

Busca 3.0

A Era dos Agentes

Alex Salgado Developer Advocate @ Elastic



PYTHON BRASIL 2025



Alex Salgado

Senior Developer Advocate
LATAM



- **Mestre** em Ciência da Computação pela UFF (Games)
- **MBA** UFF
- **PhD Candidate** UFF: Robótica/Visão Computacional
- + **25 anos** de experiência na área de desenvolvimento de software
- Ocupei diversos cargos, trabalhando em **startups**, pequenas e grandes empresas como Oracle, CSN, BRQ/IBM, **Chemtech/Siemens (9 anos)**.
- **8 anos** como professor universitário



Quer saber como funciona
uma busca 3:0 ?



Three solutions powered by one stack

3 solutions



Enterprise Search



Observability



Security

Powered by
the Elastic Stack

Kibana

Elasticsearch

Agent

Beats

Logstash

Deployed
anywhere



Elastic Cloud



Elastic Cloud
Enterprise



Elastic Cloud
on Kubernetes

Saas

Orchestration

The Power of Elasticsearch + LLMs

Imagine asking complex questions about your data in English, Portuguese or Chinese using natural language and taking time to reason

1 Natural Language (MCP)

At what time of day did I take the most steps exploring Las Vegas yesterday?

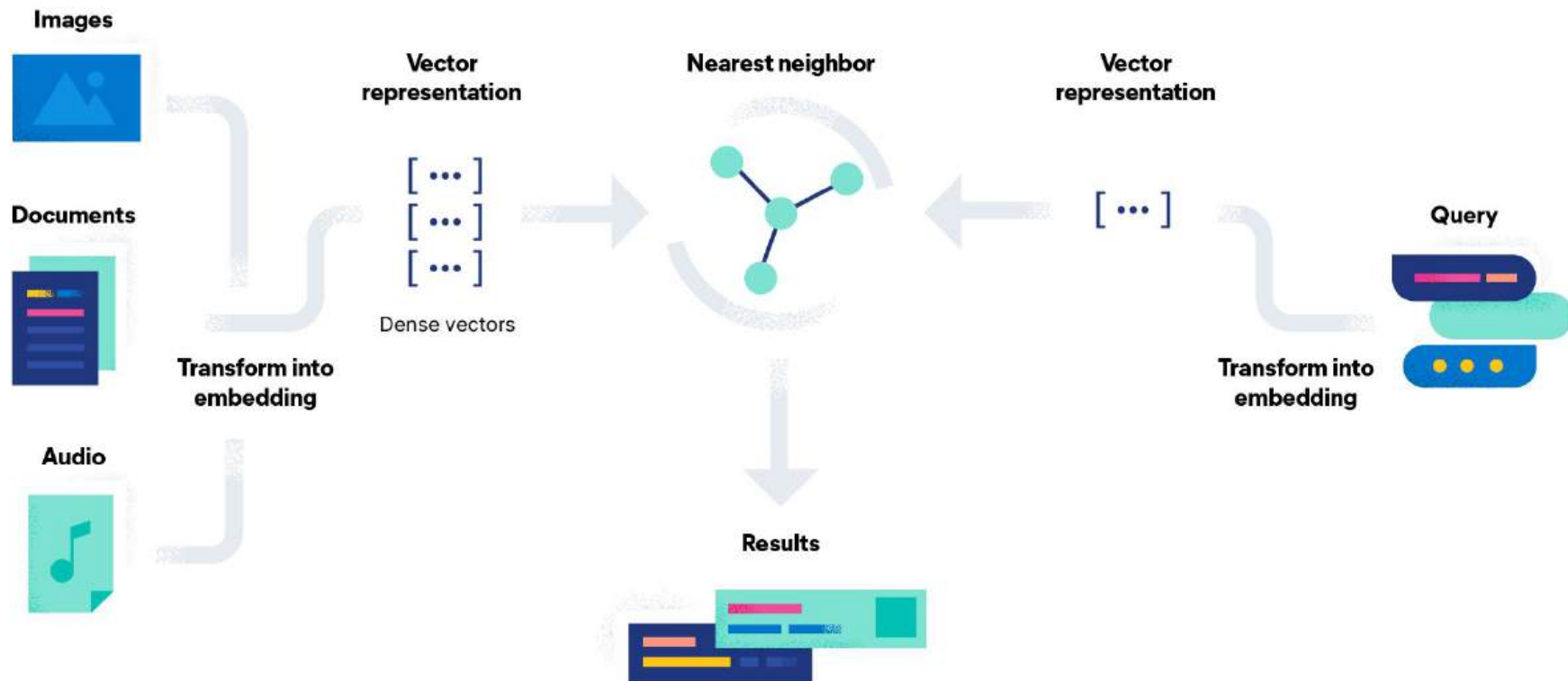
2 Elasticsearch Query

```
GET apple-health-steps/_search { "size": 0, "query": { "bool": {  
  "must": [ { "match": { "location": "Las Vegas, NV" } }, { "range": {  
    "day": { "gte": "2025-05-13", "lte": "2025-05-13" } } } ] } }, "aggs":  
  { "steps_by_hour": { "terms": { "field": "hour", "size": 24, "order": {  
    "total_steps": "desc" } }, "aggs": { "total_steps": { "sum": { "field":  
      "value" } } } } } }
```

3 SQL (Equivalent)

```
SELECT hour, SUM(value) as total_steps FROM apple_health_steps WHERE  
location = 'Las Vegas, NV' AND day = '2025-05-13' GROUP BY hour ORDER  
BY total_steps DESC LIMIT 1;
```

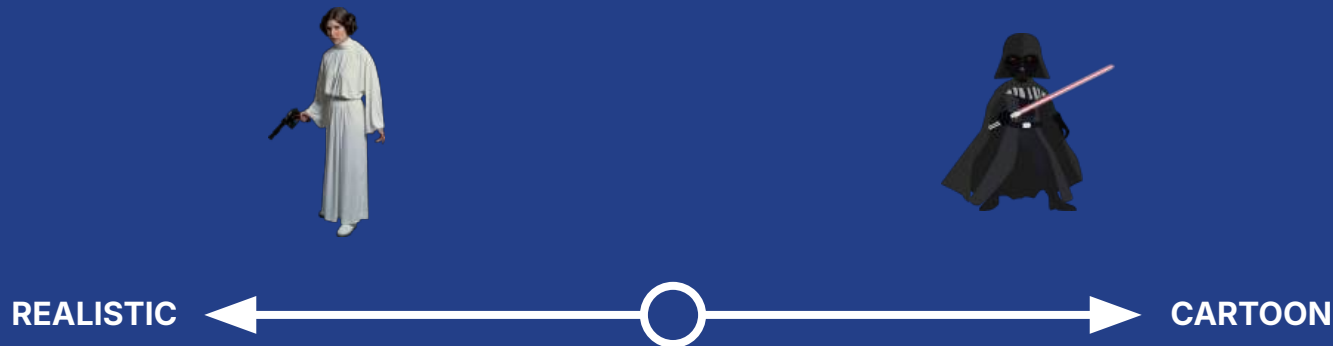
Queries are also vectorized





O que é um Vetor?

