AliveBot: a chatbot for smart map exploration

David Scarin, up202108314
Coordenadores de estágio: Carlos Rebelo, Alexandra Oliveira

Março 2024 - Julho 2024

Contents

List of Figures

1 Abstract

O AliveBot é um projeto que pretende inovar na integração de Large Language Models(LLM) com a exploração de mapas interativos, com o objetivo de proporcionar uma nova experiência para utilizadores na área do turismo e exploração, focando-se na vertente do conhecimento histórico, geográfico e cultural.

Neste relatório exploro as funcionalidades do AliveBot e o processo do seu desenvolvimento. Começo por introduzir o projeto, explicando detalhadamente em que consiste e quais são os desafios propostos. De seguida, descrevo o trabalho realizado, a nível da metodologia e ferramentas utilizadas, explorando, de forma mais técnica, a sua implementação final. Termino com os resultados empíricos e experimentais dos testes realizados com o sistema e concluo com o que ainda pode ser inovado e implementado com base no trabalho realizado até ao momento.

Desenvolvi com sucesso uma aplicação web que implementa uma pipeline capaz de receber uma mensagem do utilizador, juntamente com outra informação relevante para o seu processamento, e receber a resposta de um Assistant configurado no âmbito das áreas do turismo e viagem; para além disso, a resposta do modelo é condicionada para fornecer dados necessários, tais como coordenadas, que são processados e utilizados no funcionamento do mapa interativo, que reage de acordo com a resposta do Assistant.

2 Introdução

Este projeto é referente à Unidade Curricular CC 3048. A primeira fase de execução consistiu na seleção de um projeto de entre um conjunto de propostas em que o aluno deveria contactar os coordenadores das propostas em que estava interessado. A execução do projeto/estágio decorreu entre 1 de março e 26 de junho.

O projeto descrito neste relatório é o "AliveBot" que consiste na criação de um chatbot para exploração interativa de um mapa. No âmbito da Inteligência Artificial Generativa, esta proposta visa a fusão do já conhecido modelo de chatbot, capaz de estabelecer conversas interativas com um ser humano em linguagem natural, com um sistema de navegação ou mapa, no sentido de ser um assistente capaz de auxiliar o utilizador com conhecimentos turísticos, históricos e culturais, tal como um guia.

O modelo de linguagem estará integrado com um sistema de mapa: o chat estabelece a tradicional conversa entre modelo e utilizador, fornecendo

respostas de texto com informação detalhada; ao mesmo tempo, o modelo tem a capacidade de interagir com o mapa em união com a sua resposta, por exemplo, destacando áreas, colocando pins e realçando as localizações relevantes do mapa de forma a orientar o utilizador.

O meu conjunto de tarefas diz respeito, nomeadamente, ao backend do sistema, isto é, implementar corretamente os componentes do chat: enviar a resposta do utilizador, receber a resposta do modelo e extrair de lá as informações necessárias, bem como configurar os parâmetros do modelo para obter a resposta no formato desejado. Todo o código e documentação referentes ao backend do projeto estão documentados num repositório github[4].

Nas secções seguintes, exploro de que forma abordo os principais desafios da implementação pretendida: estabelecer uma conversa entre utilizador e modelo; garantir que o modelo tem a capacidade de fornecer respostas adequadas e de qualidade neste contexto particular e conjugar o mapa interativo com uma resposta constituída unicamente de texto, extraindo deste texto coordenadas e outras informações necessárias. Ao mesmo tempo, o sistema deve ser aplicável como um produto comercial, o que significa considerações a nível dos seus custos de produção e de utilização, bem como a interação com o utilizador final.

3 Enquadramento / Estado do Arte

3.1 Enquadramento Científico

O projeto AliveBot apresenta-se como uma fusão inovadora de diferentes tecnologias já existentes. O seu desenvolvimento está baseado na utilização de várias ferramentas avançadas das áreas da Ciência de Computadores e Inteligência Artificial, para além da componente Interação Pessoa-Máquina. A componente fundamental do projeto reside na integração das diferentes ferramentas ao nosso dispor.

