DATASTAX

# Concorra a kits da DataStax

**A melhor solução para Vector Search e RAG**

DATASTAX

# Workshop: Criando Apps de IA Generativa com NodeJS e NextJS

**Samuel Matioli**
Solution Engineer at DataStax

# Agenda

- Aplicações com IA Generativa
- Vector Search & DataStax Astra
- O que é RAG?
- Ex1: Implementando RAG *na mão*
- Ex2: Simplificando com LangchainJS
- Ex3: Usando Functions

# Ganhe um Badge


Envie o seu DB ID e um screenshot da aplicação:

samuel.matioli@datastax.com

para receber seu Badge!

DATASTAX DEVELOPERS

# Sorteio de brindes

Cadastre-se aqui para participar do sorteio.

**bit.ly/datastax-tdcsc**

# Housekeeping – tdclover / #tdclover

Versões: NodeJS 18.

Clonar: **https://github.com/smatiolids/workshop-genai-js**

Criar conta em:

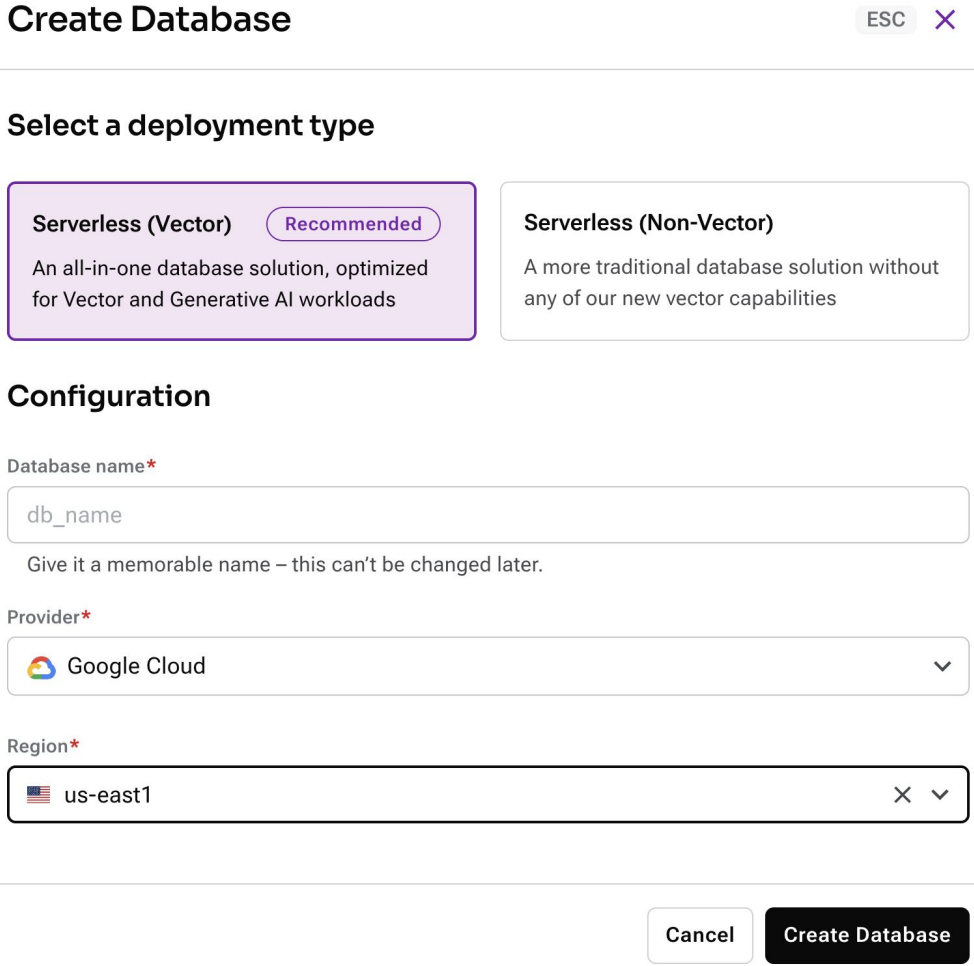
**OpenAI** (https://platform.openai.com/)

- Vamos usar os embeddings e o LLM para gerar conteúdo

**DataStax Astra** (astra.datastax.com) – **Conta GRATUITA!**

- Armazenar dados proprietários e memória

# Criando o Vector DB



# Como criar aplicações GenAI

**Language Model (LLM)**

OpenAI

Gemini

ANTHROP\C

cohere

**Código**

Frameworks

LangChain

Linguagens

node

NEXT.JS

**Dados**

- Vector Store and Search
- Contextual Information
- Agent Memory
- Cache

**Deployment**

**Monitoring**

# ❯ Dificuldades com o RAG em produção

## O apps com GenAI "quebram" muito

- Muitas dependências
- APIs externas **sem garantias**
- Novas versões de APIs externas **sem aviso**

## Escala e resiliência:

- Dados em tempo real
- Milhares de usuários
- Milhões de interações
- Muitos dados...