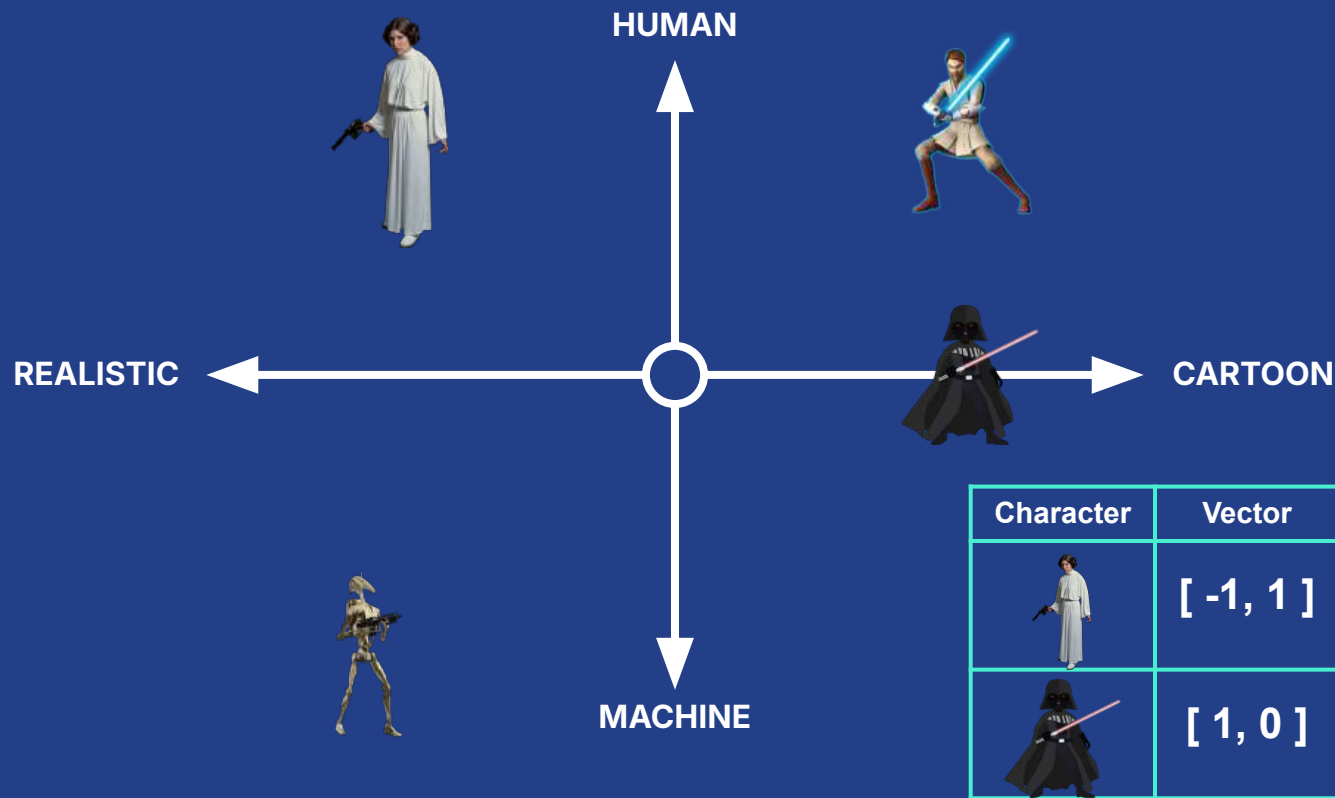
Embeddings represent your data

Example: 1-dimensional vector

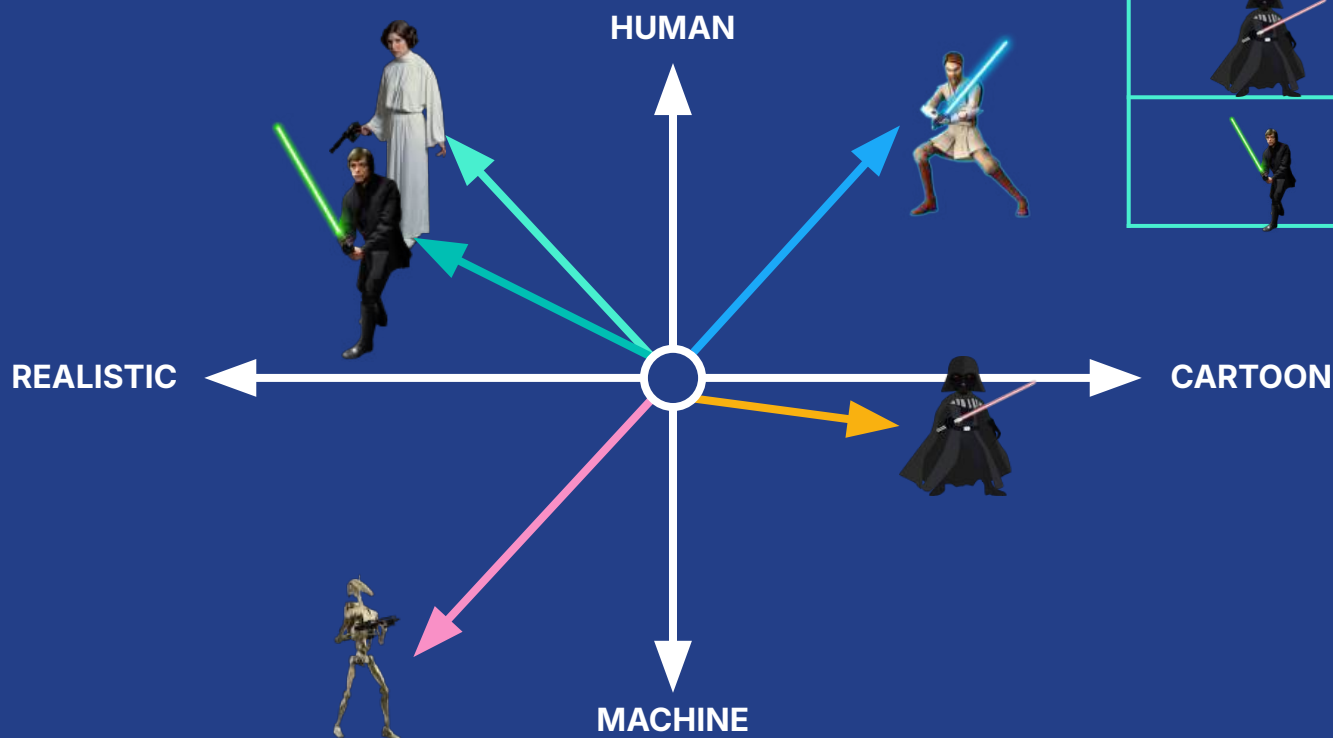





Character	Vector
	$[-1]$
	$[1]$

Multiple dimensions represent different data aspects

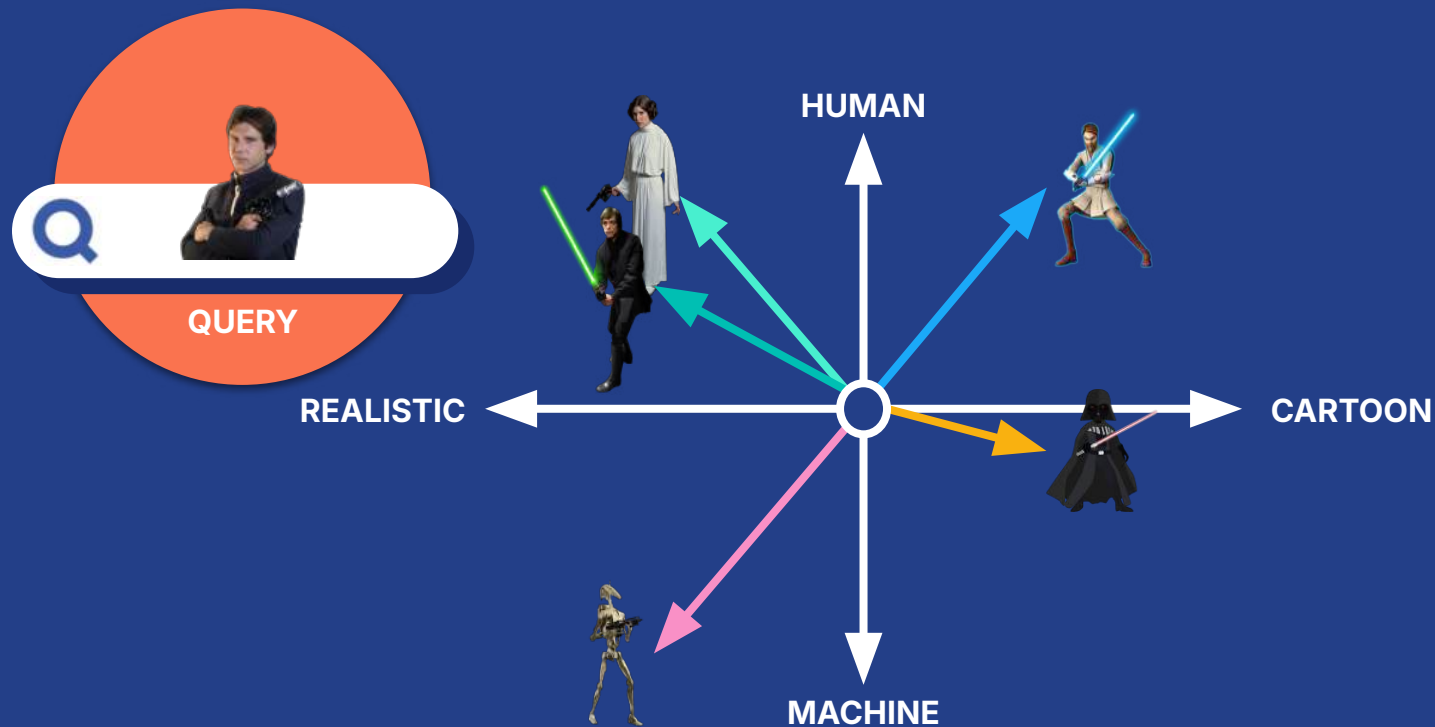


Similar data is grouped together



Character	Vector
	$[-1.0, 1.0]$
	$[1.0, -0.1]$
	$[-1.0, 0.8]$

Vector search ranks objects by similarity (relevance) to the query

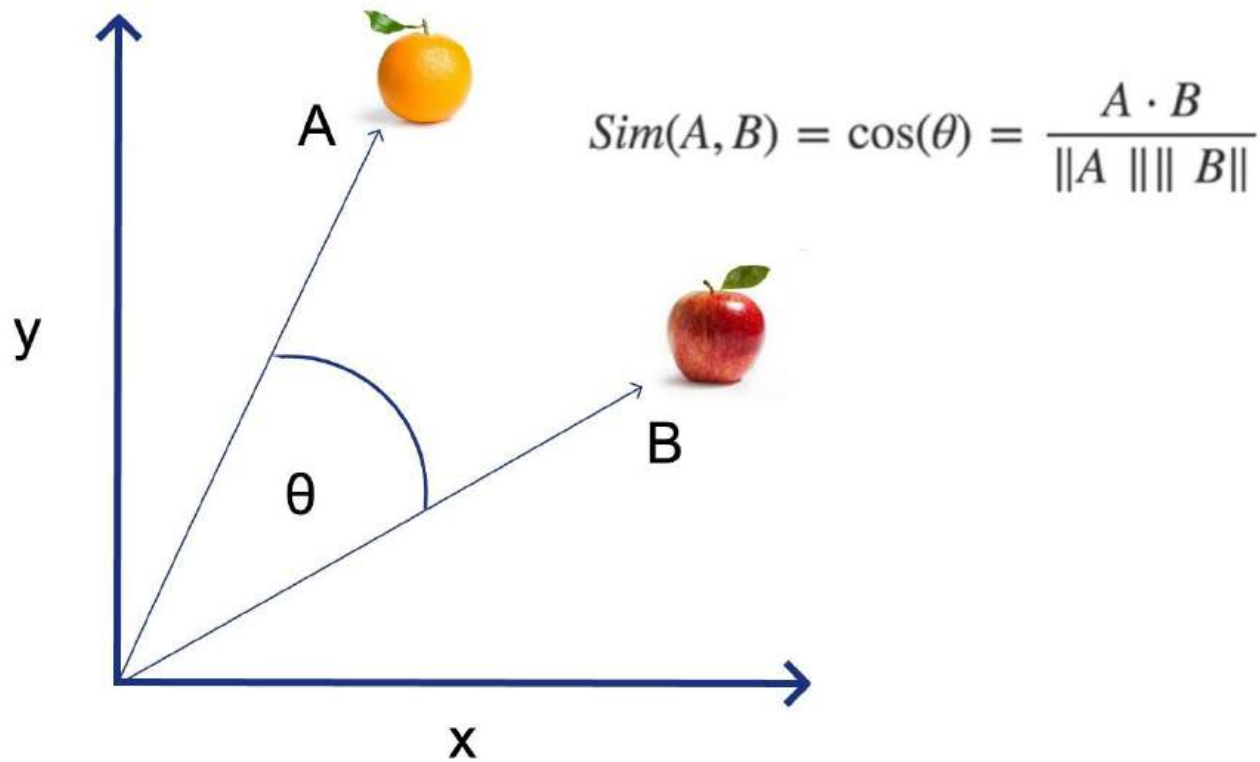