A nível do processamento de linguagem natural e geração de texto, existem disponíveis vários modelos, inseridos na modalidade open-source ou com acesso a uma API pay-as-you-go. Em particular, foi selecionada a utilização da API da OpenAI[3], que nos permite aceder a diferentes modelos GPT. Na secção seguinte, o seu funcionamento e seleção serão exploradas em maior detalhe.

Já a nível da componente da Interação Pessoa-Máquina, visto que pretendemos um sistema que seja acessível e intuitivo para todo o tipo de utilizadores, indepedentemente do seu conhecimento técnico, é necessário considerar também estes aspetos dda acessibilidade do sistema, para além

da sua funcionalidade. Como tal, aplicamos técnicas e testes de usabilidade, de forma a providenciar uma melhor experiência para o utilizador final.

3.2 Ferramentas Utilizadas

3.2.1 Backend - GPT-4

Na seleção de um modelo de linguagem, existiam diferentes parâmetros a considerar no sentido de maximizar a adequabilidade e qualidade das respostas do modelo enquanto, ao mesmo tempo, minimizando os custos associados e a complexidade do sistema.

A principal divisão pode ser traçado entre modelos open ou closed source. Os modelos open-source, para além de não terem um custo associado, são geralmente processados localmente, o que permite uma configuração mais aprofundada dos parâmetros do modelo. Em contrapartida, os modelos closed-source são geralmente acessíveis a partir de uma API, que tem um custo associado, sendo este geralmente um modelo de pagamento pay-as-you-go, em que cada prompt tem associada a ela um preço proporcional às capacidades do modelo.

Baseado na pesquisa efetuado e na discussão entre a equipa, decidimos que a API da OpenAI apresentava o equilíbrio desejado entre performence e custo. Para além de apresentar a qualidade de respostas desejadas, a funcionalidade de Assistants permitia configurar o modelo de acordo com o desejado para o projeto.

Assim, a componente do chat foi implementada usando a funcionalidade de Assistants, isto é, instâncias de um dado modelo GPT que podem ser configuradas pelo utilizador. Um Assistant pode ser configurado com um conjunto de instruções iniciais que indicam os parâmetros e o propósito do seu funcionamento, condicionando o modelo para fornecer respostas dentro do contexto pretendido. No caso do "AliveBot", foi criado um assistente do mesmo nome, com os seguintes parâmetros:

Name: AliveBot

Instructions: "you are a travel guide, an expert in geography and tourism, well suited to inform travellers on geographical, historical and cultural data. You are well versed in history and multiple cultures and will answer in any language the user utilizes to speak to you. You will only answer travel/tourism related questions or engage in normal conversation but will refuse to answer unrelated or random questions." – prompt inicial do sistema, delimita o seu comportamento e as suas respostas. Esta

prompt permite-nos controlar os temas, estilos e tom das respostas do modelo, definindo-as segundo os requerimentos específicos do projeto

Model: gpt-4 – especifica qual dos modelos de linguagem da OpenAI utilizar. Os modelos diferem na sua capacidade e qualidade de resposta, sendo que o gpt-4 é um dos mais complexos

Temperature: 1.0 – controla a estocasticidade, isto é, se o modelo se comporta de forma mais aleatória ou determinística. Uma temperatura baixa resulta em respostas determinísticas e repetitivas e uma temperatura alta resulta em respostas mais criativas e diversas. 1 é o valor por definição, que representa um equilíbrio entre estas duas vertentes.

Top P: 1.0 – controla a seleção de tokens, garantindo que só as tokens mais prováveis são consideradas na formação do output, levando a respostas mais coerentes e de maior qualidade. 1 é também o valor por definição.

Por meio do API da OpenAI, é possível enviar prompts para este Assistant e receber a sua resposta. O funcionamento da API, de forma a estabelecer uma conversa, baseia-se na criação e manipulação de threads; o utilizador adiciona uma mensagem à thread e executa uma run, em que o modelo adiciona à thread a sua resposta. Assim, dentro da mesma thread, o modelo tem o contexto não só da última mensagem do utilizador mas de toda a conversa. Para além disso, o sistema fica assim capacitado de criar várias conversas (conceptualmente correspondente a várias threads), em que um ou mais utilizadores podem tratar de diferentes tópicos conversacionais com o modelo.