## Necessidade de evolução:

- Agilidade para incrementar e refinar os casos de uso
- Aproveitar novos modelos
- Novas origens de dados
- Ferramentas para a área de negócio

# Exercício 1

# › Iniciando

Passos:

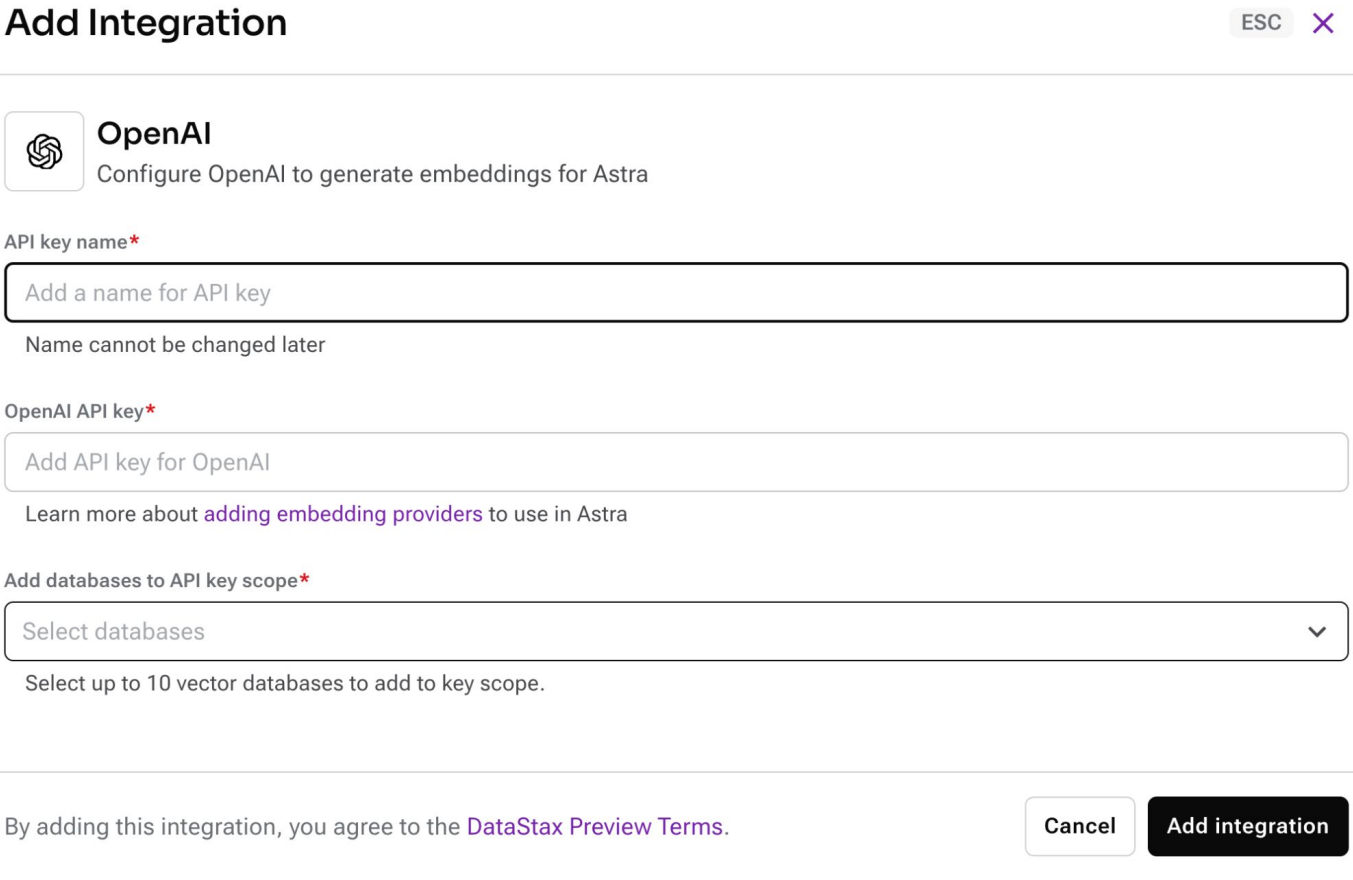
- Conta no Astra
- Clonar: **https://github.com/smatiolids/workshop-genai-js**
- Copiar o .env.example para .env.local
- Atualizar as variáveis
  - OPENAI_API_KEY
  - ASTRA_DB_API_ENDPOINT
  - ASTRA_DB_APPLICATION_TOKEN