Relevance	Result
Query	
1	
2	
3	
4	
5	

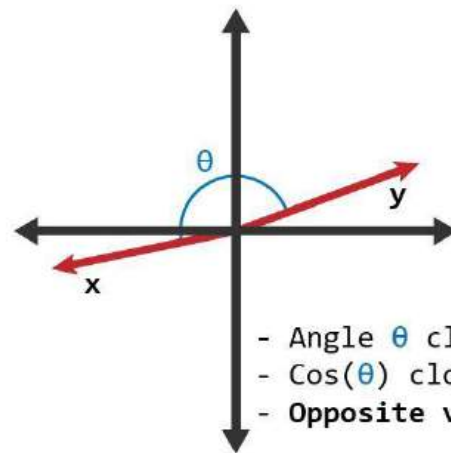
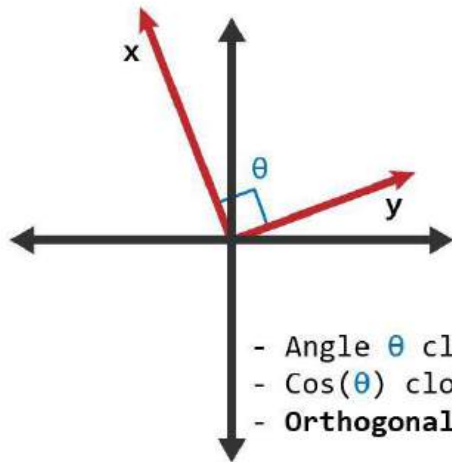
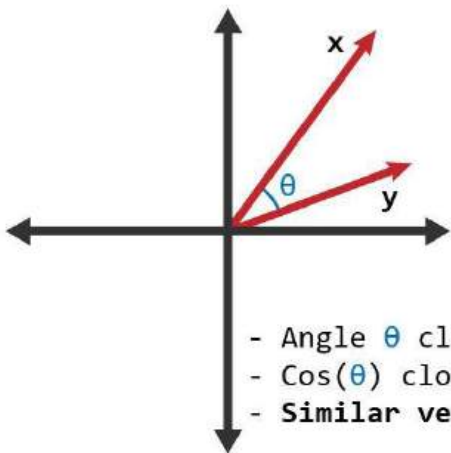


**Como essa busca por
similaridade realmente
funciona?**

Similarity: Cosine



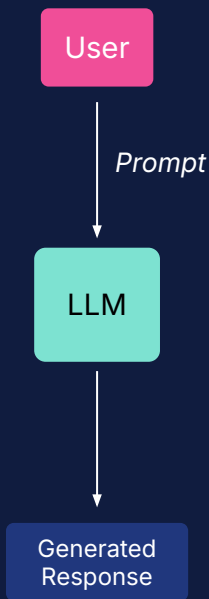
Similarity: Cosine



Da Geração de Texto à Tomada de Decisão

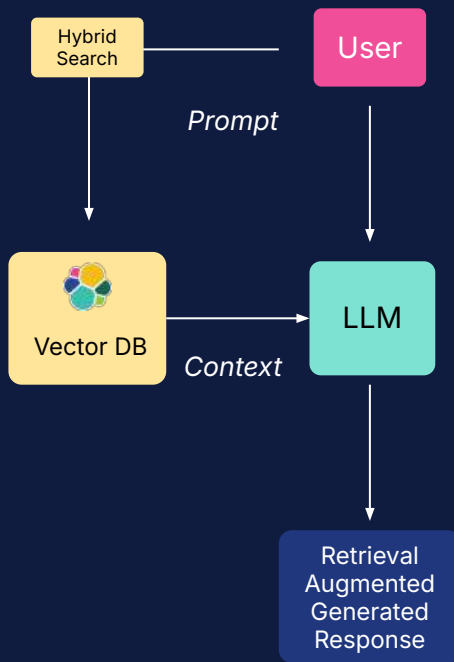
Prompting an LLM

Prompt the LLM and get a response. No other tools or components needed.