3.2.2 Frontend - Angular, Leaflet

O frontend foi implementado separadamente, utilizando Angular. Quando a componente do chat estava suficientemente desenvolvida de modo a obter um protótipo viável, foi levado a cabo um trabalho em conjunto com outros colaboradores da empresa de forma a conjugar estas duas componentes do projeto.

Trata-se de um mapa interativo desenvolvido com Angular CLI e utilizando o leaflet, uma biblioteca com funcionalidades de geração e manipulação de mapas. Esta biblioteca permite, de forma simples, definir áreas, popups, pins, etc. A tarefa de integração passa por extrair da resposta do modelo a informação apropriada e no formato correta para que ela possa ser exibida no mapa, em conjunto com o texto conversacional fornecido ao utilizador.

Durante a realização do estágio, foi estabelecida comunicação, nomeadamente na forma de reuniões digitais, com o engenheiro informático e colaborador da empresa Miguel Castro, responsável pelo desenvolvimento do frontend, de forma a orientar a implementação da aplicação web que processa o chat com as especificidades do mapa interativo. Esta comunicação foi estabelecida de forma autónoma e as tarefas comuns realizadas num repositório github partilhado exclusivamente entre os diferentes membros da equipa.

4 Descrição do Trabalho

4.1 Metodologia

O trabalho foi desenvolvido em coordenação próxima com o orientador do estágio Carlos Rebelo e a diretora da equipa de inteligência artificial dentro da empresa, Alexandra Oliveira.

Estabelecemos uma reunião semanal com todos os estagiários e com os coordenadores da empresa, em que era apresentado e debatido o trabalho desenvolvido durante a última semana e eram identificadas as áreas de melhorias e os próximos passos a tomar.

A figura 1 apresenta um plano de trabalho desenvolvido pouco tempo depois do início do período de trabalho, enfatizando a importância de um planeamento adequado, identificando corretamente as etapas de trabalho, elencando os desafios e obstáculos a ultrapassar e estruturando o trabalho dentro das limitações de recursos. Tal como apresentado no esquema, podemos definir a execução do plano de trabalho em 5 fases:

- **Inicialização do problema**: definir e identificar claramente os objetivos, isto é, que funcionalidades pretendemos que o sistema possua e que ferramentas podemos utilizar para as atingir
- **Reunião de meios/ferramentas**: analisar e comparar as diferentes possibilidades existentes, por exemplo, optar entre modelos open e closed source, considerando limitações a nível de configuração do modelo e a nível financeiro
- **Extração**: conseguir uma pipeline adequada que permita enviar a mensagem do utilizador para o modelo, possivelmente alterando-a para obter maximizar a qualidade da resposta, obter a mesma e ao mesmo tempo ser capaz de extrair desta resposta os dados necessários para a interação com o smart map
- **Aplicação**: definir claramente que informação será necessária o modelo devolver (para além de texto) de forma a que possa ser convertida ou diretamente aplicada ao mapa interativo
- **Conclusão**: avaliar o sistema, verificando que, a nível técnico, se comporta como desejado e as suas funcionalidades estão corretamente aplicadas, bem como garantindo a sua usabilidade junto do consumidor final, averiguando a interpratibilidade e acessibilidade do sistema

