Meu primeiro chabot

O que são, como funcionam e como fazer um

Sumário

01

Introdução aos Chatbots

O que são e tipos de chatbots

Introdução aos chatbots

“No nível mais básico, um chatbot é um programa de computador que simula e processa conversas humanas (escritas ou faladas)”

“Os chatbots podem ser tão simples quanto programas rudimentares que respondem a uma consulta simples com uma resposta de linha única ou tão sofisticados quanto assistentes digitais.”

Introdução aos chatbots

Baseados em Regras

Possuem respostas pré-definidas para perguntas comuns

Inteligentes (ML/DL/Gen-AI)

São capazes de conduzir conversas mais dinâmicas e sofisticadas com o público.

Híbridos (Regras e IA)

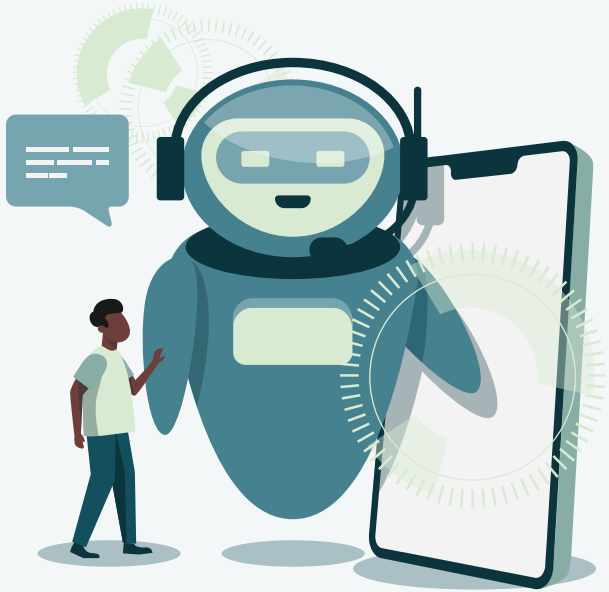
Usa fluxos de conversação predefinidos (tecnologia do chatbot baseado em regras) com respostas de IA.

02

Chatbots com Gen-AI

Conceito e hands on

Modelos de IA generativa são capazes de criar conteúdo novo e original baseando-se em padrões aprendidos de vastos conjuntos de dados.



Modelos proprietário x Open Source

GPT-4
OpenAI

Gemma 3
Google

Claude 4
Antrophic

LLaMA3
Meta

R1
DeepSeek

Onde encontrar
modelos open source?
ollama.com/library

QUAL É O MELHOR?

GPT-4
OpenAI

Gemma 3
Google

Claude 4
Antrophic

LLaMA3
Meta

R1
DeepSeek

Onde encontrar
modelos open source?
ollama.com/library

Hands on 1

Link: Colab

Hands on 1

Messages

Role	Para que serve?
system	Configurar o comportamento do modelo
user	Mensagens enviadas pelo usuário
assistant	Respostas anteriores do modelo (para manter contexto)

Hands on 1

Options

Parâmetro	Para que serve?
temperature	Controla a criatividade vs. precisão das respostas
top_k	Limita a escolha do modelo aos k tokens mais prováveis
top_p	Seleciona tokens acumulando probabilidade até atingir top_p
num_ctx	Define o tamanho do contexto (em tokens) que o modelo lembra

Hands on 1

Options

Parâmetro	Para que serve?
num_predict	Limita o número máximo de tokens gerados na resposta
repeat_penalty	Penaliza repetições de palavras/frases
seed	Define uma semente aleatória para reproduzir os mesmos resultados
stop	Interrompe a geração quando encontra essas strings

Hands on 1

E quando eu quero adicionar muito conteúdo personalizado?

03

RAG

Conceito, arquitetura e hands on

RAG

(Retrieval-Augmented Generation)

“Arquitetura híbrida que combina a capacidade generativa dos Modelos de Linguagem de Grande Escala (LLMs) com a precisão da recuperação de informações de bases de conhecimento externas.”