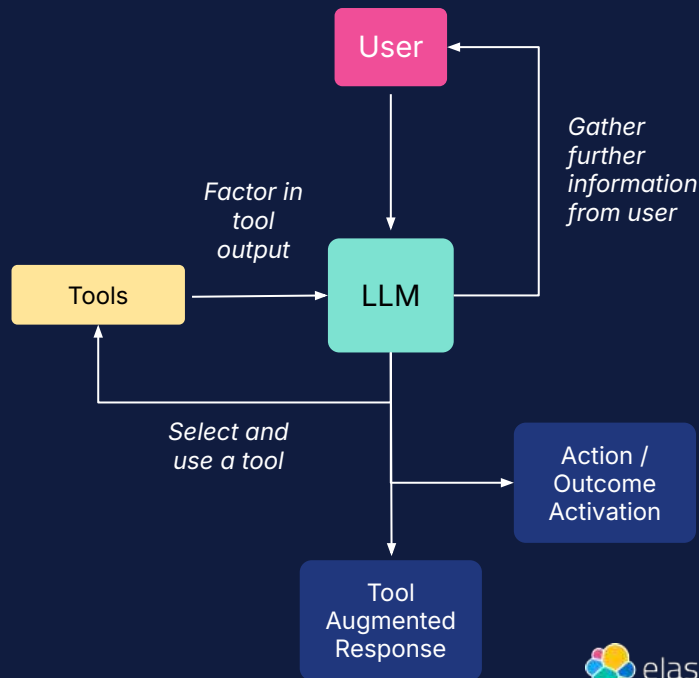
Retrieval Augmented Generation

Add a knowledge base to enhance accuracy and add novel information to the LLM



Agentic Flow

Decision Making enabled. LLM can prompt user for information, choose to use tools, interact with other agents, and affect the real world (ie. Triggering alerts, sending messages, etc...)



A Evolução da Busca

Busca 1.0

1970-2020

SQL, NoSQL

Queries rígidas

Só para DBAs

Busca 2.0

2020-2024

RAG + Embeddings

Vector Stores

Ainda limitado

Busca 3.0

2024+

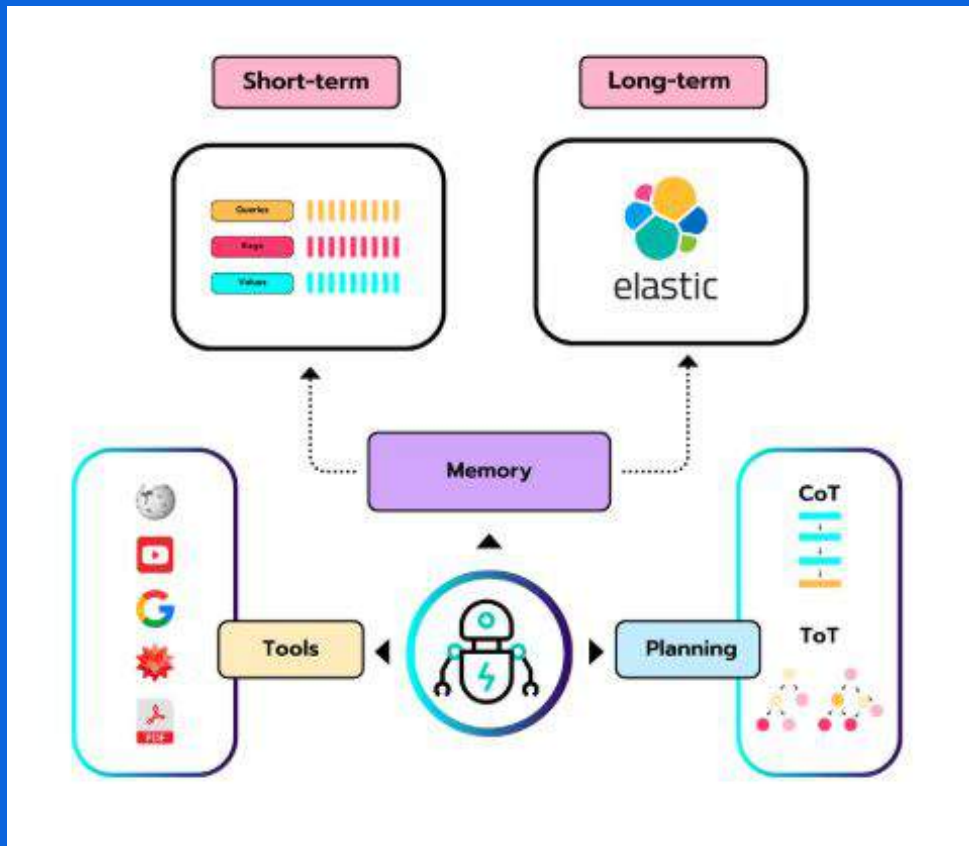
RAG Agêntico

MCP + LLMs

Conversas naturais

De **queries complexas** para **conversas simples**

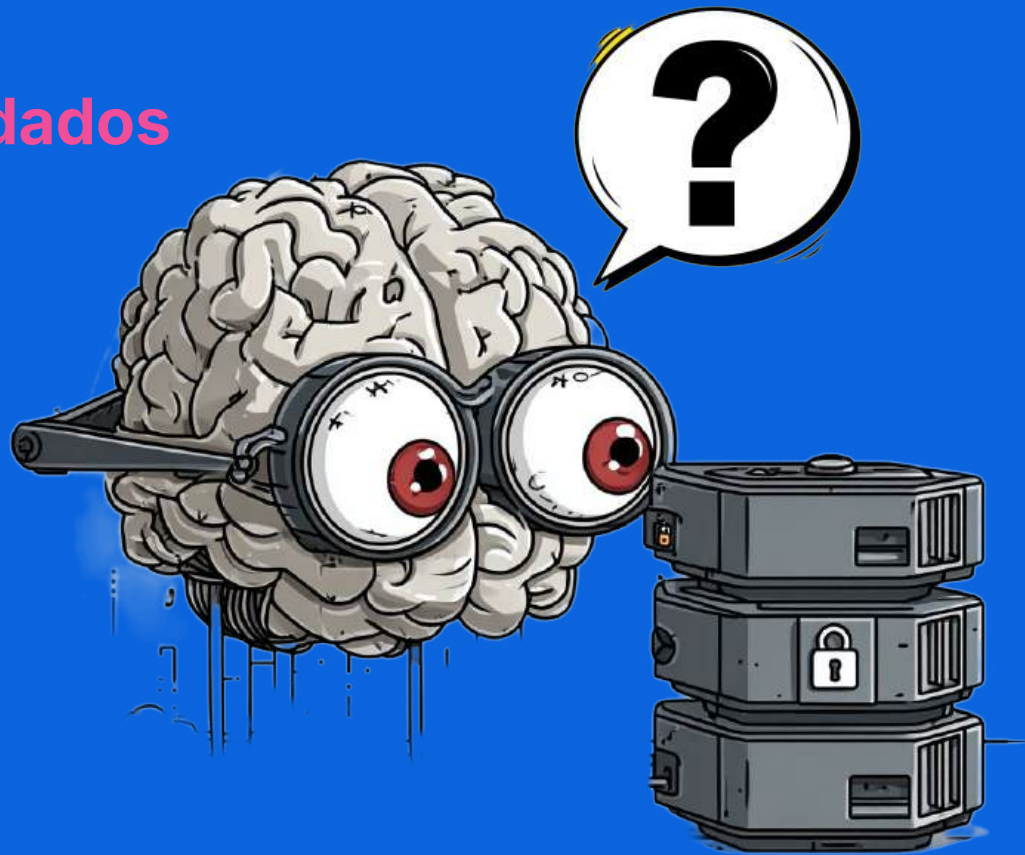
O que é um Agente de IA ?



