

# Laboratório – GenAI



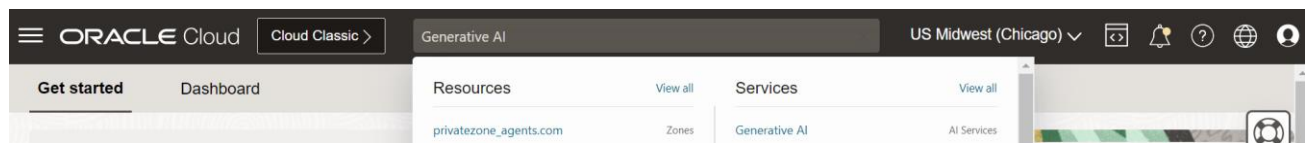
## INTRODUÇÃO

Neste laboratório, vamos explorar diferentes aspectos associados à LLMs e IA Generativa. Ele será dividido em 3 etapas, cada uma delas referente à um tema específico. Serão eles: Embeddings, Geração de Texto, Simulação de fluxo de RAG.

## ETAPA 1 – MODELOS DE EMBEDDINGS

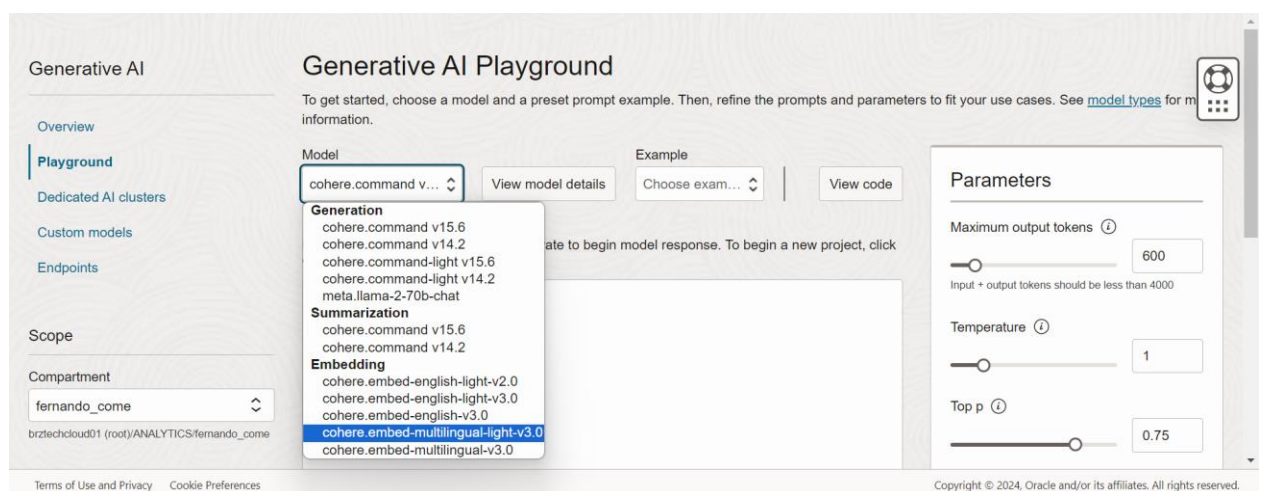
*Embeddings* é o nome dado à representação vetorial de um objeto. Isto é extremamente importante pois ao transformarmos estes objetos em vetores podemos efetuar operações matemáticas e cálculos com estes vetores, e assim realizar operações como análises de similaridade entre coisas que em princípio não seriam possíveis de se avaliar utilizando algoritmos. Nesta etapa, vamos explorar os modelos de Embeddings para texto disponíveis no serviço de GenAI do OCI.

Passo 1. Acessar o Serviço de OCI Generative AI. A forma mais simples de fazer isto é pesquisando por “Generative AI” na aba de busca:



Passo 2. Uma vez dentro do serviço, vamos selecionar “Playground”, no menu do canto esquerdo, abaixo de “Overview”.