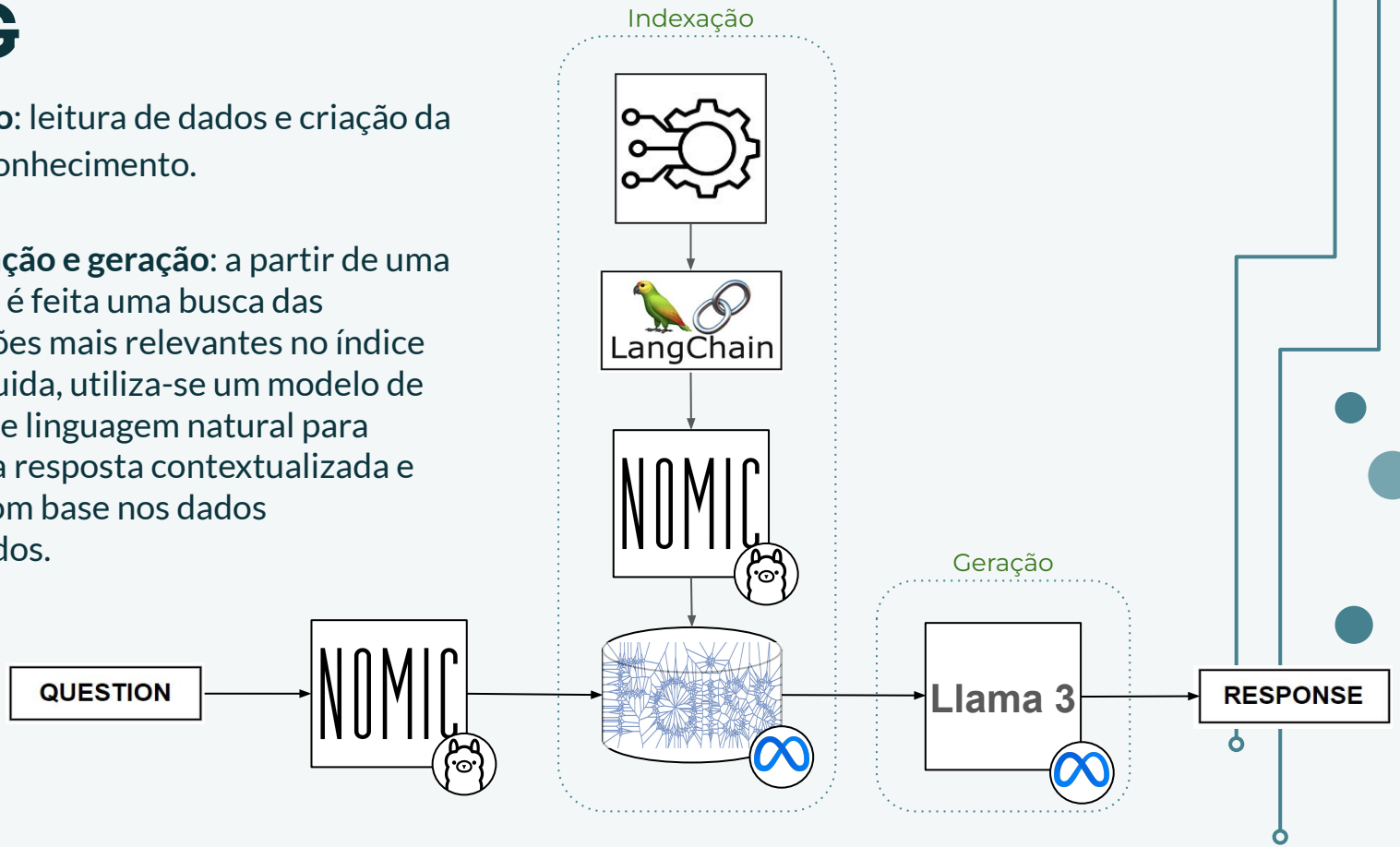
A partir de uma pergunta, o sistema identifica quais informações externas podem ser úteis, faz a busca e traz os dados para o modelo.

