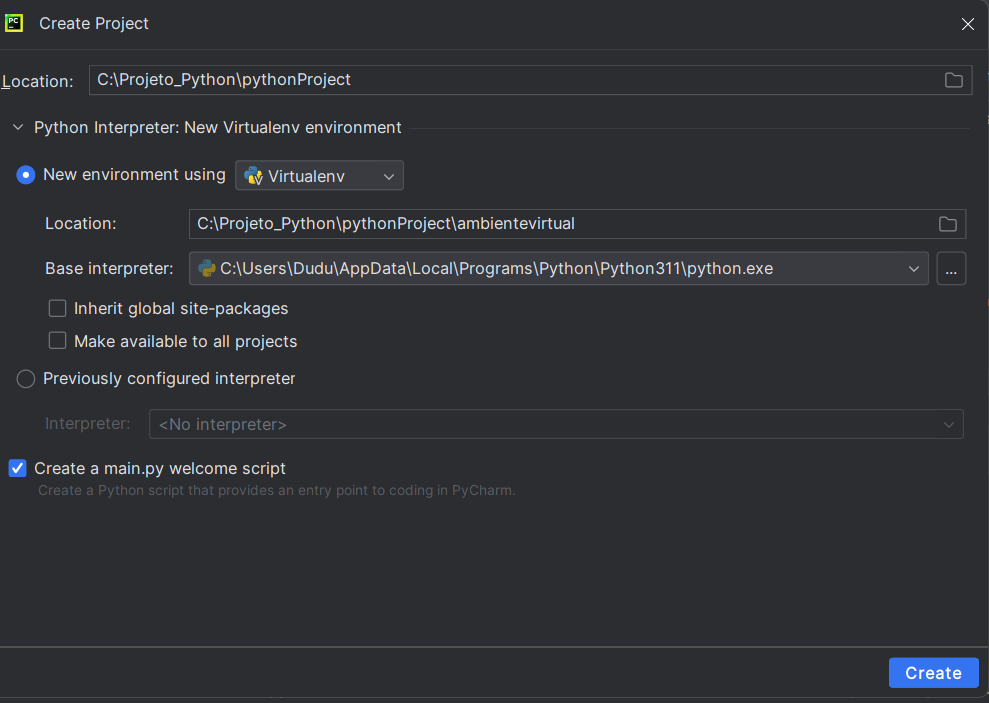
# Relatório da Construção de um chatbot baseado em IA para responder dúvidas sobre o vestibular da Unicamp 2024.

## Configuração:

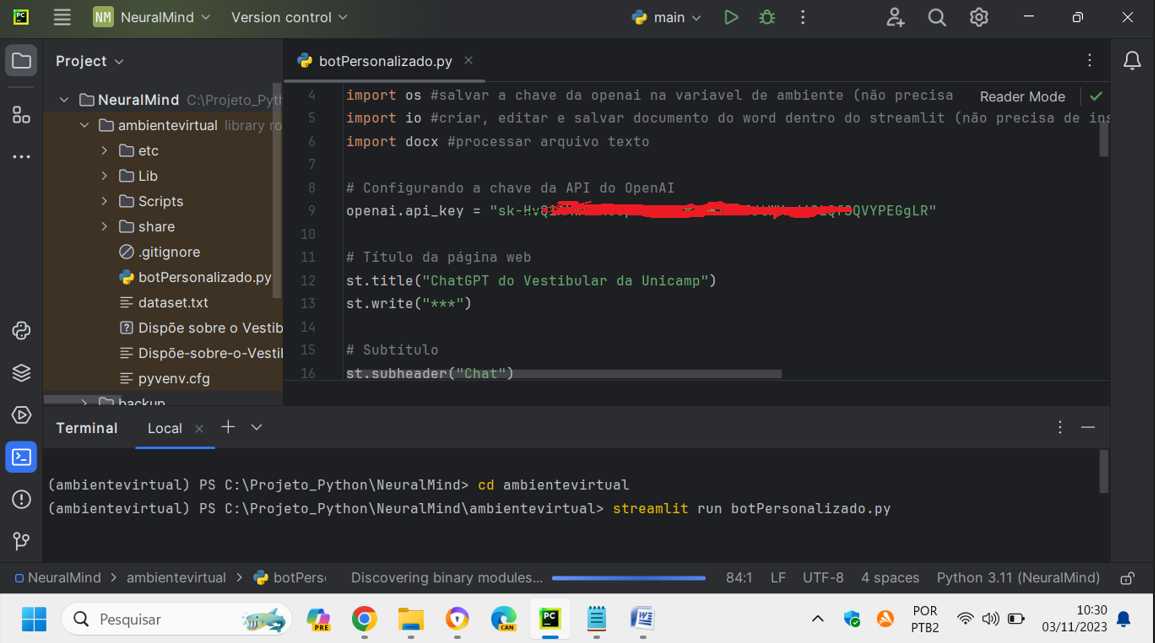
* Instalação Python versão 3.11 (opção path): https://www.python.org/downloads/windows/.
* Instalação da IDE pyCharm community, mas pode ser qualquer editor de texto.
* Criação do ambiente virtual. Obs.: Necessário ir até a pasta que criou o projeto.
  + No terminal executar o comando: pip install virtualenv
  + Informe um nome para o ambiente virtual através do comando: virtualenv NomeDoAmbiente.
  + Acesse o ambiente virtual pelo comando: cd NomeDoAmbiente.
  + Comando para ativa o Ambiente Virtual: Scripts\activate
* Instalação dos pacotes para rodar o Projeto
  + Pacote Streamlit: Serve para criar interface/aplicativos web
  + Pacote streamlit\_chat: Para dar um efeito de chat no aplicativo
  + Pacote openAI: Pacote para configurar a chave da OpenaAI.
  + Pacote docx (comando da versão compatível: pip install python-docx==0.8.11): Server trabalhar com arquivos texto.
* Quando cria um novo projeto no pycharm a instalação e configuração do ambiente virtual é criado automaticamente (sem necessidade dos passos acima), bastei apenas selecionar a pasta que desejei criar o projeto em Location e dar o nome do ambiente virtual. Conforme imagem abaixo:



