Busquedas RAG en Delphi con MakerAI

Las búsquedas por defecto utilizan un método muy sencillo que es un vector y un índice básico con un quick sort para realizar las búsquedas, es útil en pequeñas cantidades de nodos, la ventaja es que todo es en memoria.

Es posible también crear nuevos índices de búsqueda heredando de la clase ***TAIEmbeddingInde***x y asignándolo al ***TDataVec*** de la siguiente manera

TMiNuevoIndice = Class(TAiEmbeddingIndex) …

Var

DataVec := TAiDataVec.Create(Self);

DataVec.RagIndex := TMiNuevoIndice.Create;

DataVec.BuildIndex;

La otra opción es una base de datos de vectores, en el mercado hay muchas, en nuestro ejemplo utilizamos Postgres con la extensión pgVector.

<https://github.com/pgvector/pgvector>

La tabla que utilizamos en el demo es la siguiente

CREATE TABLE IF NOT EXISTS public.ragdata

(

id integer NOT NULL DEFAULT nextval('ragdata\_id\_seq'::regclass),

categoria character varying(100) COLLATE pg\_catalog."default",

texto text COLLATE pg\_catalog."default",

path character varying(1000) COLLATE pg\_catalog."default",

tags character varying(1000) COLLATE pg\_catalog."default",

embedding vector(1024),

secuencia integer DEFAULT 0,

CONSTRAINT ragdata\_pkey PRIMARY KEY (id)

)

TABLESPACE pg\_default;

ALTER TABLE IF EXISTS public.ragdata

OWNER to usrcimadb;

-- Index: ragdata\_embedding\_idx

-- DROP INDEX IF EXISTS public.ragdata\_embedding\_idx;

CREATE INDEX IF NOT EXISTS ragdata\_embedding\_idx

ON public.ragdata USING hnsw

(embedding vector\_cosine\_ops)

TABLESPACE pg\_default;

**Embeddings:** Lamentablemente Groq ni Antrhopic ofrecen servicio de embeddings, por lo tanto las alternativas serían OpenAi y Ollama local, se recomienda el modelo ***snowflake-arctic-embed*** en ollama.