O modelo, então, analisa o conteúdo e produz uma resposta integrada, usando tanto o material recuperado quanto seu próprio repertório.

RAG

Indexação: leitura de dados e criação da base de conhecimento.

Recuperação e geração: a partir de uma pergunta, é feita uma busca das informações mais relevantes no índice e, em seguida, utiliza-se um modelo de geração de linguagem natural para gerar uma resposta contextualizada e precisa com base nos dados recuperados.

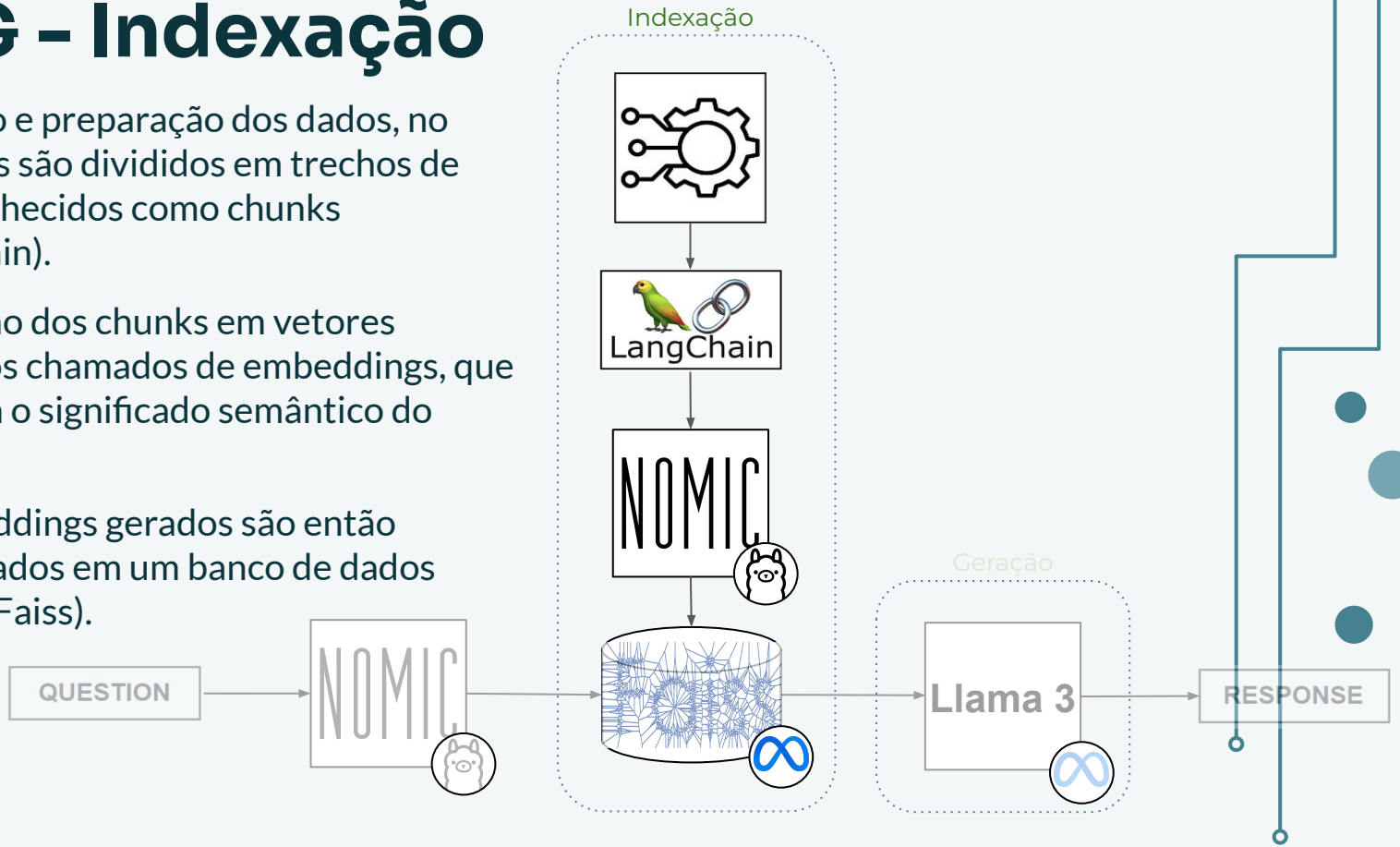


RAG – Indexação

Aquisição e preparação dos dados, no qual estes são divididos em trechos de texto conhecidos como chunks (LangChain).

Conversão dos chunks em vetores numéricos chamados de embeddings, que capturam o significado semântico do texto.

Os embeddings gerados são então armazenados em um banco de dados vetorial (Faiss).