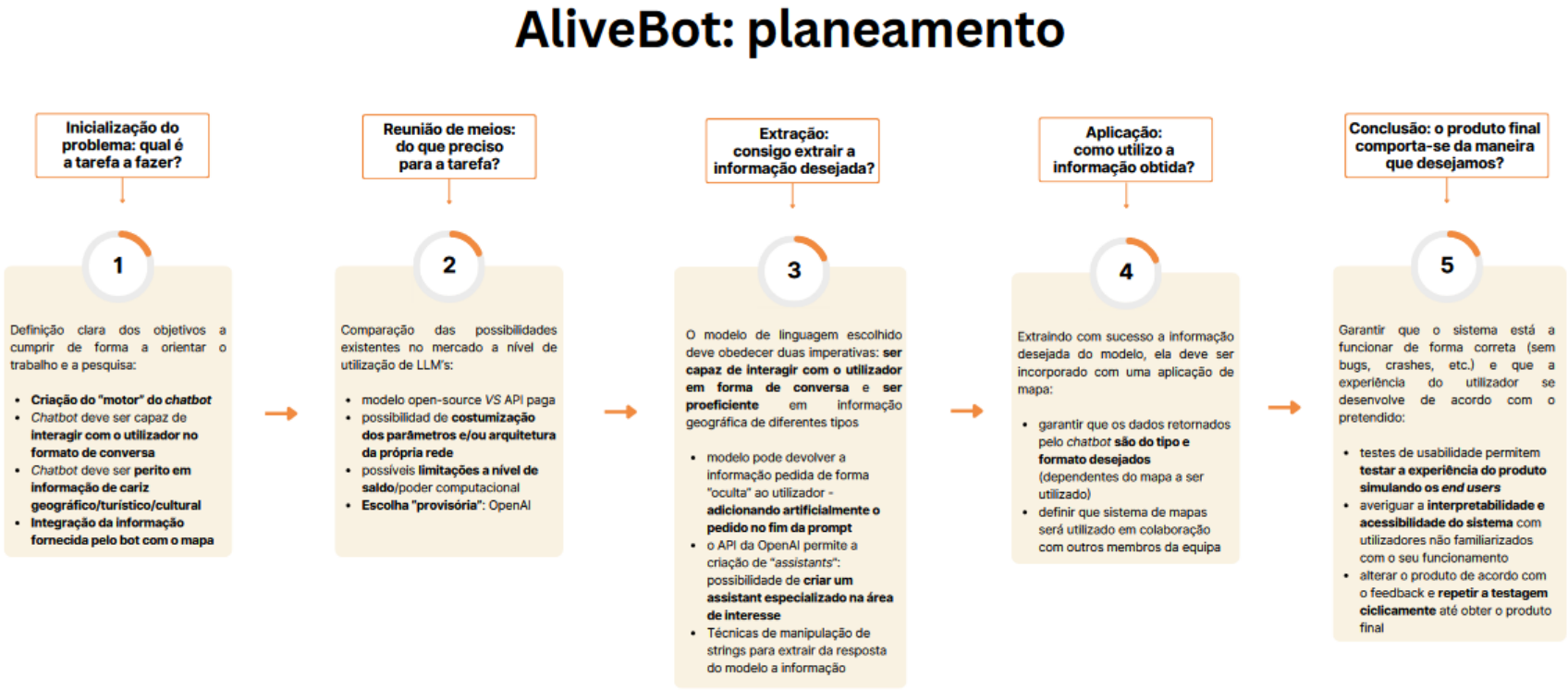


Figure 1: Primeira elaboração de um plano de trabalho

4.2 Implementação

De modo a interagir com o Assistant, foi implementado um servidor web em Python utilizando o módulo Flask, um framework que permite gerir *routes* e *requests*, em conjunto com o API da OpenAI.

Este *script* implementa uma aplicação web que usa a API da OpenAI para criar e gerir *threads*. Desta forma, processa as mensagens recebidas como input do utilizador e retorna a resposta do modelo, condicionada pela *prompt* escondida e a *prompt* do assistente, da qual extrai o texto a apresentar o utilizador e outra informação, como coordenadas, necessária para o funcionamento do sistema.

4.2.1 Gestão de Threads

O funcionamento do API baseia-se essencialmente na criação e manipulação de threads, conceptualmente equivalente a diferentes conversas, como descrito anteriormente. Cada thread contém a interação entre utilizador e modelo. Assim, este script permite a criação de novas threads, no sentido de criar uma nova conversa, ou o acesso a uma thread já existente através do seu id único, permitindo assim a existência de diferentes conversas com diferentes contextos, como é usual em aplicações que usam LLM's.

