

TRABALHO FINAL – BANCO DE DADOS VETORIAL (3ª Avaliação – 10 Pontos)

Professor: Anderson Soares

Disciplina: Projeto de Banco de Dados

Tema: *Desenvolvimento de um sistema inteligente usando Banco de Dados Vetorial*

Ferramenta sugerida: n8n

Entrega: Código / workflow + Relatório + Vídeo demonstrativo

Data: 03/12/2015

1. Objetivo do Trabalho

O aluno ou a dupla deve **desenvolver um sistema completo** que utilize:

✓ **Banco de Dados Vetorial**

(Pode ser Supabase, Postgresql, Pinecone, Qdrant, ChromaDB, Milvus ou pgvector – à escolha é livre)

✓ **Embeddings para representar textos ou dados**

(OpenAI, HuggingFace ou outro modelo)

✓ **Fluxo automatizado ou aplicação usando n8n**

(recomendado pela facilidade e lógica visual)

O foco é aplicar **embeddings + similaridade + banco vetorial + pipeline inteligente** para resolver um problema real.

2. O que deve ser entregue

O trabalho poderá ser realizado **individualmente ou em dupla**.

A escolha é livre, porém a **complexidade e a qualidade da entrega devem ser compatíveis com o esforço esperado para o valor da avaliação (10 pontos)**

O aluno deverá **publicar todo o projeto em um repositório no GitHub** e entregar o link do repositório no AVA.

O repositório deve conter:

1. Workflow do n8n

- Arquivo exportado (.json)
- Deve mostrar todo o fluxo da solução (gatilho, embeddings, vector DB, consulta etc.)
- Salvar na pasta: [/workflow/](#)

2. Código adicional, se houver

- Scripts, front-end simples, automações externas
- Salvar na pasta: [/src/](#)

3. Relatório técnico (mínimo 4 páginas)

- Pode ser em PDF ou Markdown
- Salvar na pasta: [/docs/](#)

O relatório deve conter:

- Problema escolhido
- Tecnologias utilizadas
- O que são embeddings
- Banco vetorial escolhido e justificativa
- Arquitetura da solução

- Prints do workflow funcionando
- Explicação da busca semântica
- Resultados e conclusões

4. Vídeo demonstrativo (2–5 minutos)

- Pode ser enviado no YouTube ou outra plataforma
- Colocar o link no README do GitHub
- Alternativamente, pode ser um arquivo MP4 dentro da pasta [/video/](#) (opcional; vídeo no YouTube é preferível)

5. README do GitHub (obrigatório)

O README deve conter:

- Nome do projeto
- Descrição do problema
- Tecnologias usadas
- Link do vídeo
- Como executar
- Prints ou GIFs do funcionamento

3. Critérios de Avaliação (10 pontos)

Critério	Peso
Funcionamento técnico da solução	4,0
Uso correto do banco de dados vetorial + embeddings	2,0
Qualidade do workflow n8n / automação	2,0
Relatório técnico	1,5
Vídeo demonstrativo	0,5

4. Exemplos de Sistemas que o Aluno Pode Desenvolver

Abaixo, uma lista curada de ideias altamente viáveis, modernas e alinhadas ao mercado.

A) Chatbot Inteligente com RAG (Retrieval Augmented Generation)

Sistema que responde perguntas usando um conjunto de textos que o próprio aluno escolhe.

Exemplos:

- Chatbot sobre leis brasileiras
- Chatbot sobre políticas da universidade
- Chatbot sobre um livro ou artigo
- Chatbot sobre um assunto da disciplina
- Chatbot para responder dúvidas sobre editais

Workflow mínimo:

- Trigger (webhook)
 - Embedding da pergunta
 - Consulta ao banco vetorial
 - Chamada ao modelo LLM com contexto
 - Resposta final
-

B) Sistema de Recomendação Semântico (Recomendador Vetorial)

Sugere itens com base na similaridade semântica, não apenas palavras exatas.

Exemplos:

- Recomendação de filmes pelo gosto do usuário
- Recomendação de receitas (por ingredientes)
- Recomendação de cursos
- Recomendação de vagas de emprego
- Recomendação de livros por descrição

Workflow:

- Recebe descrição do usuário
- Gera embedding
- Busca itens similares no vector DB
- Retorna as recomendações

C) Classificador Inteligente por Similaridade Semântica

O aluno cria classes (ex.: "elogio", "reclamação", "informação", "técnica")

Indexa exemplos no banco vetorial

O n8n identifica automaticamente o tipo da mensagem.

Exemplos:

- Classificar e-mails
- Classificar dúvidas dos alunos
- Classificar posts de redes sociais
- Classificar tickets de suporte

D) Indexador Automático de PDFs com Busca Semântica

O aluno escolhe um PDF (ou vários), e cria um sistema que:

1. Extrai o texto
2. Fragmenta em chunks
3. Gera embeddings
4. Armazena no banco vetorial
5. Permite busca semântica via n8n

Exemplos:

- Artigos científicos
 - Apostilas
 - Regulamentos
 - Livro-texto
-

E) Sistema para Área da Saúde (muito útil e inovador)

Exemplos:

- Busca inteligente de sintomas
 - Chatbot sobre doenças específicas
 - Análise semântica de exames (descrição textual)
 - Recomendação de dietas por similaridade (excelente para nutrição)
-

F) Sistema Empresarial Simples

Exemplos:

- Busca semântica em políticas internas de TI
 - FAQ automático para funcionários
 - Classificação semântica de documentos internos
-

5. Ferramentas recomendadas

Banco de dados vetorial:

- **Qdrant** (fácil de usar, gratuito)
- **ChromaDB** (simples e local)
- **Pinecone** (cloud, muito profissional)
- **pgvector** (para quem quiser usar PostgreSQL)

Geração de embeddings:

- **OpenAI**
- **HuggingFace**
- **Google Vertex AI embeddings**
- **Cohere embeddings**

Automatização:

- **n8n (recomendado)**
Pode ser usado:
- local
- cloud gratuita (n8n.cloud trial)
- Docker

6. Fluxo mínimo exigido (obrigatório)

Todo projeto deve possuir pelo menos estes elementos:

1. Entrada (trigger, webhook, formulário, etc.)
 2. Geração de embedding
 3. Consulta ao banco vetorial
 4. Processamento da resposta (pode ser LLM)
 5. Saída (JSON, mensagem, painel, arquivo, e-mail etc.)
-

7. Sugestões de temas prontos para escolher

- Chatbot de dúvidas sobre Python
 - Recomendador de receitas fitness
 - Analisador de mensagens dos alunos para detectar tipo de dúvida
 - Recomendador de filmes baseado na sinopse
 - Chatbot do histórico de eventos da cidade
 - Chatbot sobre Banco de Dados (tema da disciplina!)
 - Buscador inteligente em um PDF de resumo de prova
 - Sistema para ajudar estudantes a escolher disciplinas eletivas
 - Recomendador de músicas pelo humor do usuário
 - Busca semântica em leis municipais
 - Chatbot que responde perguntas sobre “Arquitetura de Computadores”
 - Buscador de jurisprudência (para quem quiser focar em Direito)
 - Escolha é livre.
-

8. Conclusão

Este trabalho deverá demonstrar:

- Compreensão profunda de **embeddings**
- Aplicação prática de **banco de dados vetorial**
- Capacidade de criar **pipelines inteligentes** no **n8n**
- Entendimento de **similaridade semântica**
- Capacidade de integrar várias tecnologias em uma solução funcional