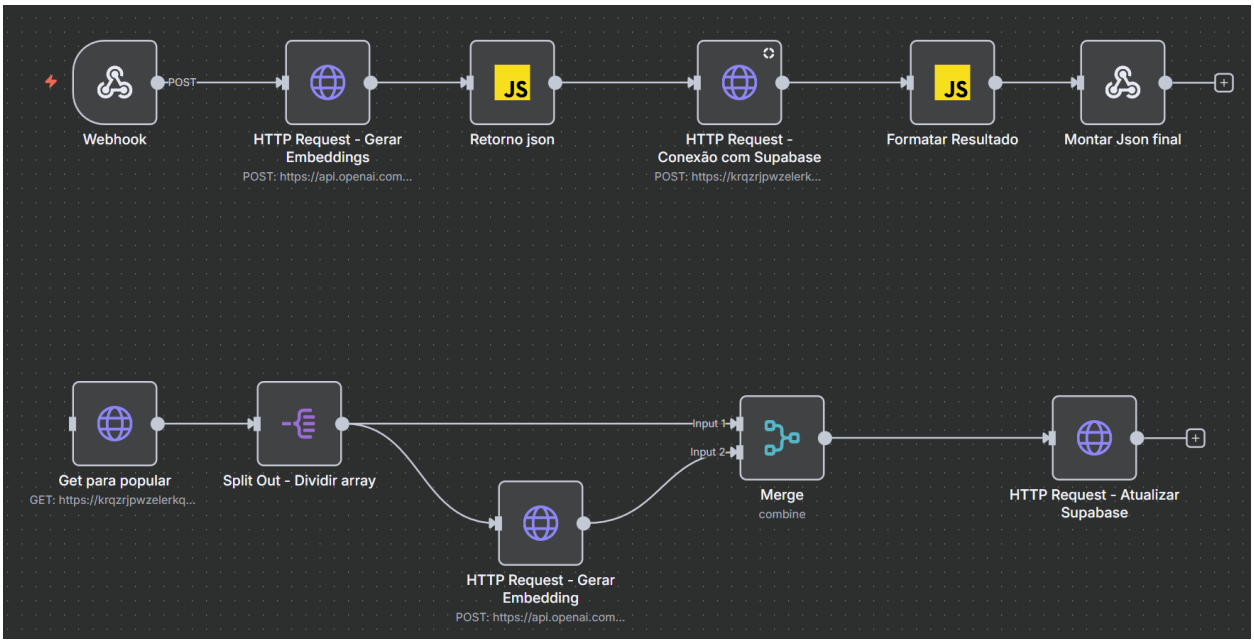
1) Recebimento da mensagem por um Webhook, conectado com o front-end feito no Lovable;
2) Geração do Embedding, feito pela API da OpenAI utilizando o modelo: *text-embedding-3-small;*
3) Depois que o embedding é gerado, a próxima etapa é a consulta ao banco vetorial, feito por outro HTTP Request, aqui o N8N envia uma chamada RPC ao Supabase, interligado na função *match_classificações;*
4) Após isso, se tem Nodes de código para formatar o resultado e depois um Node Respond to Webhook, responsável por devolver o resultado gerado para o front-end.

D) Automação do N8N (Pipeline de Alimentação)

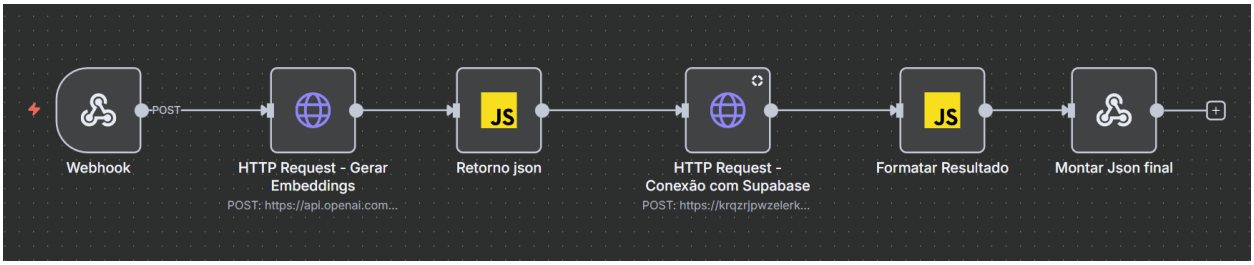
Essa pipeline é mais simples, completamente interna de uso para o administrador, serve apenas para gerar embeddings das novas frases que forem adicionadas ao banco, alimentando o sistema.

1) Um node de Get para recolher as frases do Supabase que ainda não possuem Embeddings;
2) Um Node Split Out, para dividir array e corrigir problemas de formatação;
3) Um Http Request para gerar o embedding, feito pela API da OpenAI, utilizando o mesmo modelo *text-embedding-3-small;*
4) Um Node Merge, para juntar novamente os dados que haviam sido separados pelo Split Out;
5) Um HTTP Request, com método Patch para atualizar o banco de dados no Supabase, inserindo os Embeddings em suas respectivas frases.