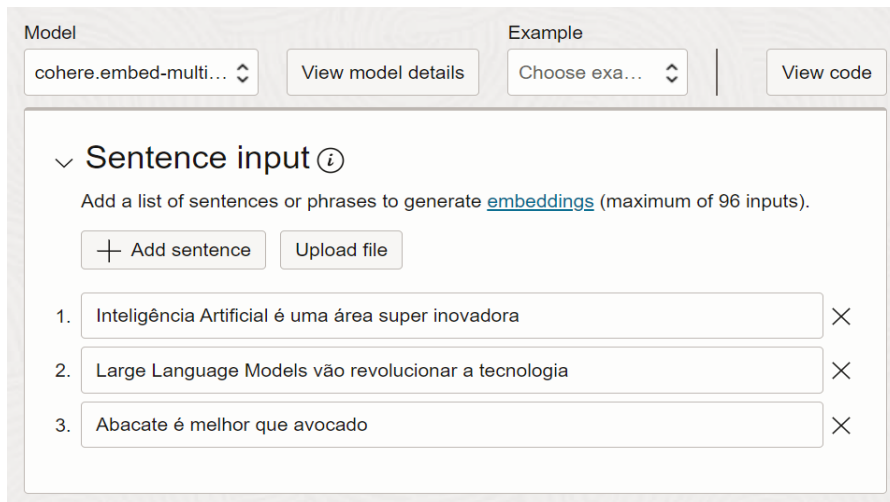
Passo 3. Dentro do PlayGround, vamos na caixa de seleção “model” e vamos selecionar o modelo **cohere.embed-multilingual-light-v3** ou **cohere.embed-multilingual-v3**.



Passo 4. Com o modelo devidamente escolhido, vamos então adicionar as frases que queremos comparar as representações de Embeddings. Basta inserir uma de cada vez, clicando em “Add Sentence” para incluir uma nova frase. Alguns exemplos interessantes de frases são:

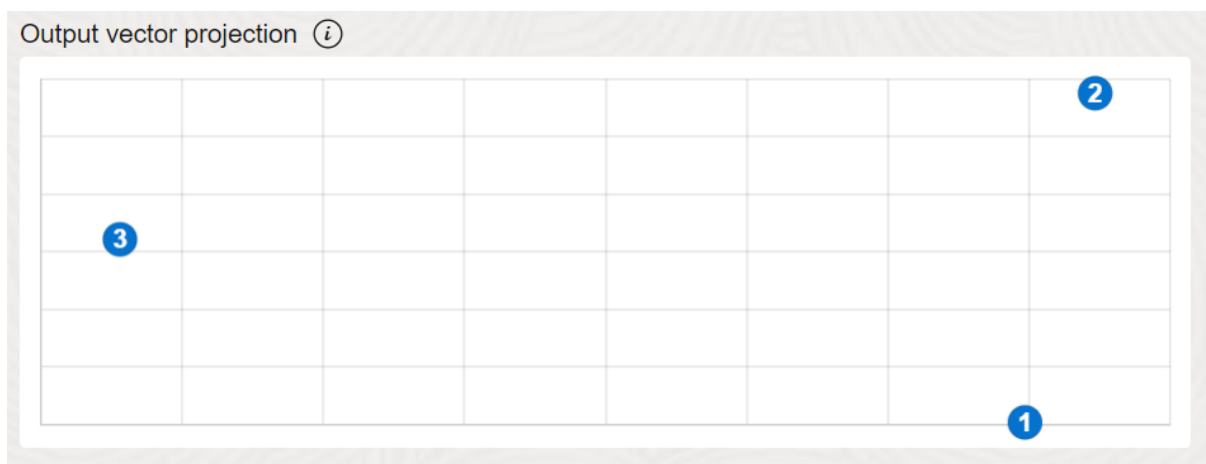
- Cachorros são animais incríveis
- Eu amo cães, são fantásticos
- A Porsche faz carros belíssimos
- Inteligência Artificial é uma área super inovadora
- Large Language Models vão revolucionar a tecnologia
- Abacate é melhor que avocado

Após a inclusão das frases, devemos ter algo parecido com:



Feito isso, basta clicar em “Run”.

Passo 5. Avaliar os resultados. Os vetores de embeddings costumam ter muitas dimensões (em geral, entre 512 e 1024 dimensões). Como é impossível visualizar graficamente algo com tantas dimensões, o que costuma ser feito é uma “Projeção” destes vetores multidimensionais em superfícies bi-dimensionais, permitindo a visualização. A imagem resultante é:



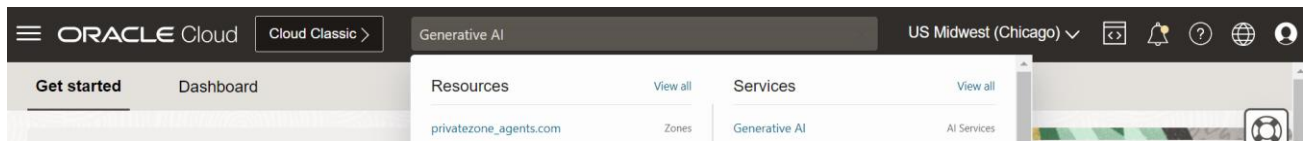
É interessante ver como as duas frases semelhantes (1 e 2) estão localizadas próximas, enquanto a frase destoante (3) está bem distante das outras duas, mostrando que de fato frases semanticamente distantes também são representadas de forma distante.