* É necessário colocar o arquivo dataset dentro da pasta do ambiente virtual, ou seja, no junto com arquivo .py.

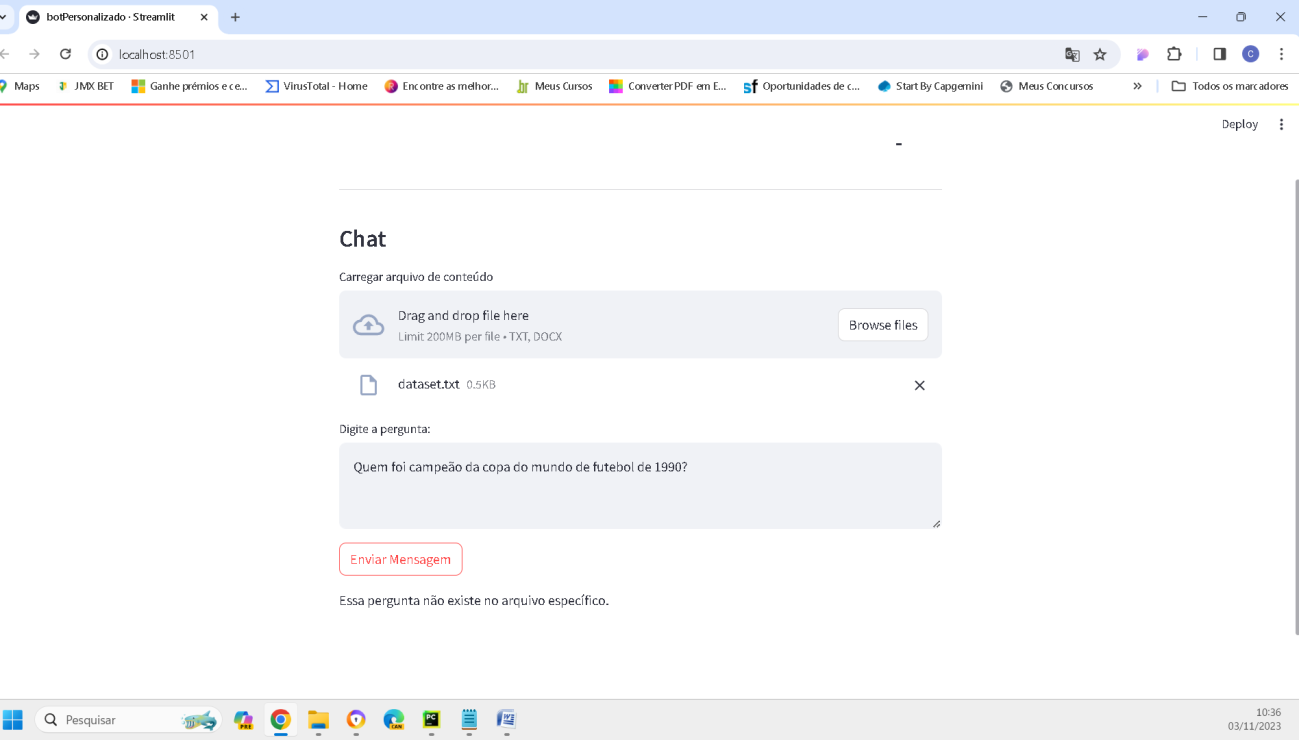
## Demonstração:

* Comando para executar o aplicativo: streamlit run nomedoarquivo.py
  + Obs.: Antes de rodar verifique se estar no diretório que se encontra no arquivo .py

