

**Tech Challenge 3**

**IA para Devs**

**Daniela Cruz de Malta - RM353365**

**Gabriela Maciel Godoi - RM355125**

**Lucas Sutelo - RM353721**

**Sumário**

[**Caso de Estudo 3**](#_heading=h.nustgvr5ax4f)

[**Metodologia 3**](#_heading=h.nustgvr5ax4f)

[1. Preparação do Dataset 3](#_heading=h.3c8pxhfgxk4h)

[2. Fine-tuning com a OpenAI 3](#_heading=h.gszsooihwhyw)

[3. Instalação de Biblioteca Necessárias para LLAMA 4](#_heading=h.yws6ghm3o5ej)

[2. Configuração do Ambiente para LLAMA 4](#_heading=h.qq3zv88ieajs)

[4. Chamada do Foundation Model LLAMA 4](#_heading=h.r1ycc7615lrn)

[5. Processo de Fine-Tuning do LLAMA 4](#_heading=h.qo66ne1btjb6)

[**Resultados 5**](#_heading=h.20nkxu74e6a5)

[1. OpenAI 5](#_heading=h.a2wznqtm927r)

[2. LLAMA 5](#_heading=h.2cuylhw8fkl0)

[**Conclusão 6**](#_heading=h.c1br9swxhfw)

# Caso de Estudo

O objetivo deste Tech Challenge é realizar o fine-tuning de um foundation model utilizando a base de dados “The AmazonTitles-1.3MM” para receber perguntas dentro do contexto da base no Prompt de usuário e a resposta conter os aprendizados do fine-tuning.

Este documento detalha o processo de seleção e preparação da base de dados e descreve o processo de fine-tuning do modelo, com os parâmetros e ajustes utilizados.

O código fonte do processo de fine-tuning está no repositório do GitHub: <https://github.com/Danizinh/Fine-Tuning>.

# Metodologia

A equipe decidiu realizar o Fine-Tuning utilizando dois métodos, usando a biblioteca da OpenAI e importando o LLM LLAMA. A seguir está o procedimento realizado para os dois casos.

## 1. Preparação do Dataset

O dataset utilizado para o treinamento é o **AmazonTitles-1.3MM**, carregado a partir do arquivo **trn.json**, contendo títulos e descrições de produtos.

As colunas "title" e "content" foram preparadas para formar os *prompts* de entrada no modelo em um arquivo com extensão JSONL, que é necessária para realizar o Fine-Tuning tanto no LLAMA, como na OpenAI.

Foi necessário realizar uma quebra no dataset nos dois casos devido a limitação de recursos.

## 2. Fine-tuning com a OpenAI

A equipe gerou uma API key para utilizar a biblioteca OpenAI. O treinamento foi feito com uma base de dados limitada (50 linhas), devido ao alto custo para um treinamento com mais linhas. O dataset em questão é o arquivo ./data/dataset2.jsonl. Os testes foram realizados usando produtos disponíveis neste dataset.

O prompt de sistema informa claramente que o modelo é um assistente que deve fornecer a descrição de um produto inserido pelo usuário.

## 3. Instalação de Biblioteca Necessárias para LLAMA

* As bibliotecas *accelerate*, *peft*, *bitsandbytes*, *transformers* e *trl* são instaladas para fornecer as ferramentas necessárias para realizar o *fine-tuning* e a quantização de modelos.

## 2. Configuração do Ambiente para LLAMA

* O código importa PyTorch e outros módulos da biblioteca **Hugging Face** para manipulação de modelos e datasets.2. Processo de fine-tuning
* As classes **AutoModelForCausalLM**, **Auto Tokenizer**, e **SFTTrainer** são utilizadas para carregar o modelo base e realizar o *fine-tuning*.

## 4. Chamada do Foundation Model LLAMA

* O modelo base escolhido é o **Llama-2-7b-chat-hf**, carregado da plataforma **Hugging Face**.
* Um teste inicial é executado para medir a performance do modelo antes do *fine-tuning*.

## 5. Processo de Fine-Tuning do LLAMA

* O modelo é configurado para executar o *fine-tuning* utilizando a quantização em 4 bits através da configuração **BitsAndBytesConfig**.
* O processo de ajuste fino utiliza a técnica **LoRA** (*Low-Rank Adaptation*) para treinar o modelo em menor escala, economizando recursos computacionais.
* Parâmetros de treinamento: 15 épocas, tamanho de lote de 4, taxa de aprendizagem 0.001 e otimizador *Paged AdamW*.