O Problema:

LLMs precisam de seus dados

- As LLMs não possuem acesso a dados privados em tempo real.
- Retrieval-Augmented Generation **injecta** dados no contexto da LLM.
- RAG **padroniza** os métodos de comunicação entre o modelo e a fonte de dados.
- LLMs sofrem com cortes de conhecimento sem dados privados.



A solução:

MCP, a "Linguagem" padrão para o **RAG**

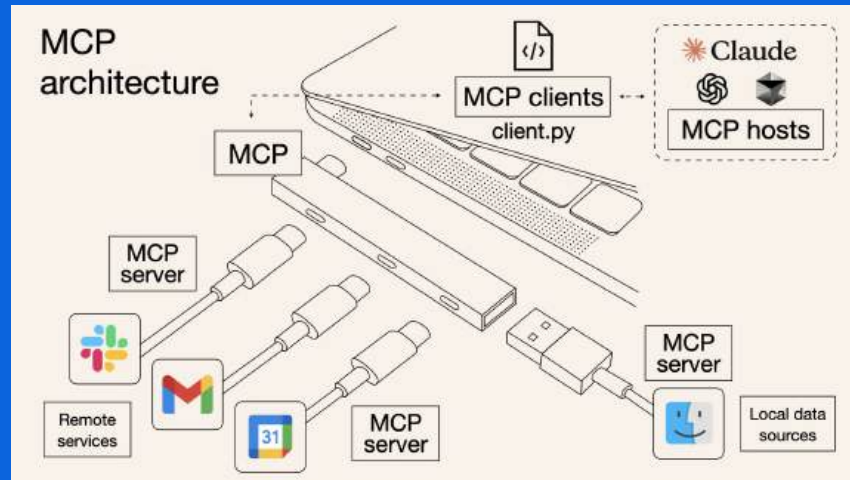
- Pense no **MCP** como o HTTP. Assim como o HTTP é uma linguagem padrão para navegadores obterem páginas de qualquer servidor web, o MCP é uma linguagem padrão para modelos de IA obterem contexto de qualquer fonte de dados.
- Ele age como um **adaptador** universal. O MCP define um conjunto padrão de "**tools**" (como busca) para que a IA não precise aprender a linguagem de consulta específica para cada banco de dados diferente com o qual precisa se comunicar.
- Qualquer modelo compatível com MCP pode descobrir e usar essas ferramentas sem código de integração personalizado.



O que é MCP

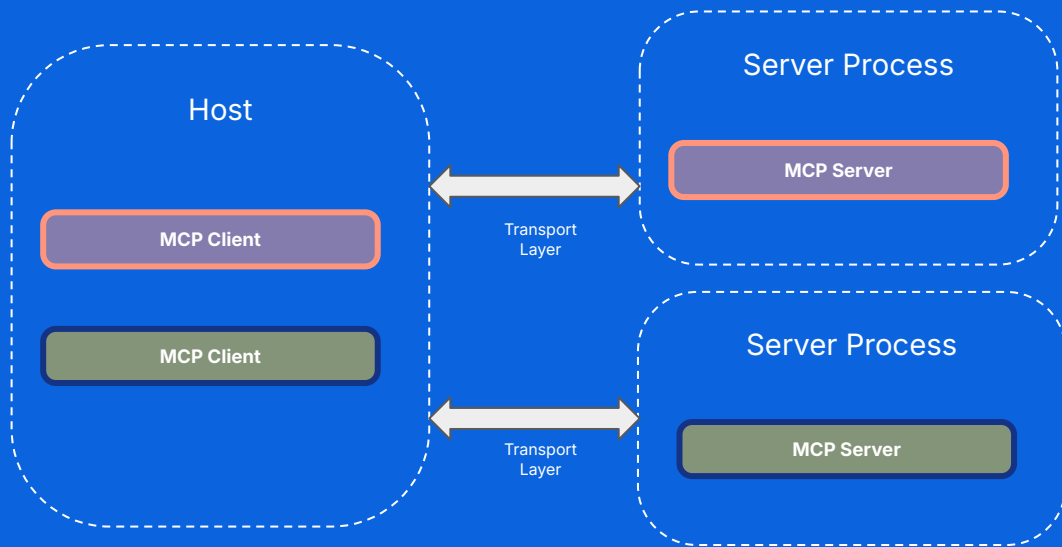
Model Context Protocol

- Forma padronizada de LLMs acessar ferramentas
- Criado pela **Anthropic** para resolver o compartilhamento de contexto
- Três primitivos principais: Recursos, Ferramentas e Prompts
- Pense nisso como uma API especificamente para ferramentas de LLM



Anthropic defines it as the USB-C port equivalent for agentic systems

MCP Server Architecture



MCP follows client-server architecture:

Hosts = LLM apps (Claude Desktop) that initiate connections

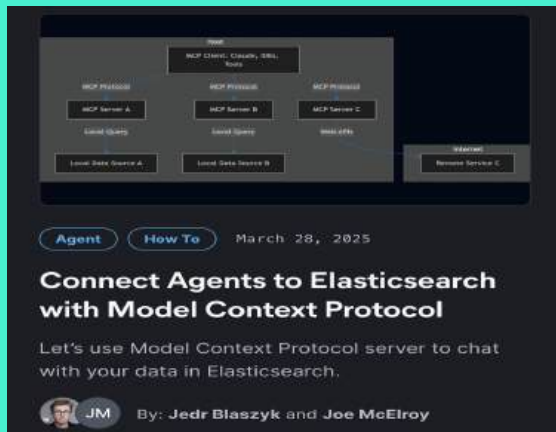
Clients = Maintain 1:1 connections with servers, *INSIDE* host

Servers = Provide context, tools, prompts to client

Duas Abordagens para MCP + Elastic Personalizada

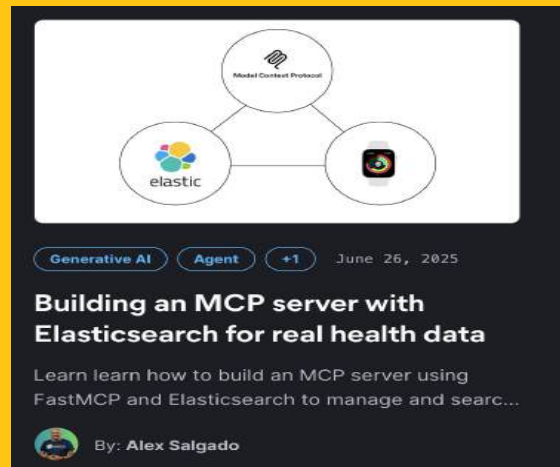
Oficial

- Propósito geral *
- Configuração fácil
- Customização limitada



Customizada

- Específica para o domínio
- Requer codificação
- Controle completo



Hoje vamos focar na abordagem personalizada =)

Portfolio Pilot AI

Chat - Elastic

Console - Dev Tools - Elastic

Portfolio-Pilot-AI

localhost:5601/app/chat/conversations/905f3ea3-...