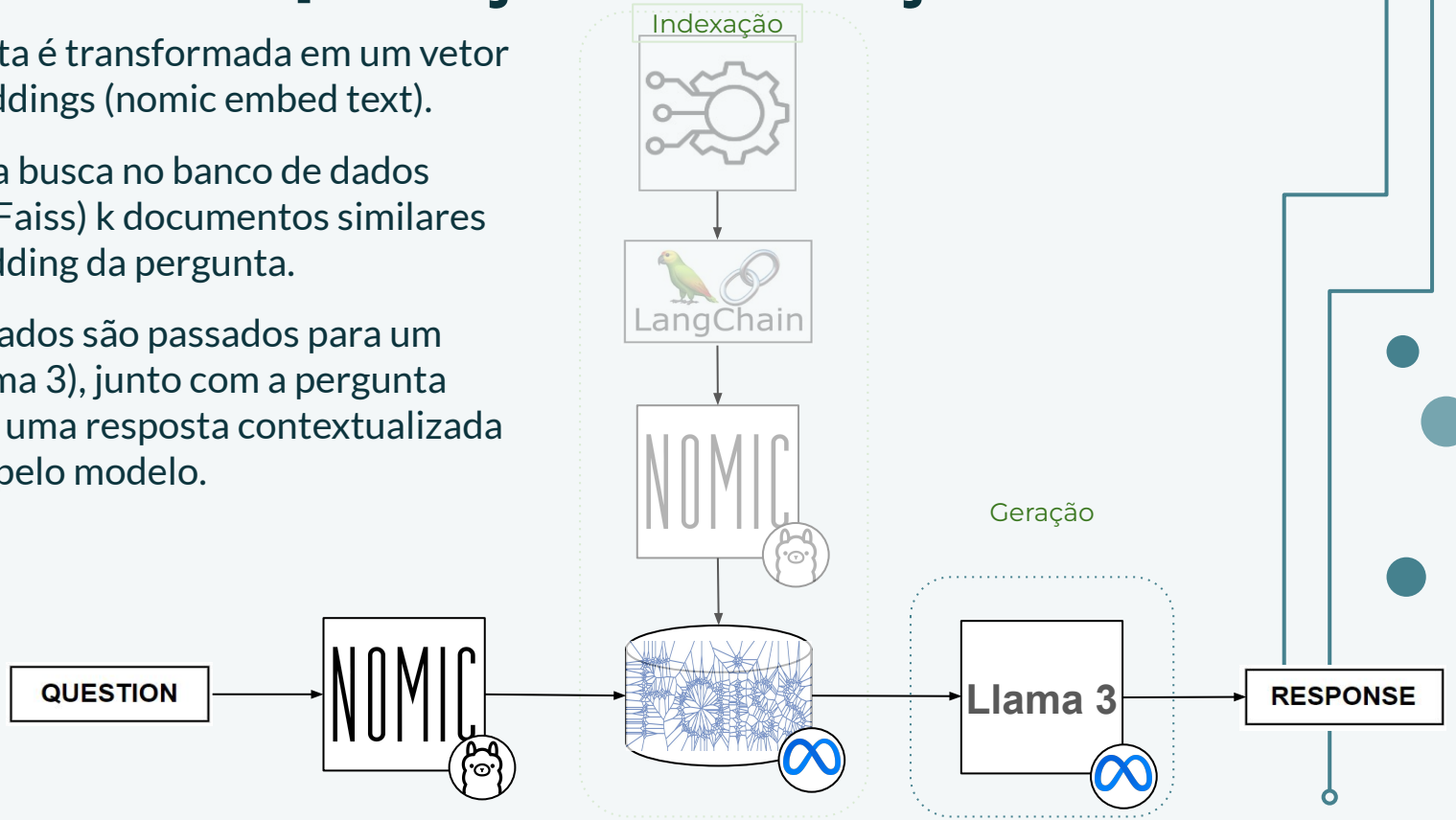


RAG – Recuperação e Geração

A pergunta é transformada em um vetor de embeddings (nomic embed text).

O sistema busca no banco de dados vetorial (Faiss) k documentos similares ao embedding da pergunta.

Os resultados são passados para um LLM (Llama 3), junto com a pergunta original e uma resposta contextualizada é gerada pelo modelo.



Hands on 2

Link: Colab

Hands on 2

Chunk

Param	Para que serve?
chunk_size	Define o tamanho máximo de cada chunk (em caracteres ou tokens)*.
chunk_overlap	Quantidade de sobreposição entre chunks**.
separators	Lista de separadores para dividir o texto.

* Valores muito grandes podem trazer informações irrelevantes, enquanto valores muito pequenos podem fragmentar o contexto.
** Ajuda a manter o contexto entre chunks, mas excesso pode gerar redundância.

Hands on 2

Recuperação de documentos

Param	Para que serve?