Para saber mais sobre modelos de embeddings, acesse o link: <https://txt.cohere.com/sentence-word-embeddings/>

## ETAPA 2 – MODELOS DE GERAÇÃO DE TEXTO

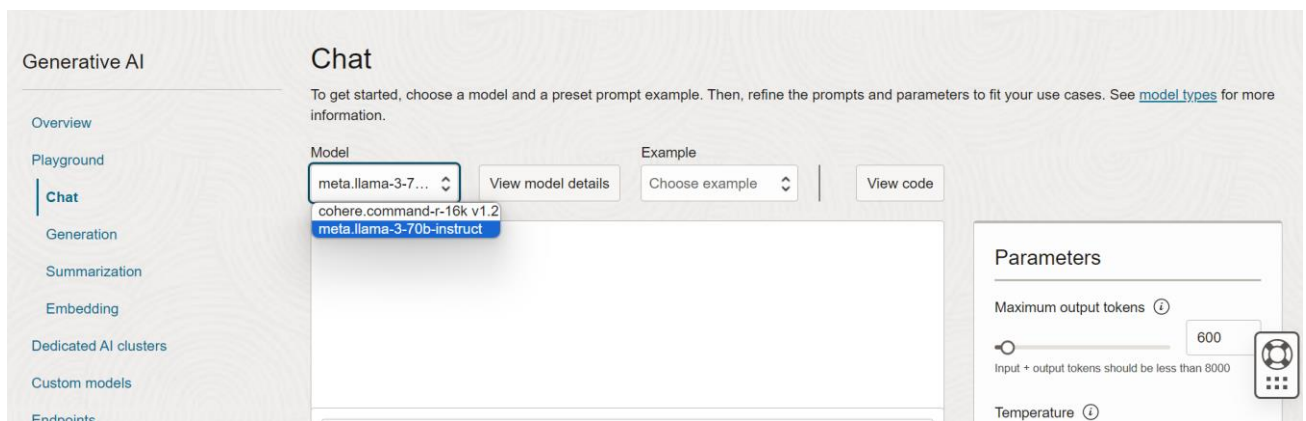
Nesta etapa, vamos estudar o comportamento do modelo para geração textual e em especial a influência do parâmetro Temperatura nos resultados. Na geração de texto, há diversos parâmetros que moldam como será feita a geração de cada palavra (ou token, para ser mais exato). Para isto, vamos testar diferentes cenários de geração textual com diferentes combinações de parâmetros.

Passo 1. Acessar o Serviço de OCI Generative AI. A forma mais simples de fazer isto é pesquisando por “Generative AI” na aba de busca:



Passo 2. Uma vez dentro do serviço, vamos selecionar “Playground”, no menu do canto esquerdo, abaixo de “Overview”.

Passo 3. Dentro do PlayGround, vamos clicar seleção “chat” e vamos selecionar o modelo **meta.llama-3-70b-instruct**.



Passo 4. Vamos testar inicialmente os prompts de exemplo que o serviço nos oferece. Para isso, vamos selecionar na aba “Example” e selecionar a opção “Generate a product pitch” (mas fique a vontade para escolher as outras opções também).

The screenshot shows the Llama Playground interface. At the top, there's a 'Model' section with a dropdown menu showing 'meta.llama-3-7...' and a 'View model details' button. To the right is an 'Example' section with a dropdown menu showing 'Generate a prod...' and a 'View code' button. The 'Example' dropdown menu is open, displaying a list of options: 'Choose example', 'Generate a job description', 'Generate a product pitch' (which is highlighted in blue), 'Generate an email', 'Rewrite instructions with steps', and 'Summarize a blog post'. Below the dropdowns is a large text input area. At the bottom of this area, there's a text box containing the prompt: 'Generate a product pitch for a USB connected compact microphone that can record surround sound. The microphone is most useful in recording music or conversations. The'. Below the text box are two buttons: 'Submit' and 'Clear chat'.