Work

Project Chat Conversations

Give feedback

TU

developer

Elasticsearch

Home

Discover

Dashboards

Chat

Conversations

Agents

Tools

Build

Index Management

Playground

Connectors

Relevance

Synonyms

Query Rules

Inference Endpoints

Identifying the Highest Valued Portfolio

+ New

Tool: neg_news_reports_with_pos

Tool call args

```
{  "time_duration": "48 hours"}
```

Tool call result

In the last 48 hours, there have been two significant negative news reports affecting financial assets:

1. Freeport-McMoRan Inc. (FCX) Q3 Earnings Summary:

- Sentiment:** Negative
- Sector:** Materials
- Summary:** The report highlights negative aspects of Freeport-McMoRan Inc.'s third-quarter earnings, impacting various account holders with positions in FCX.
- Account Holders Affected:** Numerous account holders across different risk profiles and account types, including Retirement, Growth, Conservative, and Aggressive Growth.

2. McDonald's CEO Abruptly Exits Amid Franchisee Unrest:

- Sentiment:** Negative
- Sector:** Consumer Discretionary
- Summary:** The news reports the sudden exit of McDonald's CEO, causing unrest among franchisees and affecting the company's stock sentiment negatively.
- Account Holders Affected:** Many account holders with positions in McDonald's Corp., spanning various risk profiles and account types, such as Income-Focused, Conservative, Balanced, and

Ask anything

Financial Data Analyst

elastic

Use New Fin Agent

Desafio:

Converse com sua saúde

Pergunta 1:

"Quantos passos eu dei **ontem**?"

Question 2:

"Quantas voltas ao redor do maracanã eu teria completado se eu caminhasse o mesmo número de passos que dei **ontem**?"

Construindo um servidor MCP
personalizado que conecta o **Claude AI**
Desktop aos meus dados de **fitness**



RAG Agêntico: A **Inteligência** da Busca 3.0

RAG Tradicional

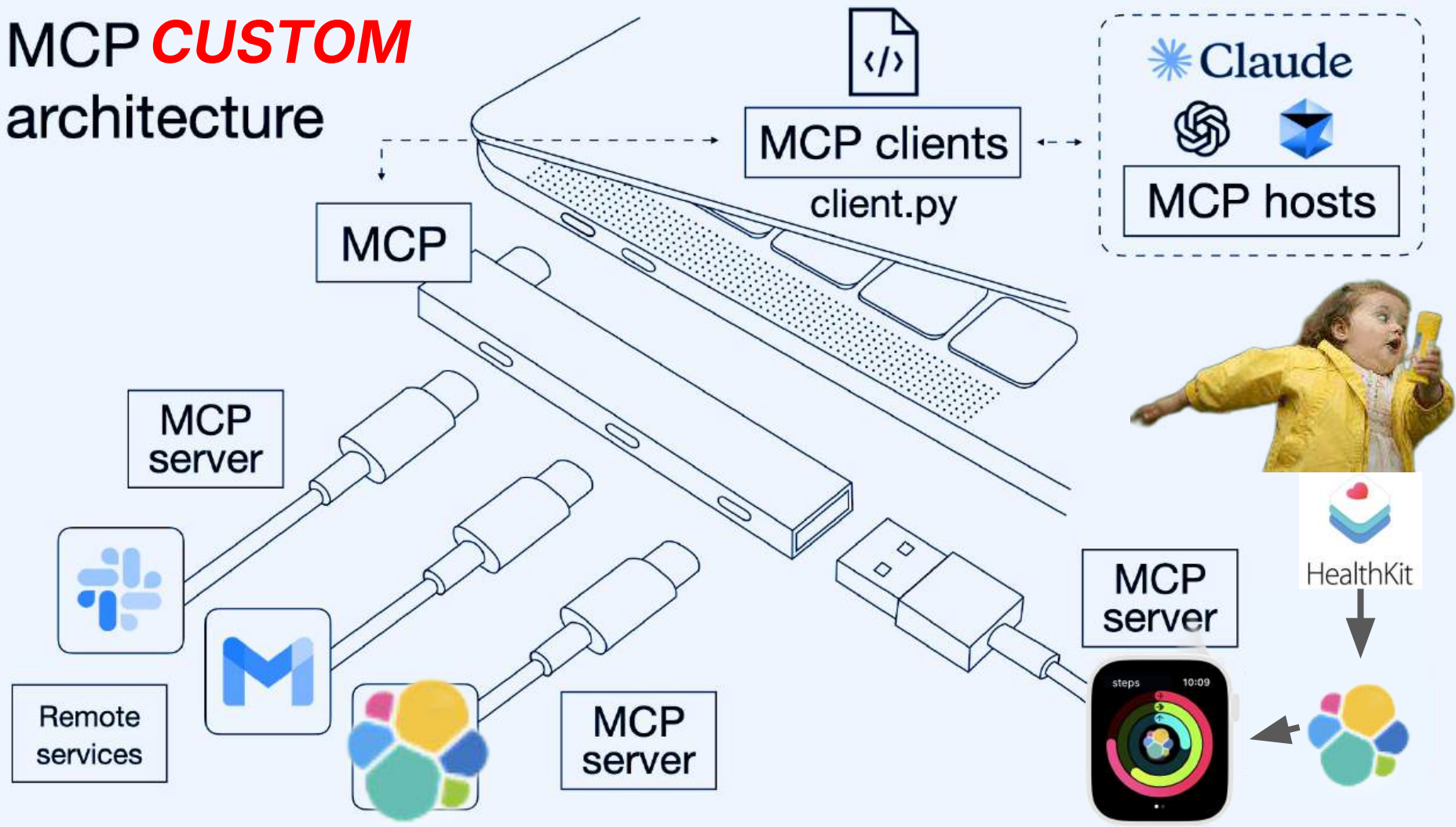
- Busca → Retrieve → Generate
- Pipeline fixo
- Uma fonte por vez
- Sem raciocínio

RAG Agêntico com MCP

- Planeja → Executa → Valida → Itera
- Adaptativo
- Multi-fonte simultâneo
- Raciocínio complexo

O LLM **decide** quando, onde e como buscar

MCP *CUSTOM* architecture



Como começar?

```
bash

# Create project and install dependencies
uv init apple-watch-mcp
cd apple-watch-mcp
uv venv
source .venv/bin
uv add "mcp[cli]" elasticsearch httpx pydantic
```

Configuração simples com ferramentas Python comuns

A CLI do MCP fornece ferramentas de desenvolvimento

Apenas algumas dependências necessárias

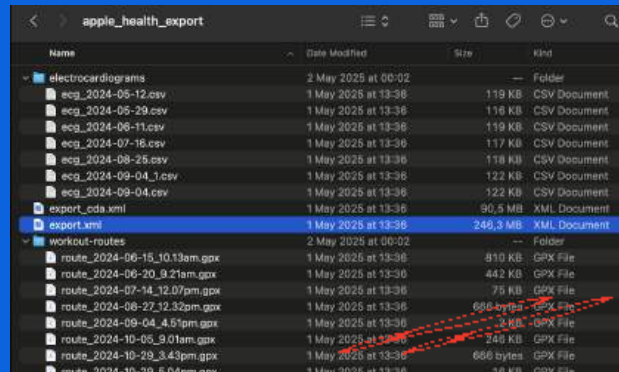
Os dados de exemplo

```
{
  "type": "HKQuantityTypeIdentifierStepCount",
  "sourceName": "Apple Watch",
  "startDate": "2025-05-01 07:58:42",
  "endDate": "2025-05-01 07:59:10",
  "value": 34,
  "day": "2025-05-01",
  "dayOfWeek": "Thursday",
  "hour": 7
}
```