# ❯ Ex 1: RAG + Vectorize

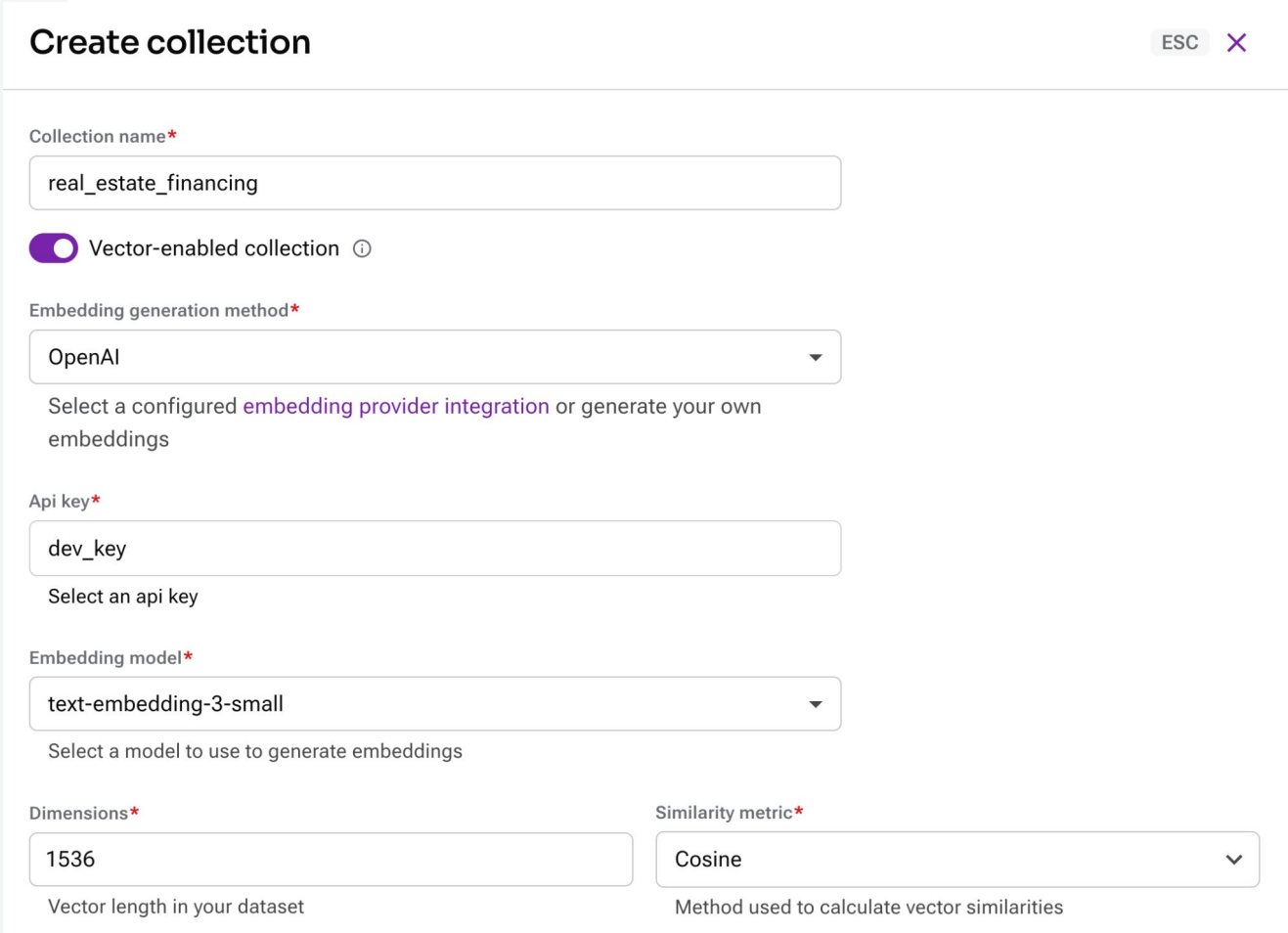
Passos:

- Acessar localhost:3000/v1/chat e fazer perguntas sobre financiamento imobiliario
- Configurar o $vectorize
- Criar collection "**real_estate_financing**" no Astra
- Carregar documento em localhost:3000/v1/upload