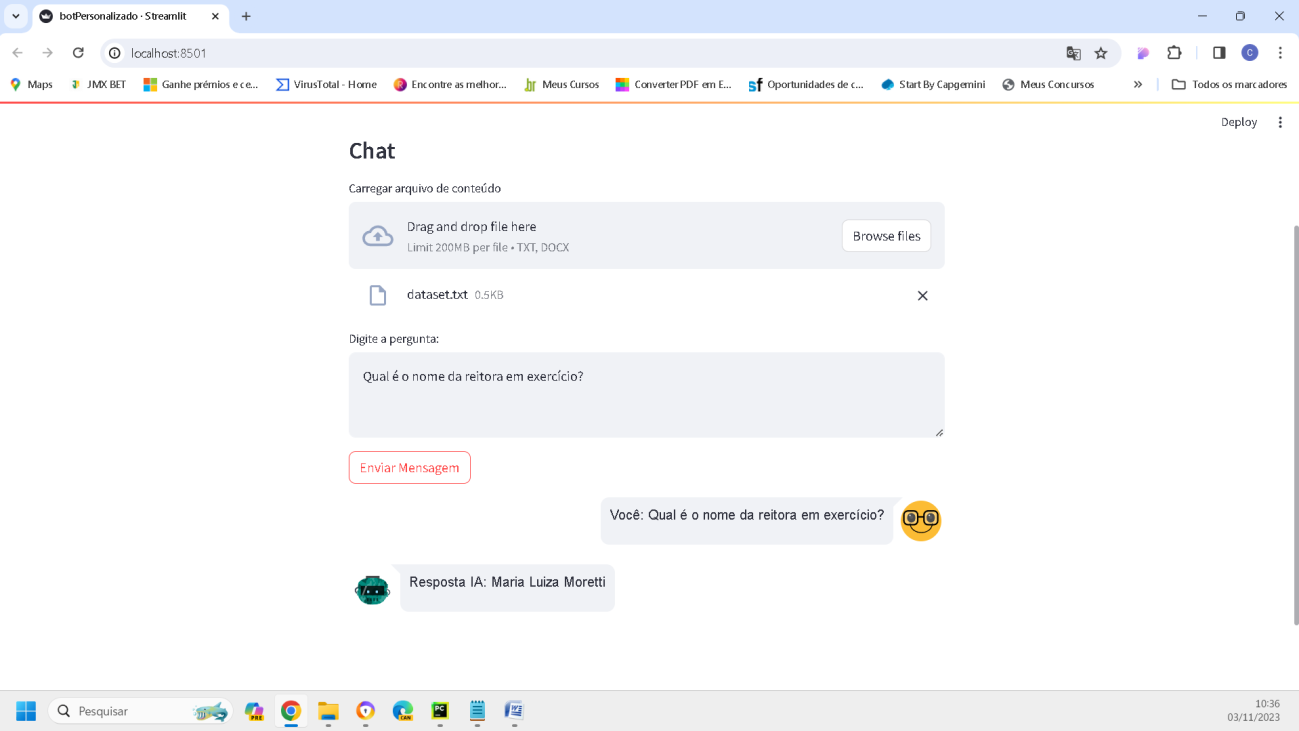


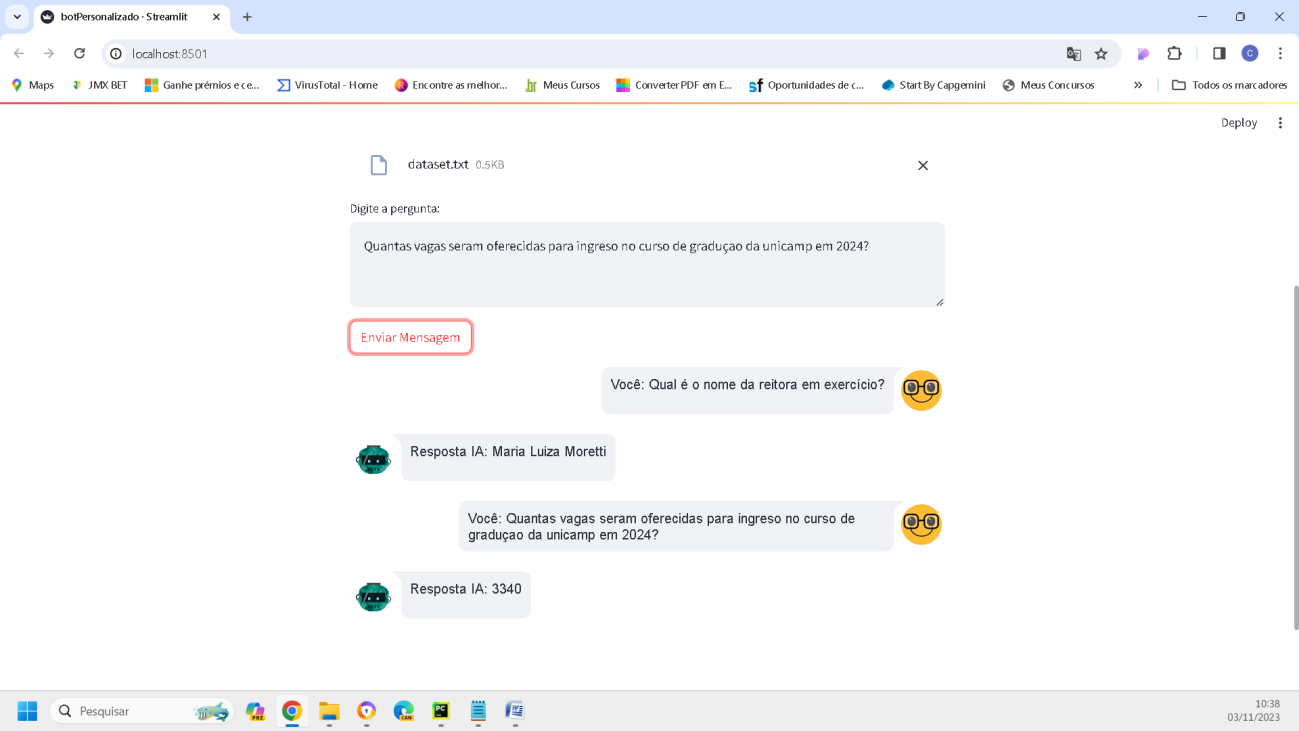


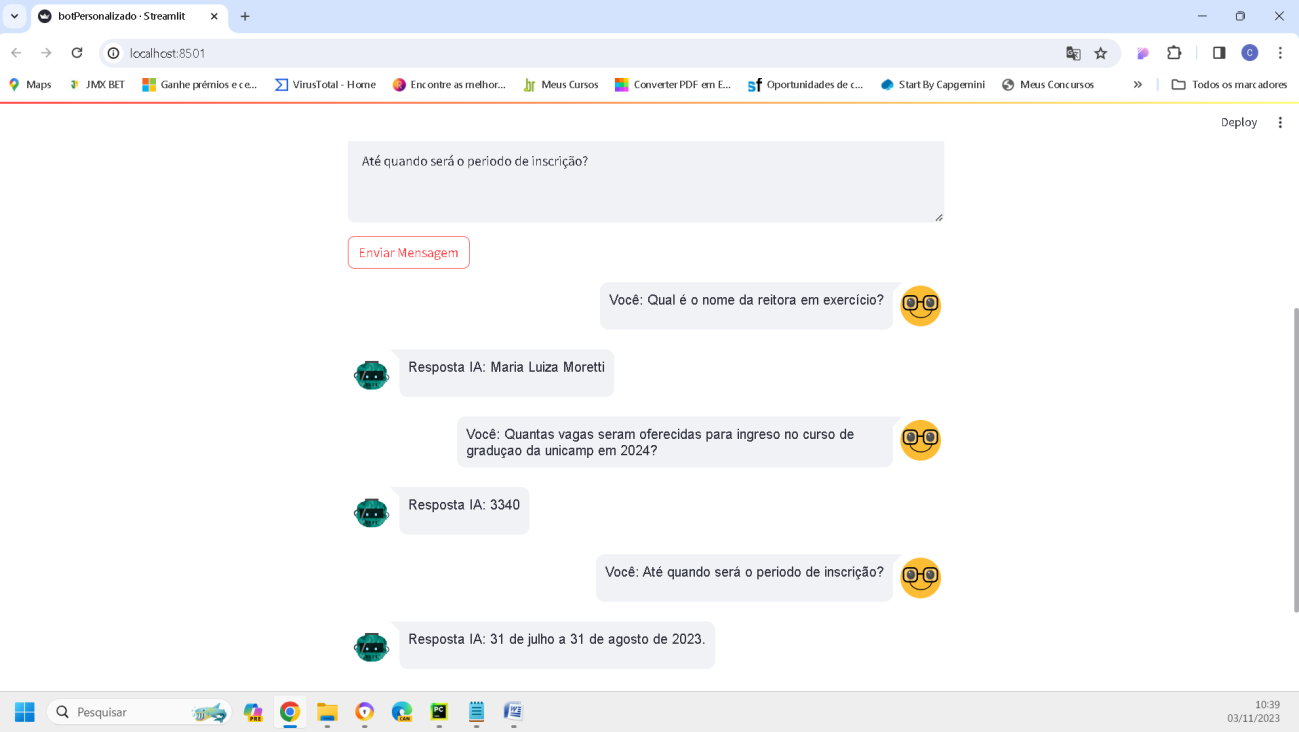
* Carreguei o documento dataset.txt e fiz uma pergunta que não estava no conteúdo do arquivo. Foi apresentado que “Essa pergunta não existe no arquivo específico”.