4.2.2 Processamento de Mensagens

O processamento da mensagem do utilizador e devolução da resposta do modelo baseia-se em POST requests. Quando é feita uma destas requests, é passada a mensagem do utilizador, um parâmetro que especifica se pretende criar uma thread nova ou a aceder a uma já existente (e o id da thread especificada, nesse caso). À mensagem do utilizador, é acrescentada uma "prompt extra/escondida", que define o comportamento do modelo. No fim do texto, é inserida a frase:

If you mention a landmark, location, business or anything of the sort, include at the end of your response its latitude and longitude according to this regex pattern: $([-+]?[0-9]*\dot{?}[0-9]+),*([-+]?[0-9]*\dot{?}[0-9]+)$

Esta *prompt* delimita a resposta do modelo de forma a que ela esteja de acordo com aquilo que o código espera. Dado que, como qualquer LLM, todas as versões do GPT são estocásticas, seria impossível prever exatamente de que modo o string que devolve estaria construído; desta forma, podemos afirmar com um elevado grau de certeza que o string de resposta, ainda que

varie, vai conter sempre o mesmo tipo de informação, o que nos permite processá-la.

4.2.3 Resposta do Modelo

A mensagem completa é inserida na thread e é executada uma run, ou seja, a chamada ao modelo GPT definido para inserir na thread a sua resposta. Quando a run está completa, obtemos a última mensagem da thread, equivalente à resposta do modelo, dividindo-a em texto utilizando a função referida anteriormente. Por fim, o request retorna um json com os seguintes parâmetros:

response: apenas o texto na resposta do modelo, geralmente informação relevante acerca de um monumento ou marco histórico no caso de ser esse o tópico da pergunta, ou apenas uma resposta ao utilizador

coordinates: as coordenadas extraídas da resposta do modelo de acordo com o padrão regex definido

role: um indicador de quem produziu a mensagem, user ou model

4.2.4 Mapa Interativo

O mapa interativo apresenta uma janela de chat, onde, como habitual com assistentes de texto, é exibida a conversa entre utilizador e modelo, ou seja, a componente de texto da resposta. Já a componente de coordenadas extraídas da resposta é utilizada para exibir pins no mapa referentes ao local que está a ser abordado na conversa.

Em baixo, na figura 2, vemos um exemplo de como seria a User Interface final. O mapa interativo, produzido com o leaflet, aparece ao lado de uma caixa de texto onde o utilizador pode escrever prompts e estabelecer a chat com o modelo de linguagem. No mapa estão realçadas duas áreas, um polígono e uma circunferência. O objetivo final será que as respostas do modelo interajam com o mapa de forma a realçar e identificar as áreas referidas na conversa.