Dados de fitness de série temporal no Elasticsearch

Múltiplos dispositivos rastreando as mesmas métricas

Perfeito para demonstrar capacidades de busca



Inicialização do servidor

```
apple_watch_mcp.py

# Step 1: Import the FastMCP framework
from mcp.server.fastmcp import FastMCP
import httpx

# Step 2: Create your MCP server instance
# FastMCP handles protocol details so you focus on implementation
mcp = FastMCP("apple-watch-steps")

# Step 3: Define your Elasticsearch connection params
ES_HOST = "http://localhost:9201"
ES_INDEX = "apple-health-steps"

# Step 4: Create a helper function for Elasticsearch queries
# This centralizes your query logic and error handling
async def query_elasticsearch(query: dict):
    async with httpx.AsyncClient() as client:
        response = await client.post(
            f"{ES_HOST}/{ES_INDEX}/_search", json=query
        )
    return response.json()
```

```
uv run mcp dev apple_watch_mcp.py
```

1

CLI Loads Python Module

MCP imports your Apple Watch module and finds FastMCP instance

2

Uvicorn ASGI Server Starts

Server launches on **localhost:8000** with SSE protocol

3

Resources, Tools, Prompts Registered

Server scans for decorated functions and registers capabilities

4

Browser Opens with MCP Inspector

Web UI for testing Resources and Tools connects via SSE

5

First Connection to Elasticsearch

Established when first Resource or Tool is invoked via Inspector

MCP: Os Três Primitivos Fundamentais



Resources

Data Access
Provide context to LLM
Like GET endpoints



Tools

Actions & Computation
Execute operations
Like POST endpoints



Prompts

Interaction Templates
Guide conversations
Like workflow recipes

juntos, essas primitivas formam um sistema completo para conectar LLMs a funcionalidades externas

Resources: Os Olhos do LLM

```
apple_watch_mcp.py

@mcp.resource("health://steps/latest")
async def get_latest_steps() -> str:
    """Get latest step counts"""
    # URI pattern choice: namespaced by domain
    # Return format: Always JSON strings for consistency
    # Query design: Simple, focused on single concern
    query = {
        "query": {"match_all": {}},
        "sort": [{"endDate": {"order": "desc"}}],
        "size": 10
    }
    data = await query_elasticsearch(query)
    # Process and return formatted results...
```

- **URI Pattern:** "health://steps/latest"
- **Purpose:** Retrieves data without modification
- **Acts like:** Read-only GET endpoints

Recursos fornecem **contexto** ao expor seus dados diretamente ao LLM

Tools: As Mãos do LLM

```
apple_watch_mcp.py

@mcp.tool()
async def query_step_data(params: QueryParams) -> str:
    """Query step data with parameters"""
    # Extract parameters
    start_date = params.start_date
    end_date = params.end_date
    aggregation = params.aggregation

    # Build Elasticsearch query
    query = {"query": {"match_all": {}}}
    filters = []

    # Add date filters if specified
    if start_date or end_date:
        date_filter = {"range": {"day": {}}}
        if start_date:
```

- **Accepts Parameters:** Structured input objects
- **Dynamic Logic:** Varies based on parameters
- **Active Processing:** Not just data retrieval

Tools enable **action and computation** — essential for complex queries and analysis

Prompts: Os Mapas para o LLM

```
apple_watch_mcp.py

@mcp.prompt()
def daily_report(date: str = None) -> str:
    """Daily step analysis report"""
    if date:
        return f"""Analyze my step data for {date}.
Please include:
1. Total steps taken
2. Most active periods
3. Device breakdown
4. Compare to my weekly average
5. Visualize my activity pattern"""
    else:
        return """Analyze today's step data..."""
```

- **Appears as:** / commands in Claude
- **Provides Structure:** For common tasks
- **User-Triggered:** Start specific workflows

Prompts criam **experiências consistentes** para padrões de análise frequentes.

Validação de Tipo - Crítico para Confiabilidade

```
apple_watch_mcp.py

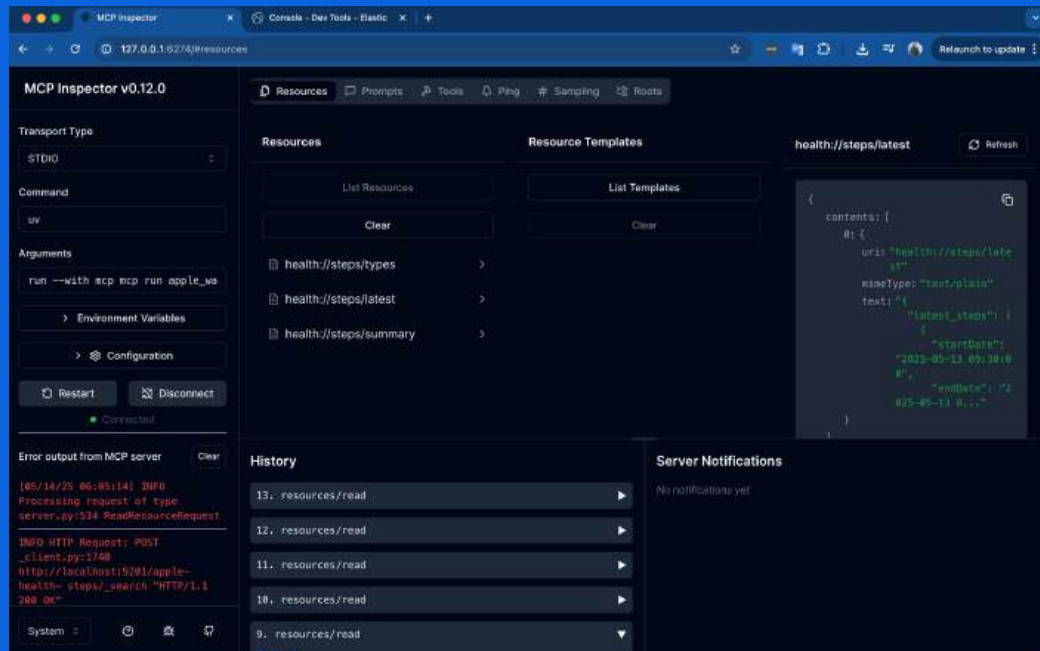
# Pydantic model for parameter validation
class QueryStepDataParams(BaseModel):
    start_date: Optional[str] = None
    end_date: Optional[str] = None
    aggregation: Optional[str] = None
    device: Optional[str] = None

    @field_validator('start_date', 'end_date')
    def validate_date_format(cls, value):
        if value is None:
            return value
        try:
            datetime.strptime(value, "%Y-%m-%d")
            return value
        except ValueError:
            raise ValueError("Invalid date format. Use YYYY-MM-DD")
```

Why Pydantic?

1. Runtime **validation** with clear error messages
2. Self-**documenting** code
3. **Seamless** FastMCP integration

Testando com o MCP Inspector



For Developers

- Test resources and tools interactively
- View raw JSON responses from Elasticsearch

Key Features

- Real-time API testing
- Request history tracking

Integration

- Shows live Elasticsearch queries
- Helps debug before deploying to Claude

Deployment & Integration

mcp install apple_watch_mcp.py

🌟 Bem-vindo(a), alex

Como posso ajudar você hoje?



Pesquisa **DELTA**

Claude 3.7 Sonnet ▾



```
claude_desktop_config.json

{
  "mcpServers": {
    "Apple Health Steps": {
      "command":
        "/Users/alexsalgado/.local/bin/uv",
      "args": [
        "--directory",
        "/Users/alexsalgado/Desktop/blog-mcp-server/new-blog/apple-watch-mcp",
        "run",
        "apple_watch_mcp.py"
      ]
    }
  }
}
```

Demo: A Mágica Acontecendo

Busca 2.0 (Tradicional)

```
GET /health_metrics/_search
{
  "query": {
    "bool": {
      "must": [
        { "range": {
          "timestamp": {
            "gte": "now-30d"
          }
        } },
        { "match": { "user_id": "user123" } }
      ]
    }
  },
  "aggs": {
    "daily_steps": {
      "date_histogram": {
        "field": "timestamp",
        "interval": "day"
      },
      "aggs": {
        "avg_steps": {
          "avg": { "field": "metrics.steps" }
        }
      }
    }
  }
}
```

Busca 3.0 (Com MCP)

💬 "Como foram meus passos este mês?"