k	Número de chunks/documents recuperados.
similarity_threshold	Limiar para filtrar resultados por score*.

* Depende da métrica de similaridade (cosseno, L2, etc.)

04

Showcase de Técnicas Avançadas

Conceitos e casos de uso

Agentes

“Sistemas responsivos em assistentes proativos capazes de executar tarefas complexas.

Diferentemente dos chatbots convencionais que apenas fornecem informações, os agentes são capazes de planejar, executar e validar sequências de ações através de integração com APIs externas, bancos de dados e sistemas corporativos.”

➔ Assistente pessoal
- Building an AI Agent for Google Calendar - Part 1
- Building an AI Agent for Google Calendar - Part 2

➔ Tutor Educacional Personalizado
- Khanmigo (Khan Academy) – explica conceitos de matemática e corrige exercícios passo a passo.

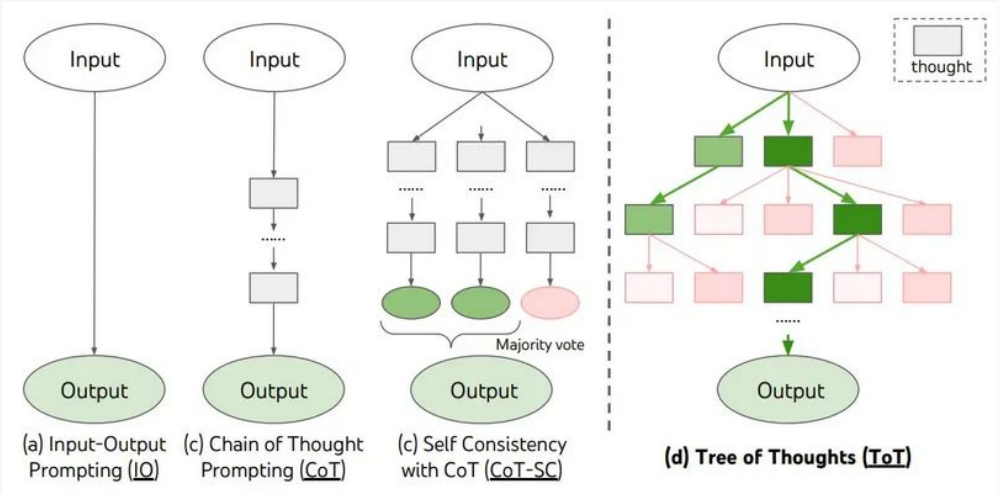
➔ Assistente Médico com Diagnóstico Preliminar
- https://glass.health/