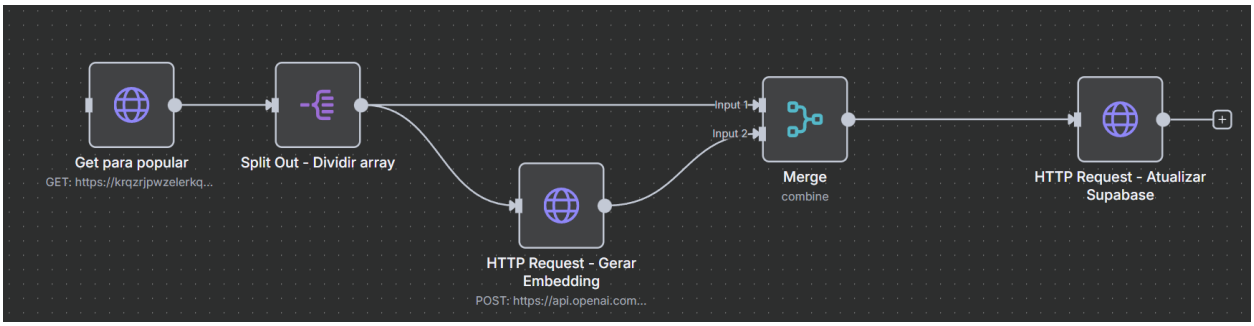
PRINTS DO WORKFLOW



Print do Workflow Completo.



Pipeline 01 – Fluxo principal do workflow, responsável por receber a mensagem, comparar e devolver resultado.

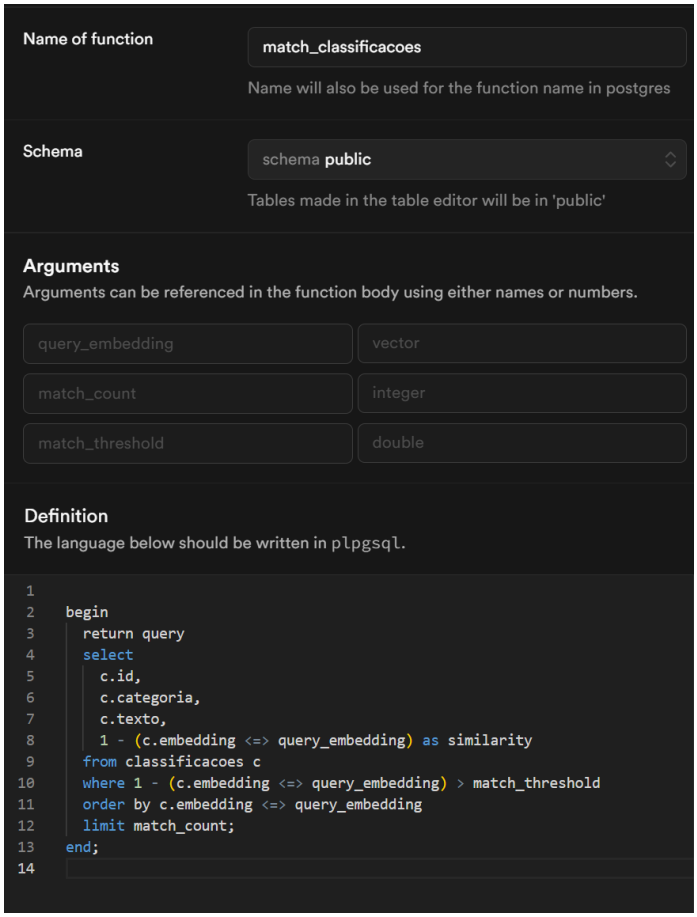


*Pipeline 02 – Responsável por gerar embeddings quando **são** adicionadas novas frases diretamente no banco de dados. Essa pipeline não precisa ser ativada junto a principal!*

EXPLICAÇÃO DA BUSCA SEMÂNTICA

A busca semântica é realizada pela função match_classificacoes no Supabase, que recebe os vetores da mensagem, compara usando operador de distância e ordena do mais semelhante ao menos semelhante. As categorias utilizadas para a classificação das mensagens foram: Interesse, Documentação, Sugestão, Dúvida e Reclamação.

Segue o print da função utilizada no projeto, sendo conectada no n8n a partir da URL https://krqzrjpwzelerkqxuxee.supabase.co/rest/v1/rpc/match_classificacoes



OBS: O Threshold (valor mínimo de similaridade exigido) foi colocado diretamente no n8n para facilitar alterações e testes, sendo usado 0.60, e o match_count (usado para pegar N mensagens semelhantes) está configurado como 5, também definido no n8n.

RESULTADOS E CONCLUSÕES

O funcionamento proposto foi validado, uma pipeline funciona como alimentação, para gerar embeddings de exemplos recém adicionados ao banco de dados, enquanto a outra pipeline é usada pelo usuário final, ela é responsável por receber uma mensagem real do usuário, gerar o embeddings dessa mensagem, consultar o banco vetorial pela função RPC (match_classificações), aplicar o Threshold com o valor de corte escolhido e retornar à categoria classificada.

Testes foram bem-sucedidos, frases como “Quero visitar o apartamento na segunda-feira”, “Preciso da documentação” e “Corretor não compareceu no horário marcado” foram classificadas corretamente em suas respectivas categorias.

LINK DO VÍDEO EXPLICATIVO DO PROJETO

https://youtu.be/cV4V33-oE_c