O LLM:

1. Entende a pergunta
2. Chama `search_health_data()`
3. Processa resultados
4. Gera resposta natural

Demo time

🌟 Bem-vindo(a), alex

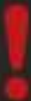
How many laps around the Las Vegas Sphere would I have completed if I walked the same number of steps I took yesterday?



Pesquisa

BETA

Claude 3.7 Sonnet ~



starting to reason...

Casos de Uso Prático com MCP Customizado



Análise de Logs

"Por que o sistema caiu ontem às 15h?"

MCP conecta aos logs de aplicação, servidores e infraestrutura para identificar padrões e causas raiz automaticamente.



Business Intelligence

"Qual produto tem melhor margem no Q3?"

Integra dados de vendas, custos e inventário para análises complexas sem necessidade de SQL ou dashboards.



DevOps & Monitoramento

"Qual serviço está consumindo mais recursos?"

Conecta métricas de Prometheus, CloudWatch ou Datadog para insights operacionais em tempo real.



Gestão de Documentos

"Quais contratos vencem este mês?"

Analisa PDFs, contratos e documentos legais extraindo informações críticas automaticamente.



Segurança & Compliance

"Houve tentativas de acesso suspeitas?"

Monitora logs de segurança, detecta anomalias e garante conformidade com políticas internas.



Integração de Sistemas

"Sincronize dados entre CRM e ERP"

Orquestra fluxos de dados entre sistemas legados e modernos sem código adicional.

Padrões de Comportamento Agêntico

Chain of Thought (CoT)

Processo linear de raciocínio passo a passo, onde o LLM articula explicitamente seu raciocínio em uma sequência lógica.

Quando usar: Problemas com caminho de solução relativamente claro, tarefas de precisão factual e matemática.

Tree of Thoughts (ToT)

Exploração de múltiplos caminhos de raciocínio simultaneamente, avaliando diferentes possibilidades antes de escolher o melhor caminho.

Quando usar: Problemas complexos com múltiplas abordagens possíveis, tarefas criativas e raciocínio sob incerteza.

ReAct (Reasoning + Acting)

Combina raciocínio com ação em um ciclo interativo: Raciocínio → Ação → Observação → Raciocínio.

Quando usar: Cenários que requerem interação contínua com o ambiente, busca iterativa de informações.

Self-RAG

O modelo decide autonomamente quando e como recuperar informações, e depois critica a relevância dos resultados obtidos.

Quando usar: Quando a confiabilidade das fontes varia, para filtrar informações irrelevantes.

Self-Reflection

O LLM avalia criticamente suas próprias respostas, identificando falhas e refinando suas conclusões iterativamente.

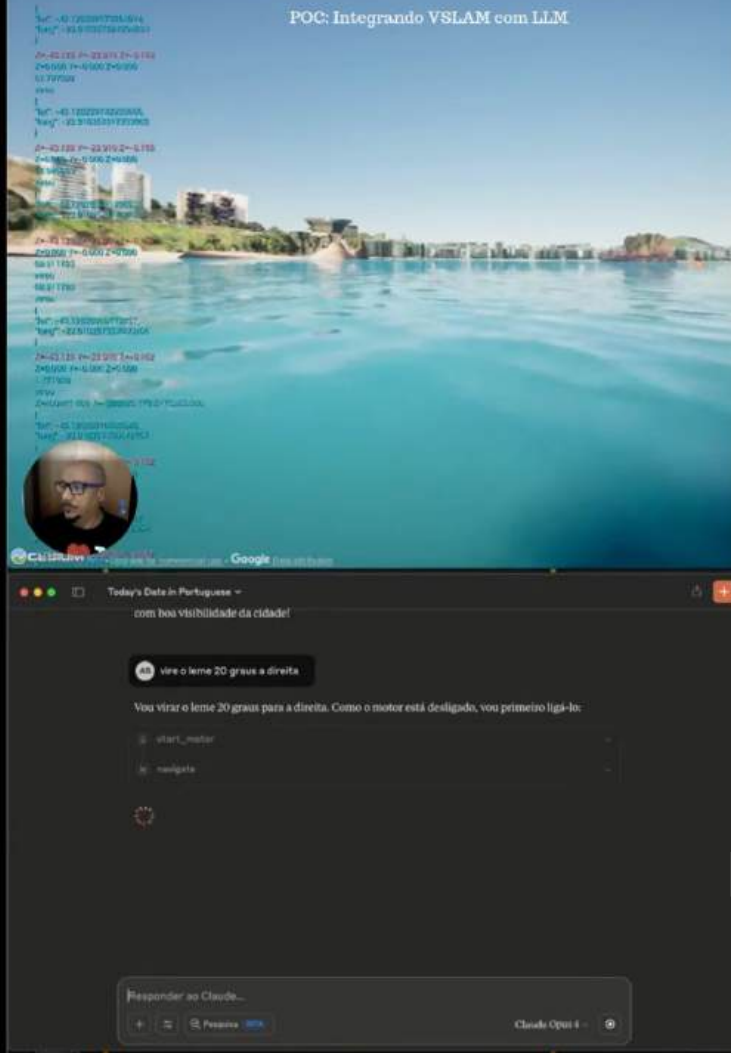
Quando usar: Para aumentar precisão em domínios críticos, reduzir alucinações e inconsistências.

Plan-and-Solve (PS)

Enfatiza o planejamento explícito antes da resolução, dividindo o problema em etapas claramente definidas.


Quando usar: Tarefas complexas com múltiplas etapas, problemas que requerem organização cuidadosa.

Tendência atual: Sistemas estado-da-arte frequentemente combinam vários destes padrões, selecionando dinamicamente o mais adequado para cada contexto. O ReAct é particularmente adequado para Agentic RAG por integrar naturalmente raciocínio e ações como consultas a bases de conhecimento.



Barcos Autonomos

Recursos para desarrolladores: Elasticsearch Labs



Generative AI Agent +1 June 26, 2025

Building an MCP server with Elasticsearch for real health data

Learn how to build an MCP server using FastMCP and Elasticsearch to manage and search...

By: Alex Salgado

elasticsearch-labs Public

About

Elasticsearch Guides, Notebooks & Example Apps for Search Applications

search-labs.elastic.co/search-labs

python search elasticsearch ai
vector applications opensearch elastic
chatgpt chatgpt langchain
opensearch-chatgpt genai genstack
vectordatabase

Readme

Apache-2.0 license

Security policy

Activity

109 stars

178 watching

40 forks

Report repository

Languages

Jupyter Notebook	93.7%
Python	2.9%
TypeScript	1.3%
Handlebars	0.7%
JavaScript	1.6%
CSS	0.2%
Other	0.2%



Vector Database November 8, 2023

Implementing image search: vector search via image processing in Elasticsearch

Learn how to implement image search with an example. This blog covers how to use vector search through image processing in...

By: Alex Salgado

Alex Salgado · Developer Advocate @alexsgalgon

youtube.com/@OfficialElasticCommunity

elastic.co/search-labs

github.com/elastic/elasticsearch-labs

Recursos para desenvolvedores: Junte-se à Comunidade Elastic

Meetups

Elastic User Groups

Estamos sempre em busca de organizadores, palestrantes e participantes. Encontre mais eventos Elastic em todo o mundo em community.elastic.co

elastic.co/community



Meetup Elastic: Rio de Janeiro/RJ



Meetup Python Floripa: Florianópolis/SC



@alexsalgadoprof

100% FREE

ELASTIC

Cursos

POR TEMPO LIMITADO



elastic



@alexsalgadopro

f
PYTHON BRASIL 2025

Não se trata de **se** você vai usar,
mas **quando**. E, nessa jornada,
cada **passo** importa.

Apenas comece.



Obrigado