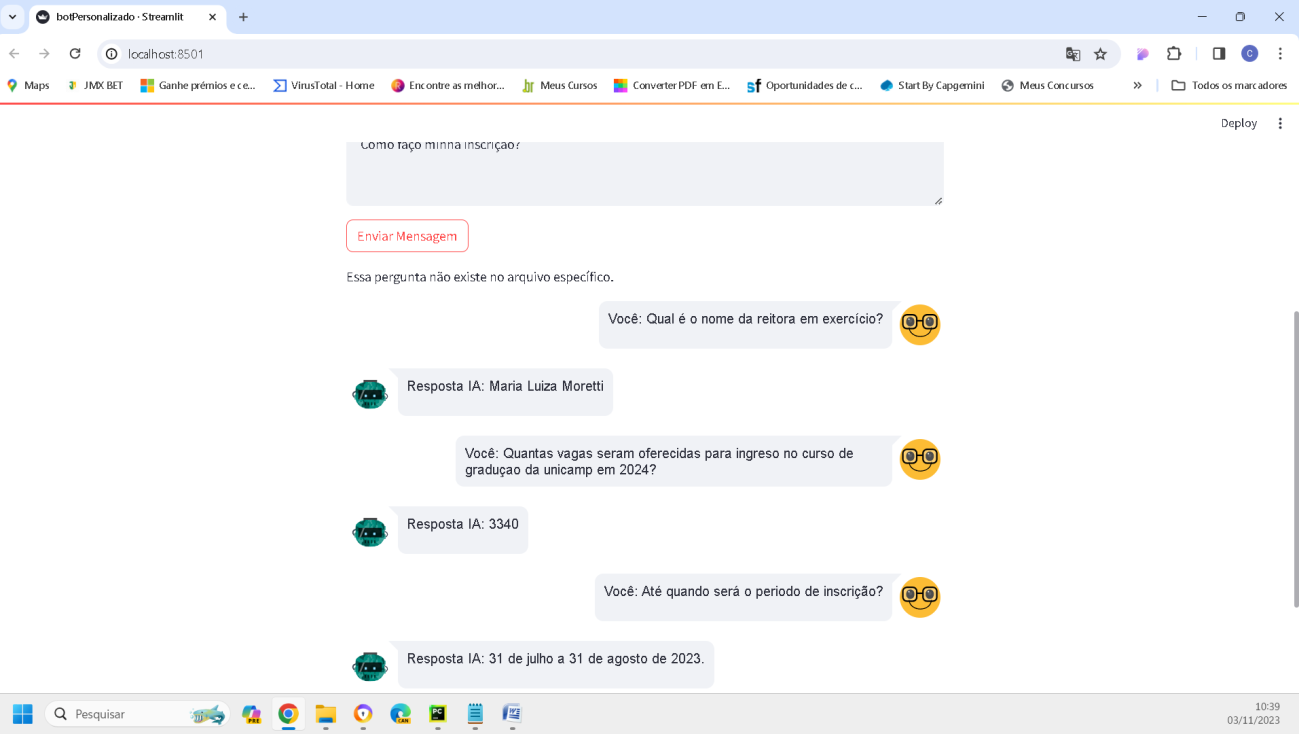


* As imagens abaixo são uma sequencia de perguntas feitas que existe dentro conteúdo do arquivo (dataset.txt). Obs.: Para garantir que o chat da OpenAI responda apenas a perguntas com base no conteúdo do arquivo específico, foi necessário treinar o modelo com esse conteúdo específico e fornecer instruções claras para que ele não respondesse perguntas fora desse escopo.
* Desta forma construir um modelo de dataset de avaliação que foi usado para verificar o conteúdo do arquivo. Criei um arquivo de texto (dataset.txt) que contenha pares de pergunta e resposta no seguinte formato:
  + P: Pergunta?
  + R: resposta.