# API Key



# Criando a collection

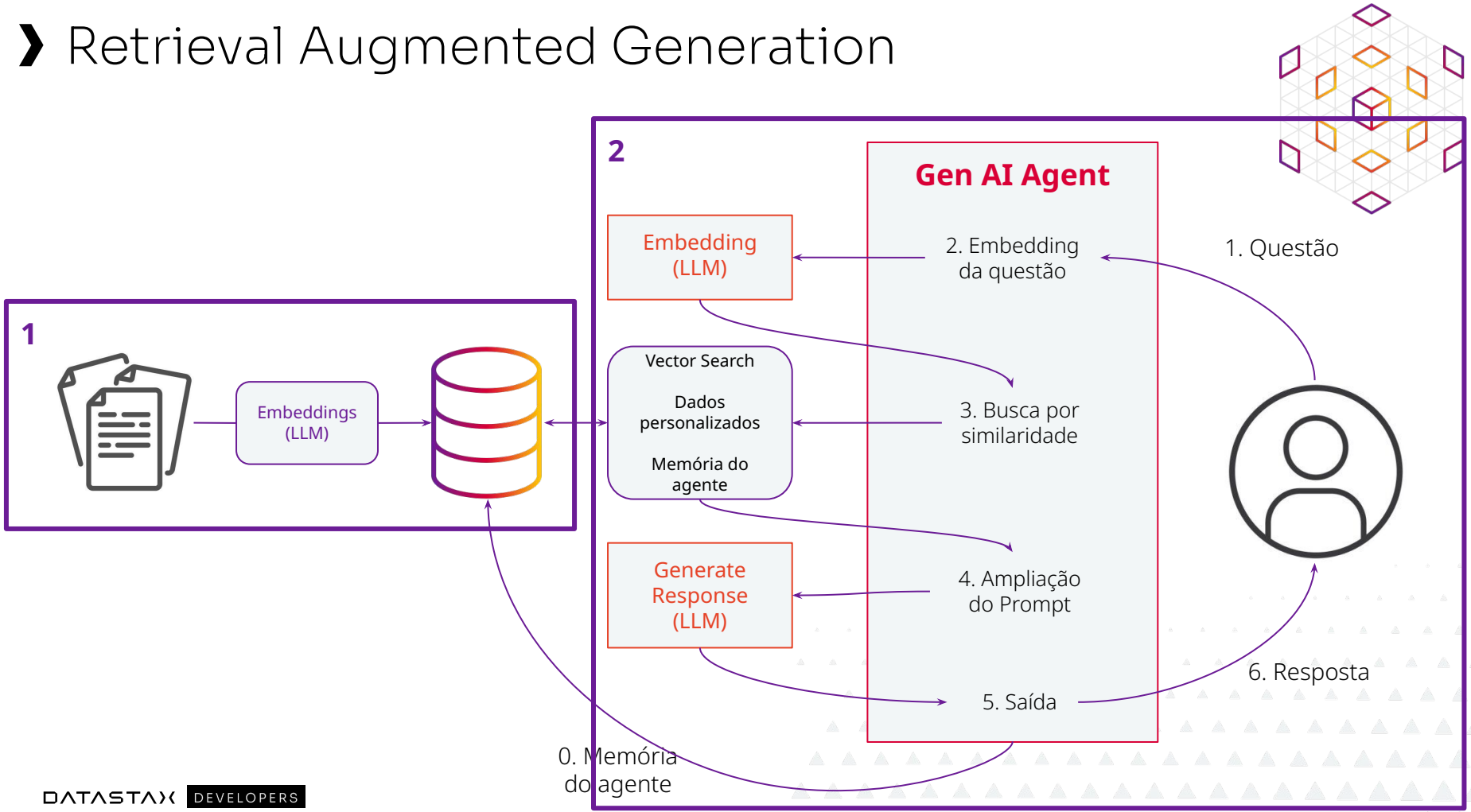


# ❯ Carregar dados para o Astra

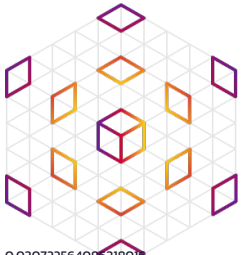
- Criar API Key
  - Vincular DB ao escopo da chave
- Criar collection "**real_estate_financing**"
- Vincular modelo OpenAI "**text-embedding-3-small**"
  - Dimensões: **1536**
  - Similarity Metric: **Cosine**

# Retrieval Augmented Generation (ou RAG)

# Retrieval Augmented Generation



# O que são embeddings



To create a security token that can be used to log into a database, select Token Management from the User Management menu. Then, choose an appropriate role for the user, and click the Generate Token button. Copy the token details to a safe place, as the secret that is shown can never be reproduced in the Astra console for security reasons.