Passo 5. Ao selecionar a opção desejada, um prompt de exemplo será automaticamente gerado (porém em Inglês). Para efeitos didáticos, podemos usar em Inglês sem problemas, mas caso desejemos um resultado em Português, podemos traduzir o prompt gerado e, como a família de modelos Llama (bem como o Cohere) foi originalmente treinado para língua inglesa, é interessante incluir no prompt uma instrução como “Responda somente em Português PT-BR”. Uma vez que estamos satisfeitos com o prompt, basta clicar em “Generate”.

The screenshot shows the Llama Playground interface after the prompt has been translated into Portuguese. The 'Model' section remains the same. The 'Example' dropdown menu is still open, showing 'Generate a prod...'. The large text input area now contains the translated prompt: 'Por favor, gere uma descrição para o seguinte produto: um microfone e fone de ouvido compacto, com conector via USB, que permite captar os sons dos arredores em alta qualidade cabendo no bolso, sendo muito útil tanto para ouvir músicas quanto para gravar ideias. Responda com atenção pois isto é muito importante para minha carreira.' Below the text box are two buttons: 'Submit' and 'Clear chat'.

Inicialmente, vamos manter os parâmetros padrões. Uma vez gerada a resposta, vamos repetir com exatamente o mesmo prompt, mas aumentando a temperatura para 2. O que mudou na geração da descrição? E agora, trocando o parâmetro “k” de -1 para 2, o que acontece?

Para mais informações referentes aos parâmetros de um LLM:

Temperatura: <https://docs.cohere.com/docs/temperature>

Top-K e Top-P: <https://docs.cohere.com/docs/controlling-generation-with-top-k-top-p>

Demais parâmetros: <https://txt.cohere.com/llm-parameters-best-outputs-language-ai/>

### ETAPA 3 – SIMULANDO UM FLUXO DE RAG

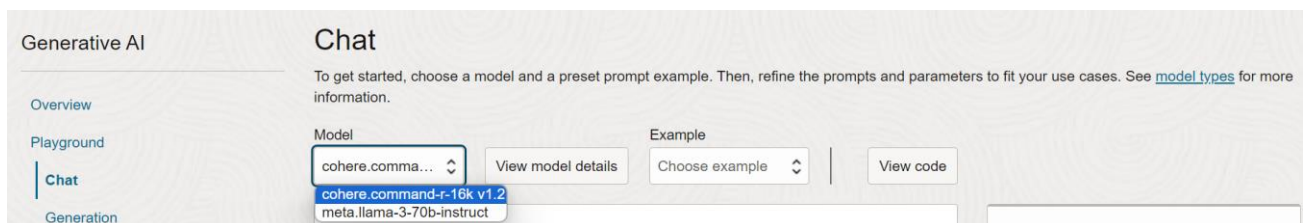
Nesta etapa, vamos simular no Playground um fluxo de RAG, ou *Retrieval-Augmented Generation*. Esta é uma técnica extremamente poderosa para aplicarmos modelos generativos sobre domínios específicos de conhecimento, mas sem a necessidade de *fine-tune*. Em termos de aplicações empresariais e *enterprise*, soluções baseadas em conceitos de RAG se mostram muito eficientes pois reduzem a quantidade de halucinações e informações incorretas que um modelo generativo pode retornar. Como a arquitetura de um sistema completo de RAG é razoavelmente extensa (e pode ser um tanto complexa), vamos simular um fluxo de RAG através da criação de prompts.

Passo 1. Acessar o Serviço de OCI Generative AI. A forma mais simples de fazer isto é pesquisando por “Generative AI” na aba de busca:



Passo 2. Uma vez dentro do serviço, vamos selecionar “Playground”, no menu do canto esquerdo, abaixo de “Overview”.

Passo 3. Dentro do PlayGround, vamos na caixa de seleção “model” e vamos selecionar o modelo **cohere.command-r-16k v1.2**.



Passo 4. Para RAG, como temos um contexto fixo e delimitado, em geral não é desejável que o modelo retorne informações que não estejam presentes neste contexto. Portanto, vamos selecionar a seguinte combinação de parâmetros: reduzir a Temperatura para 0.1, Aumentar o Top-P para 0.95 e Aumentar o Top-K para 20 (mas sintase a vontade para explorar variações nestes parâmetros). Em geral, soluções de RAG adotam combinações de parâmetros mais conservadoras.