➔ Agente de Resumo de Reuniões
- Fireflies.ai – grava calls, transcreve e gera resumos

Reasoning Chains

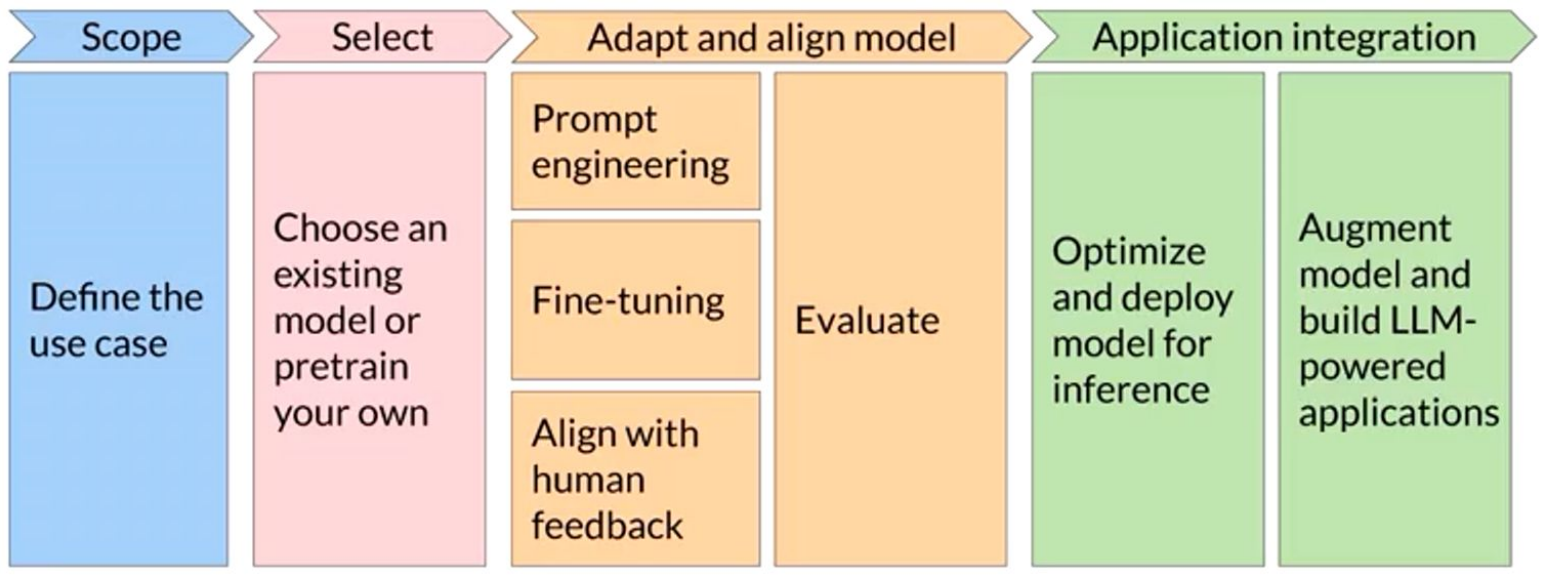
“As reasoning chains (cadeias de raciocínio) implementam transparência no processo de tomada de decisão dos LLMs, forçando o modelo a explicitar cada etapa do seu raciocínio antes de chegar a uma conclusão final.

Decompõem problemas complexos em etapas intermediárias para melhorar a precisão e a transparência de respostas de IA.”



Fonte: medium

Ciclo de vida do projeto



Perguntas?