**text-embedding-ada-002**

[-0.029334254562854767, 0.06338247656822205, 0.03711941838264465, 0.06770425289869308, 0.030722564086318016,
-0.03855780512094498, 0.05715630576014519, 0.011225797235965729, 0.03209076076745987, 0.018565965816378593,
0.005725057329982519, 0.003278332995250821, 0.019661232829093933, 0.008483093231916428, 0.01011530589312315,
-0.006859812211990036, -0.02742725796997547, 0.004272086545825005, 0.006464742589741945, 0.03338147327038864,
-0.06456394493579865, -0.00168662762735039, -0.02538292482495308, -0.013529211282730103, 0.006745325401425362,
-0.09090574085712433, -0.004533097147941589, -0.011557350866496563, -0.017933078110218048, -0.013565625995397568,
0.0373402051626763596, -0.013345368206501007, -0.046318963170051575, 0.018221553415060043, -0.030514687299728394,
0.06087908893823624, -0.015947293490171432, -0.004384475760161877, 0.011510275304317474, 0.03181201219558716,
0.0004961538943462074, -0.009607694111764431, -0.0026698557194322348, -0.021202115342020099, -0.02445143461227417,
0.018808122724294662, -0.04526928439736366, -0.03507508337497711, 0.011936023831367493, -0.04246317967772484,
-0.0453871265053749, 0.0030571885872632265, -0.026456169784606906, 0.004339783918112516, -0.004989228677004576,
-0.0019529943820089102, -0.015389597043395042, -0.008066490292549133, -0.043611098080873490, 0.018591511994600296,
-0.008249715901911259, 0.031450431793928146, 0.008753931149840355, -0.06284607946872711, -0.02690013311803341,
0.06175339594483755, 0.0314679853618145, 0.005365008022636175, -0.034285105764865875, -0.06475269049406052,
0.06229010224342346, -0.01609119959155867, -0.038237784057855606, -0.01697474904358387, 0.002332095988048067,
-0.02873411774635315, -0.07216104120016098, 0.04663623124361038, 0.023897146806120872, -0.02821142040193081,
-0.037146959453821118, -0.055613692849874496, -0.0028377221897244453, -0.06574894487857819, -0.061038188118572235,
0.06294400244951248, 0.003413078840821981 4, 0.07920042425394058, 0.007338271476328373, 0.06506536900997162,
-0.025222681462764 74, 0.027450041845440865, -0.017200432717800 14, 0.046272337436676025, -0.05018896237015724,
0.015779439359903336, -0.026586400344967842, -0.019746011123061 18, -0.000366890570148825 65, -0.016816521063447,
-0.025464840233325958, 0.000710080785211 1757, 0.04524853080511093, 0.001050888327881 6938, -0.005472411867231131,
0.011604292318224907, -0.042707648128271 1, -0.020046440884477094, -0.06824997067451477, -0.08084388822317123,
-0.018167271316051483, 0.038480401039123535, -0.041494846343994 14, 0.062140572816133 5, 0.01636849343776703,
-0.02775057591497898, 0.024102320894598 96, 0.021344885230064392, 0.056428126990795135, 0.02979239635169506,
-0.05207456275820732, 0.004299748223274946, 0.034176126122474 67, 0.034210722893476486, 0.000108426531369332 22,
0.011242502368986607, 0.037193566560745 24, -0.004098605364561081, 0.013202376663684845, 0.021659305319190025,
0.038507014513015 75, -0.039795678108930 59, 0.024909289553761482, 0.003612052416489236, 0.030269717797636986,
0.035327754914760 59, 0.040484450757503 51, -0.021236592903733253, 0.058955520391464 23, 0.049137584865095 23,
-0.047305576503276825, 0.052723329514265 06, 0.012154217809438705, -0.025136532261967 66, -0.010558293201029 3,
-0.049685653299093246, 0.032950107008218765, -0.007436738815158606, -0.07494320720434189, -0.044711060822010 04,
0.038164049386978 15, -0.029877835884690285, -0.020543526858091354, 0.025327796116471 29, 0.011234065517783165,
0.073742501437664 03, 0.042883597314357 76, 0.034353174269199 37, -0.029512003064155 58, -0.093858875334262 85,
-0.005317367613315582, 0.017055159434676 17, -0.009346966631710 53, 0.012932358309626 58, 0.021080965176222 47,
0.030062183737754 822, 0.004270109347999 096, -0.005795920733362436, 0.006119553931057 453, -0.009726069867610 931,
-0.001605408848263 3233, -0.128231540322303 77, 0.005963715258985 758, -0.016070995479822 16,...]