# Resultados

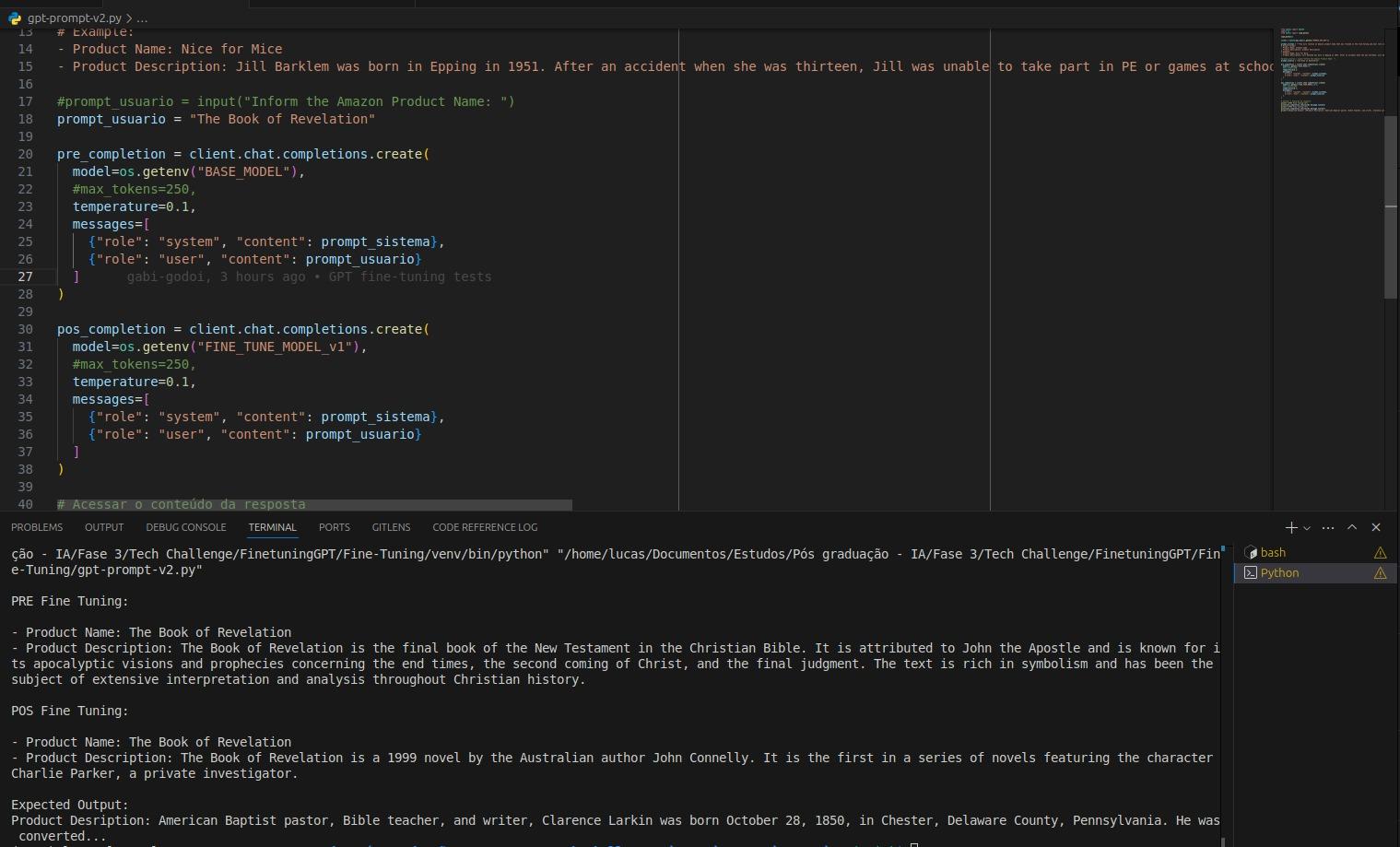
## 1. OpenAI

O algoritmo está atingindo o objetivo proposto de atuar como um assistente virtual que informa a descrição de produtos inseridos no prompt de usuário.