Passo 5. Vamos então montar o prompt. Para um bom resultado, nosso prompt precisa conter algumas informações: uma persona, descrição da tarefa a ser executada, instrução e formatação da resposta, contexto e pergunta. Em uma aplicação real, o contexto é determinado de forma automática pelo sistema de recuperação de informação, mas aqui forneceremos um exemplo de contexto extraído de documentações de serviços de IA da Oracle. Um exemplo de prompt seria:

Você é um especialista em Inteligência Artificial, e deve responder perguntas sobre dois dos serviços oferecidos pela Oracle, o OCI Speech e OCI Language. Resposta somente em Português PT-BR e de forma direta e resumida. Construa a resposta somente baseado no contexto fornecido. Se não for possível construir uma resposta, não tente inventar informações que não estejam fornecidas no contexto. Responda com atenção pois isto é muito importante para a minha carreira.

Pergunta: (Adicione sua pergunta aqui)

Contexto: (cole o contexto aqui)

Para o contexto, podemos usar o seguinte exemplo:

O OCI Speech suporta 12 formatos de áudio, incluindo o formato OGG (formato de áudio do WhatsApp), além dos mais comuns como MP3 e WAV. Suporta também vídeos em formato MP4.

O Speech suporta 10 idiomas diferentes, incluindo 4 tipos de Inglês (americano, britânico, australiano e indiano), além de Português, Espanhol, Alemão e outros. A transcrição também inclui pontuação e pode ser feita também em formato SRT.

O OCI Language é um serviço gerenciado de inteligência artificial com foco nas atividades de análise de textos e processamento de linguagem natural. Um ponto importante: o Language não é uma ferramenta de IA Generativa. Seu alvo é realizar análises extrativas em cima de textos.

Nativamente, os modelos pré-treinados do OCI Language são capazes de realizar as seguintes tarefas: Classificação de textos em centenas de categorias; Detecção do Idioma com dezenas de opções; Extração de dezenas de Entidades Nomeadas diferentes; Extração de frases-chave; Análise e Detecção de sentimentos; Detecção e mascaramento de dezenas de informações pessoais; Tradução com suporte para diversos idiomas.

Algumas opções interessantes de pergunta:

- Quais idiomas o OCI Speech suporta?
- Quais funcionalidades o OCI Language oferece?
- Quais formatos de áudio o OCI Speech suporta?

Sendo assim, um exemplo completo de prompt seria:

Você é um especialista em Inteligência Artificial, e deve responder perguntas sobre dois dos serviços oferecidos pela Oracle, o OCI Speech e OCI Language. Resposta somente em Português PT-BR e de forma direta e resumida. Construa a resposta somente baseado no contexto fornecido. Se não for possível construir uma resposta, não tente inventar informações que não estejam fornecidas no contexto. Responda com atenção pois isto é muito importante para a minha carreira. Reforçando, responda somente em Português PTBR.

Contexto: O OCI Speech suporta 12 formatos de áudio, incluindo o formato OGG (formato de áudio do WhatsApp), além dos mais comuns como MP3 e WAV. Suporta também vídeos em formato MP4.

O Speech suporta 10 idiomas diferentes, incluindo 4 tipos de Inglês (americano, britânico, australiano e indiano), além de Português, Espanhol, Alemão e outros. A transcrição também inclui pontuação e pode ser feita também em formato SRT.

O OCI Language é um serviço gerenciado de inteligência artificial com foco nas atividades de análise de textos e processamento de linguagem natural. Um ponto importante: o Language não é uma ferramenta de IA Generativa. Seu alvo é realizar análises extrativas em cima de textos.

Nativamente, os modelos pré-treinados do OCI Language são capazes de realizar as seguintes tarefas: Classificação de textos em centenas de categorias; Detecção do Idioma com dezenas de opções; Extração de dezenas de Entidades Nomeadas diferentes; Extração de frases-chave; Análise e Detecção de sentimentos; Detecção e mascaramento de dezenas de informações pessoais; Tradução com suporte para diversos idiomas.

Pergunta: Quais funcionalidades o OCI Language suporta?

É muito interessante fazer o teste da pergunta com e sem o contexto fornecido, e avaliar o comportamento do modelo para cada exemplo.