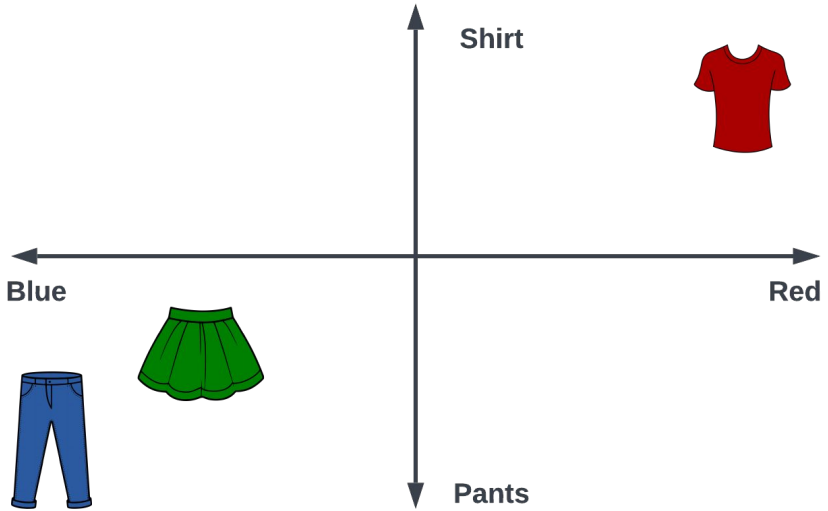
Texto

Embedding

1536 dimensões

# Como funciona o Vector Search?

- Documentos, imagens e outros conteúdos são transformados em **embeddings** (vetores com N dimensões) que representam seu significado.
- Busca semântica é realizada a partir da **similaridade** de de suas representações numéricas (os embeddings)
- As métricas de similaridade mais comuns são: Dot Product, Coseno e Euclidiana.



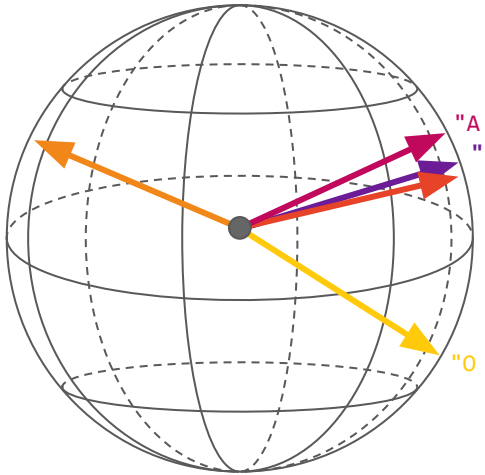

https://www.datastax.com/guides/what-is-a-vector-embedding

# Vector embeddings

**Representação fiel de "coisas" como vetores em um espaço de alta dimensionalidade**

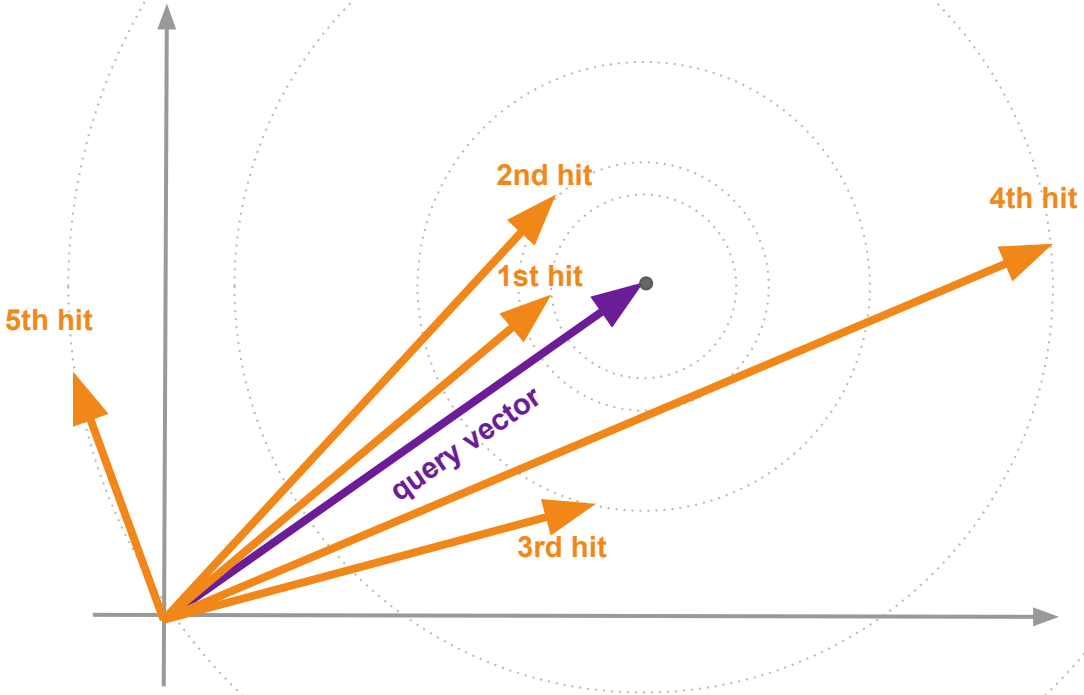
- Quais coisas? Principalmente **texto** (também: imagens, vídeos, sons ...)
- Transformar um problema semântico em um problema geométrico

"Aquele que busca a igualdade entre os desiguais ..."