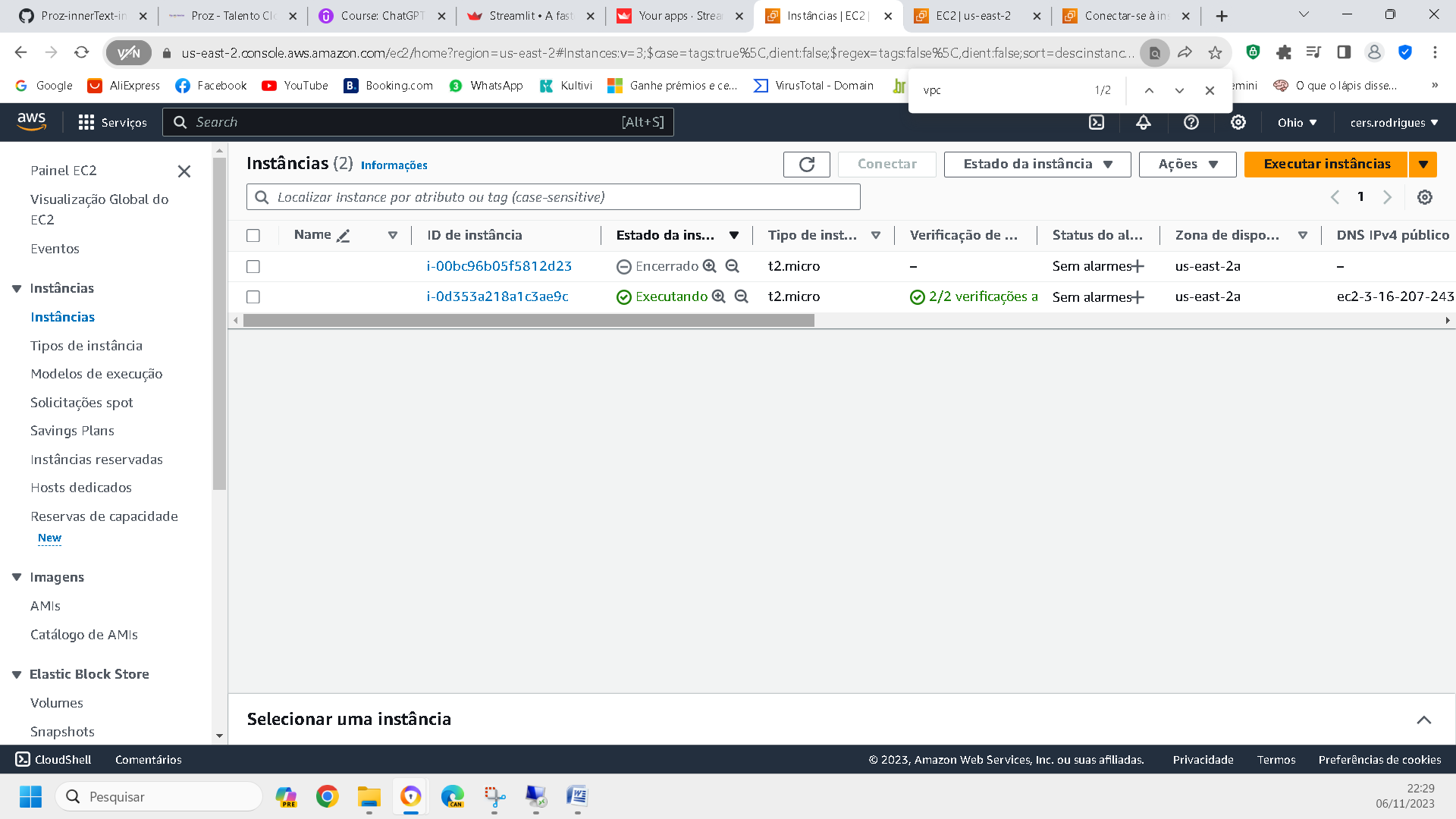


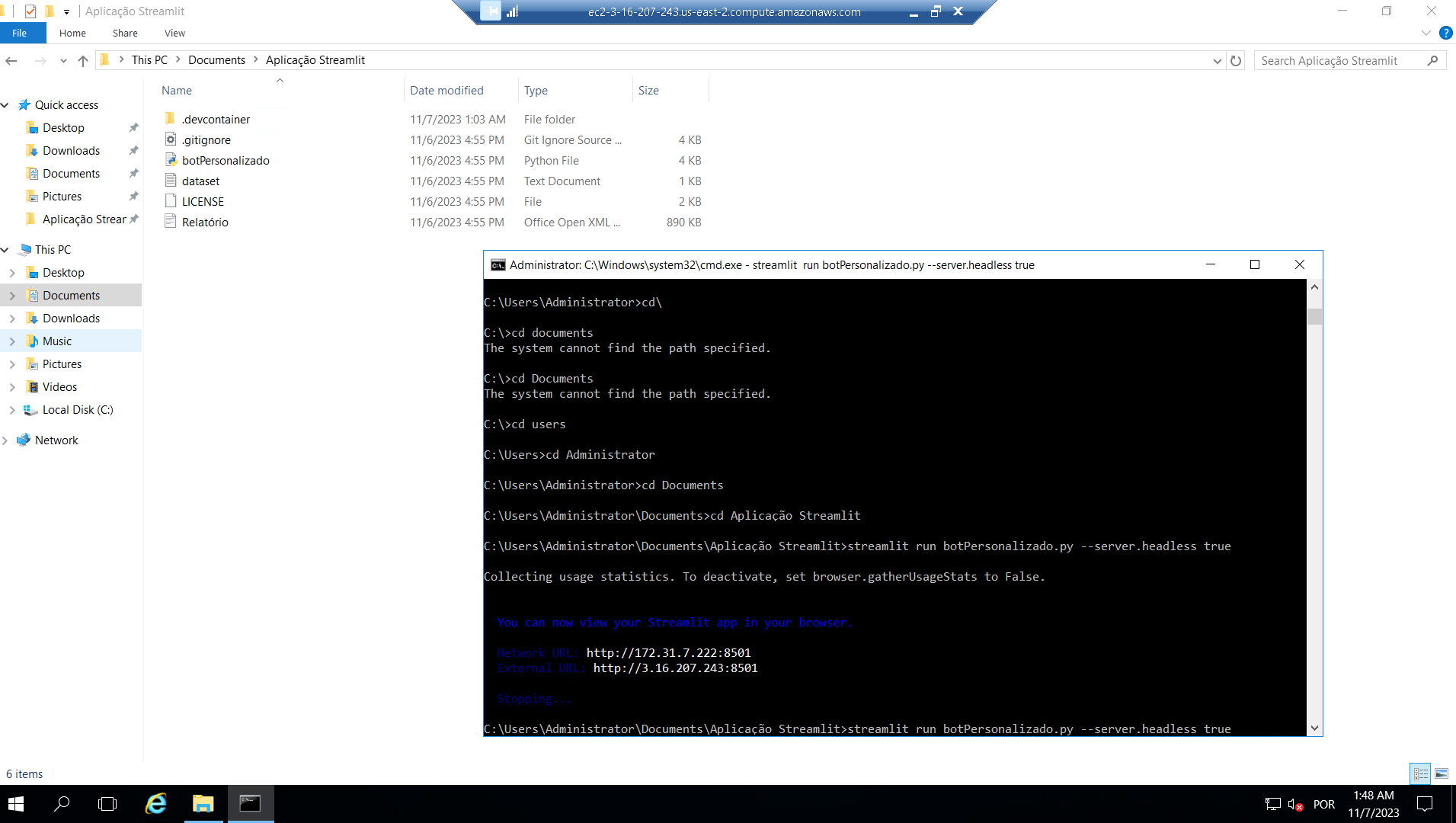




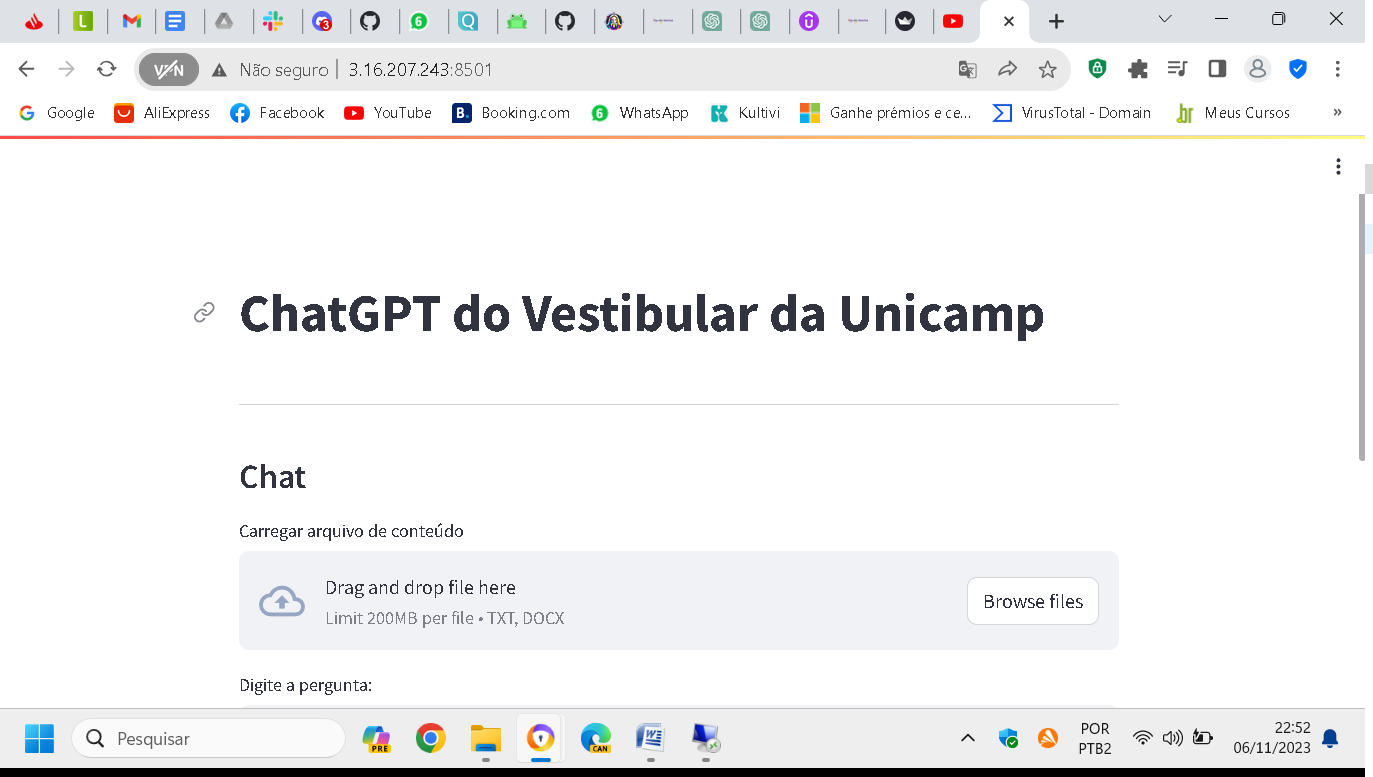


## Deploy – Streamlit (Python com AWS:

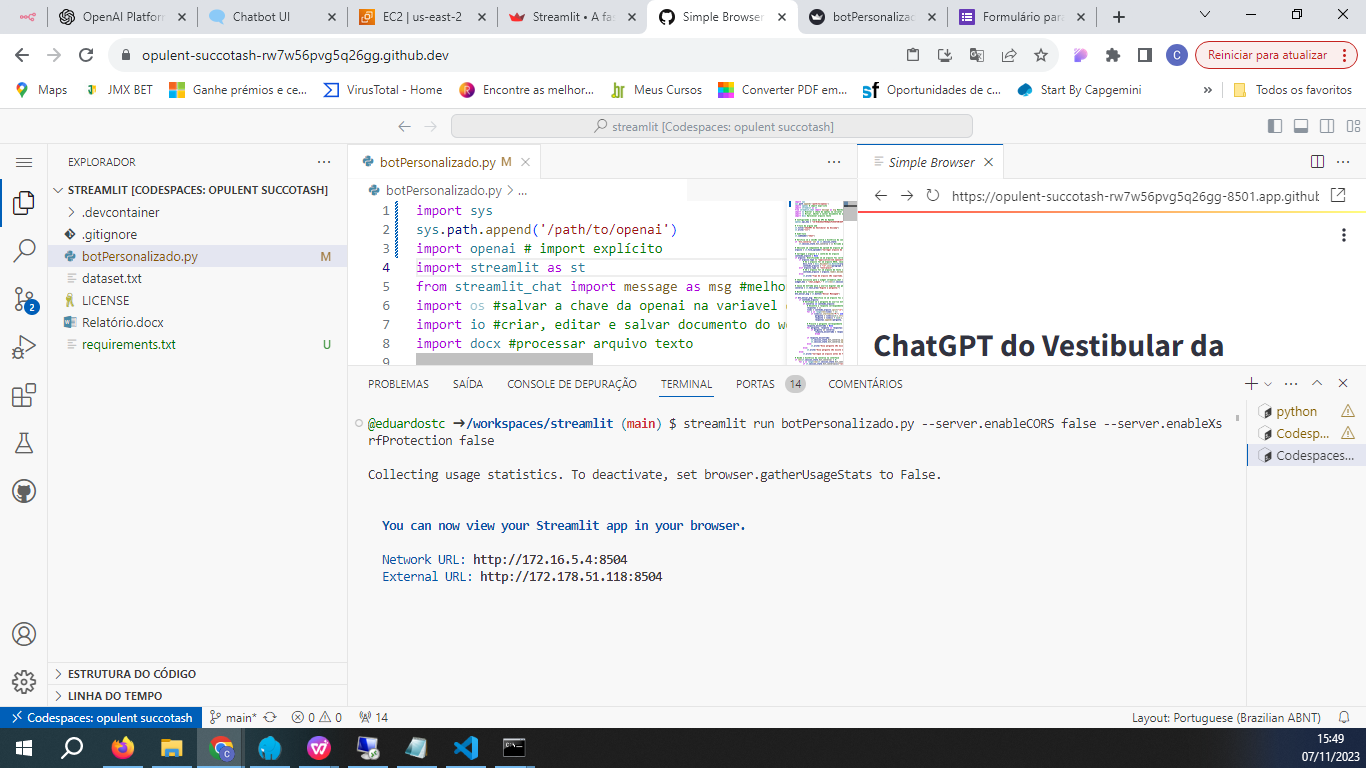
* Criado uma conta na AWS.
* Foi criado uma instancia do Windows Server pelo serviço ECS2, conforme imagem abaixo:
* Após instalado o python versão 3.11.
* No terminal foi necessário instalar os pacotes cidados a cima e executar o comando no “streamlit run botPersonalizado.py --server.headless true” no diretório que se encontra o arquivo da aplicação no servidor.