A descrição está de acordo com os produtos, porém, ela não é exatamente o conteúdo da base de dados treinada.

Na imagem a seguir, trazemos o exemplo do “The Book of Revelation”, com a resposta do modelo base (Pre Fine Tuning) , resposta do modelo treinado (Pos Fine Tuning) e a descrição fornecida na base de dados.

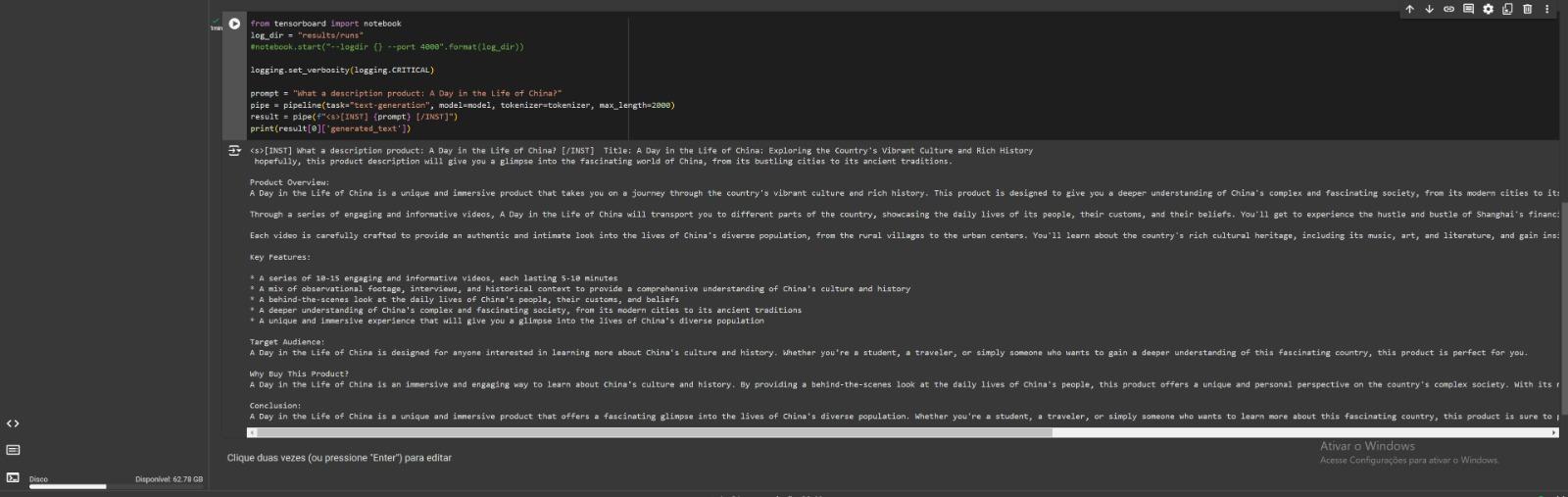
Veja que na base de dados a descrição do produto inicia com a descrição do autor, enquanto que, no modelo tanto pré como pós treino, a resposta traz apenas a descrição do produto.



## 2. LLAMA

O resultado foi muito similar ao caso da OpenAI, em que o algoritmo está atingindo o objetivo proposto de atuar como um assistente virtual que informa a descrição de produtos inseridos no prompt de usuário, porém, ela não é exatamente o conteúdo da base de dados treinada.

Na imagem abaixo é possível verificar o teste feito com a série “A day in the life of China”, em que é retornada a descrição detalhada do produto.



# Conclusão

O código cumpre o objetivo do **Tech Challenge** de realizar o *Fine-Tuning* de um modelo de linguagem grande utilizando o dataset **AmazonTitles-1.3MM**.

O fluxo de trabalho documentado garante que o modelo esteja preparado para responder perguntas baseadas em descrições de produtos, oferecendo uma base sólida para aplicações em processamento de linguagem natural no contexto de comércio eletrônico.

A equipe verificou, que mesmo após o Fine-Tuning ser realizado, o assistente retorna um resumo geral das descrições dos produtos contidos no dataset, e não a descrição exata. Por isso, para análise futura, o grupo pretende aplicar o modelo de **RAG** para testar se esse método gera descrições mais parecidas com o input direto do dataset.