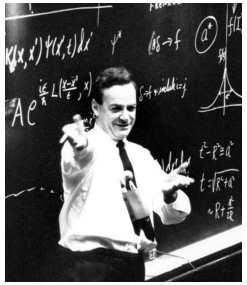


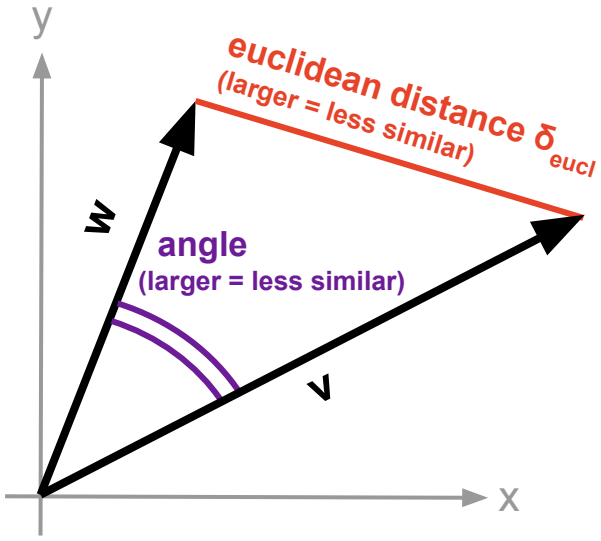

Embedding vector space

# Vector Search em uma imagem



# Similaridades





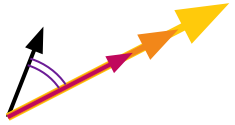

**Euclidean** similarity

$$S_{\text{eucl}}(\mathbf{v},\mathbf{w}) = \frac{1}{1+\sum_i (v_i - w_i)^2} = \frac{1}{1+\delta^2_{\text{eucl}}(\mathbf{v},\mathbf{w})}$$

$$\delta_{\text{eucl}}(\mathbf{v},\mathbf{w}) = \sqrt{\sum_i (v_i - w_i)^2}$$

**Cosine** similarity (a.k.a. "*only the angle counts*")

$$S_{\cos}(\mathbf{v},\mathbf{w}) = \frac{1+\frac{\sum_i v_i w_i}{|\mathbf{v}||\mathbf{w}|}}{2} = \frac{1+\frac{\mathbf{v}\cdot\mathbf{w}}{|\mathbf{v}||\mathbf{w}|}}{2}$$



DATASTAX ASTRA

# Data Platform all in one:

NoSQL, Vector Search, APIs, Streaming, ZeroOps, Multi Cloud, etc...