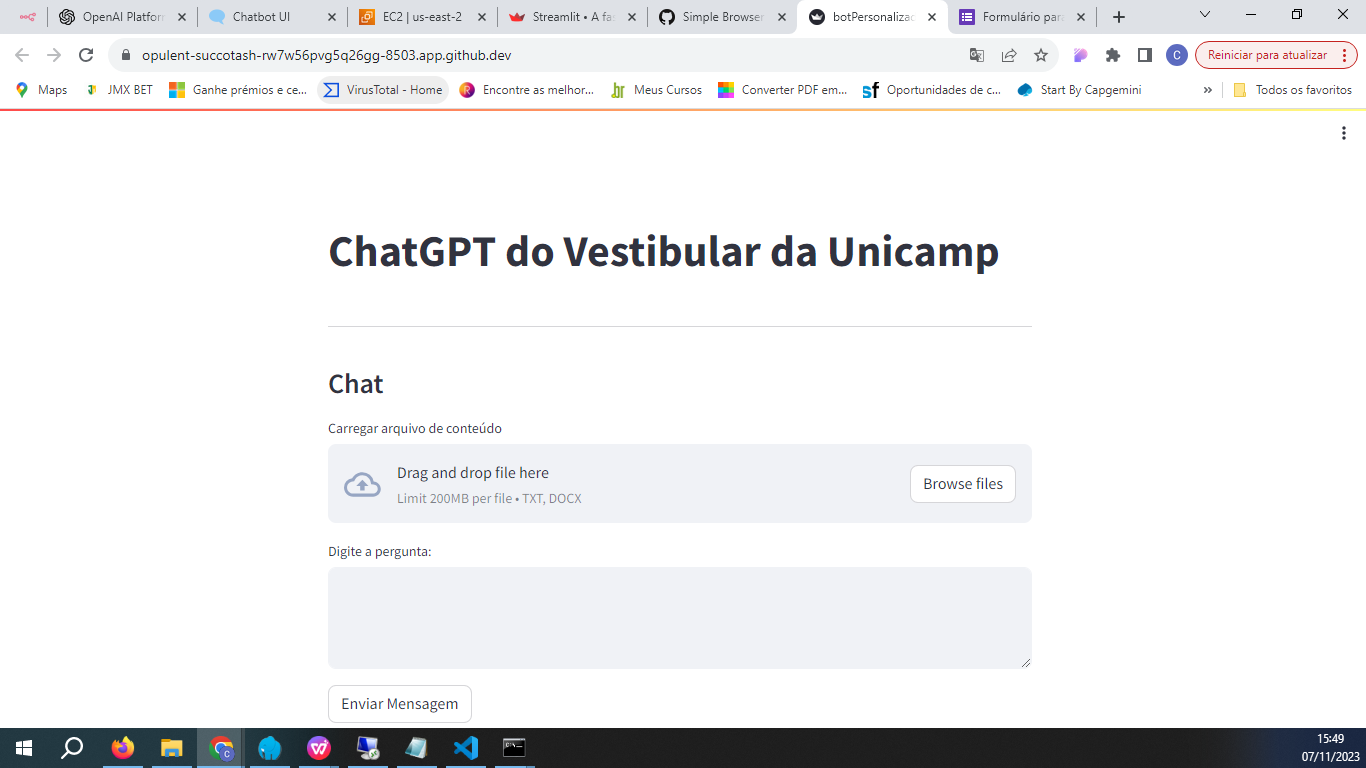


* Por fim foi configurado o VPC e suas portas nas regras de entrada do aws e no firewall do servidor remoto.
* A aplicação ficou disponível no endereço: <http://3.16.207.243:8501/>

[](http://3.16.207.243:8501/)

## Deploy no Streamlit:





Link externo: https://opulent-succotash-rw7w56pvg5q26gg-8503.app.github.dev/

## Fontes:

* [https://www.udemy.com/course/chatgpt-35turbo-domine-a-novidade-da-openai-de-01032023/learn/lecture/36685022?start=15#content](https://www.udemy.com/course/chatgpt-35turbo-domine-a-novidade-da-openai-de-01032023/learn/lecture/36685022?start=15" \l "content)
* <https://www.youtube.com/watch?v=2vb22jMwiiw>
* <https://www.youtube.com/watch?v=vw0I8i7QJRk&t=126s>