Applications



## Data sources

Web

Applications

IoT

DATASTAX

Starlight

JMS, Kafka, RabbitMQ

gRPC, REST, GraphQL, Document (JSON) and CQL APIs

STARGATE

PULSAR

DATASTAX ASTRA STREAMING

Streaming service based on Apache Pulsar®

Apache CASSANDRA

DATASTAX ASTRA DB

Database as a service based on Apache Cassandra®

CDC

Invisible operations

Infrastructure, Security, Scalability, High Availability, Resiliency

Multi-Cloud & Cloud-Native

Multi-region, Inter-cloud, Serverless and K8S microservices based

## Data ecosystem

Analytics

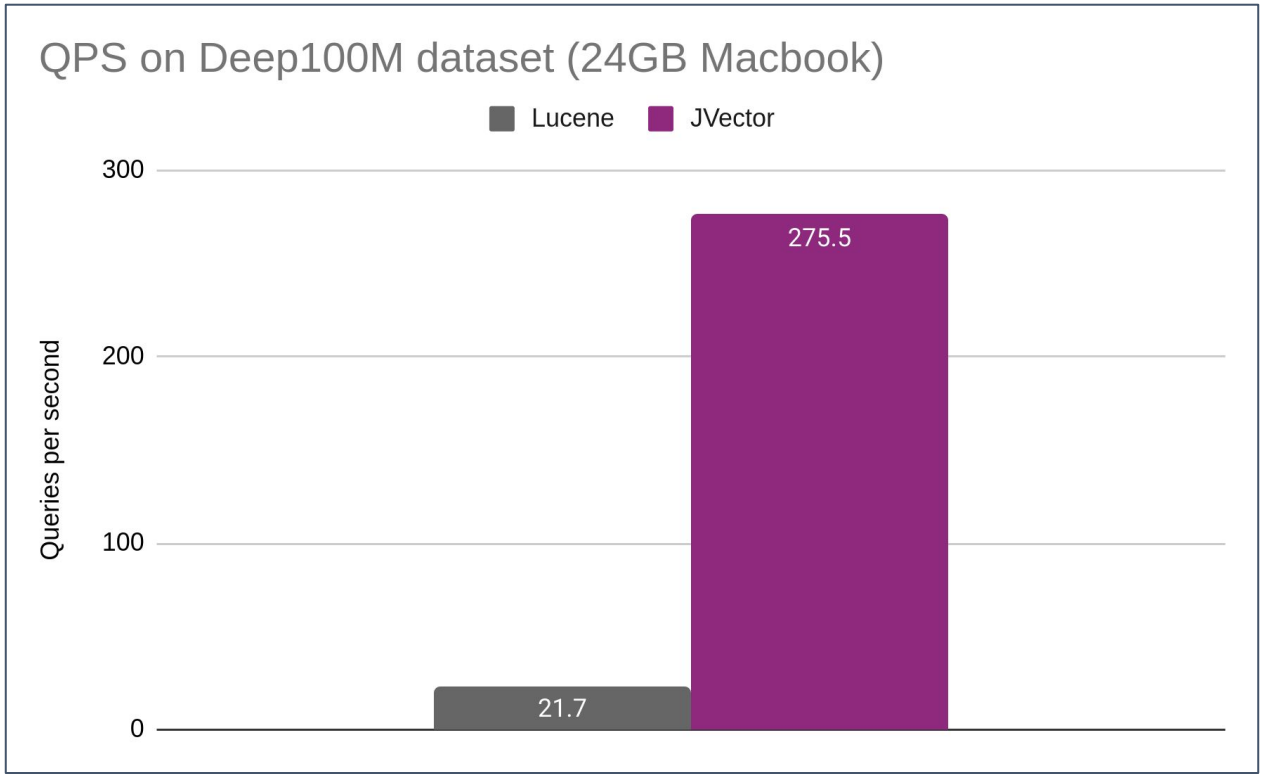
Search

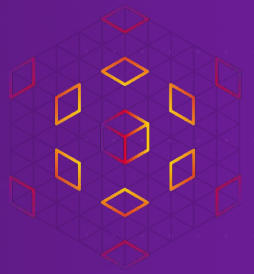
AI

aws

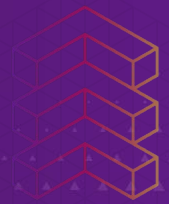

kubernetes

# JVector performance: read throughput



# Ex2 - RAG +  LangchainJS

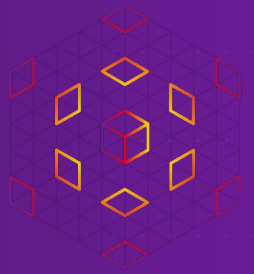




# ❯ Ex 2: Simplificando com LangchainJS

Passos:

- Acessar localhost:3000/**v2**/chat e fazer perguntas sobre financiamento imobiliario
- Criar collection "**real_estate_financing_langchain**" no Astra
- Carregar documento em localhost:3000/**v2**/upload
- Repetir perguntas em v2/chat




# Ex3 - Stocks

# ❯ Ex 3: GenAI + DataAPI + NextJS

Passos:

- Acessar localhost:3000/**stocks**
- Criar collection "**stocks**" no Astra
- Carregar dados para a collection
- Perguntar sobre preço de ação: AAPL, GOOG, SBUX

# Ganhe um Badge


Envie o seu DB ID e um screenshot da aplicação:

samuel.matioli@datastax.com

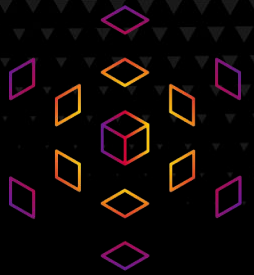
para receber seu Badge!

DATASTAX DEVELOPERS

# Sorteio de brindes

Cadastre-se aqui para participar do sorteio.

**bit.ly/datastax-tdcsc**




## Samuel Matioli

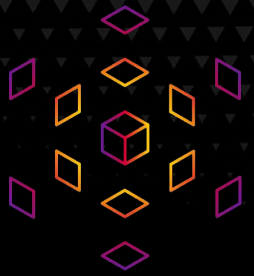
**Solution Engineer @ DataStax**

samuel.matioli@datastax.com

linkedin.com/in/samuelmatioli

samuelmatioli.medium.com

youtube.com/@samuelmatioli



# Appendix

# Referências adicionais

AstraDB no Langchain
https://python.langchain.com/docs/integrations/vectorstores/astradb

RAG na prática
https://www.youtube.com/watch?v=wc22tG5IGUU

AstraDB Vector & Vertex AI
https://www.youtube.com/watch?v=E0Wv4ALVp5w