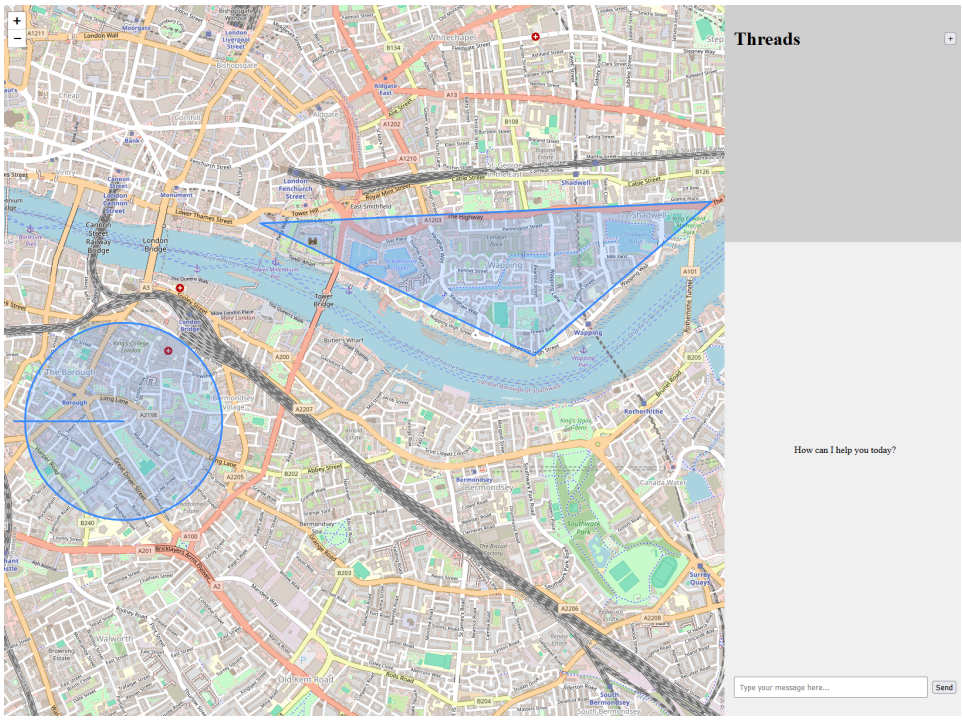


Figure 2: Exemplo de um mapa de áreas realçadas e a caixa de texto

Na figura 3, ainda em fase de prototipagem, as funcionalidades do frontend, nomeadamente a nível da caixa de texto, não estavam implementadas na sua totalidade. Porém, podemos recorrer à consola e verificar a resposta do modelo ao nosso prompt. Neste exemplo, o utilizador refere-se ao Palácio de Buckingham (sendo que o mapa que estamos a ver era de Londres). Como esperado, o modelo prontamente fornece a resposta com o adequada contexto histórico, cultural e turístico, que é processada para as suas três componentes de coordenadas, texto e role. As coordenadas são passadas para o sistema do mapa interativo que realça a área referida no mapa.

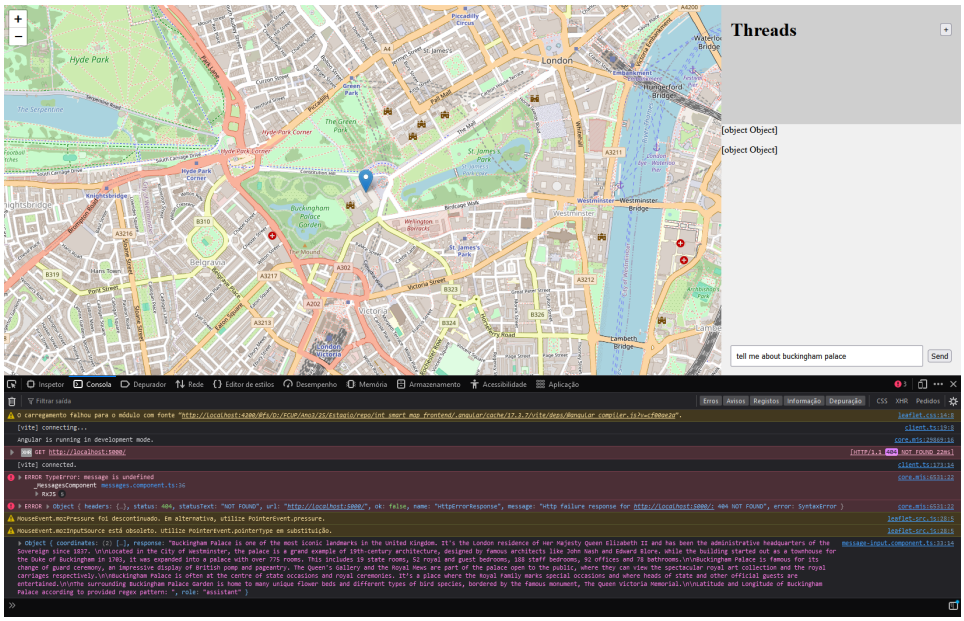


Figure 3: Resposta do modelo (componente de texto) e o local correspondente realçado

5 Resultados Experimentais e Conclusões

De modo a testar o protótipo, para além das funcionalidades demonstradas na secção anterior, devemos ainda considerar a usabilidade e a viabilidade financeira do produto.

5.1 Avaliação de Usabilidade

No que toca à usabilidade e acessibilidade, foi aplicado um teste de usabilidade baseado no System Usability Scale (SUS), em que oss utilizadores do protótipo preenchem um conjunto de 10 perguntas relativas à usabilidade do produto.

Estes dados foram recolhidos através do preenchimento de um inquérito Google Forms [5], onde estão disponíveis na sua totalidade. Neste relatório, escolhi incluir 3 perguntas em particular que demonstram a recetividade dos inquiridos perante o sistema, realçando a sua acessibilidade.

Verificamos que todos os utilizadores inquiridos gostariam de utilizar o sistema com, no mínimo, grande frequência.

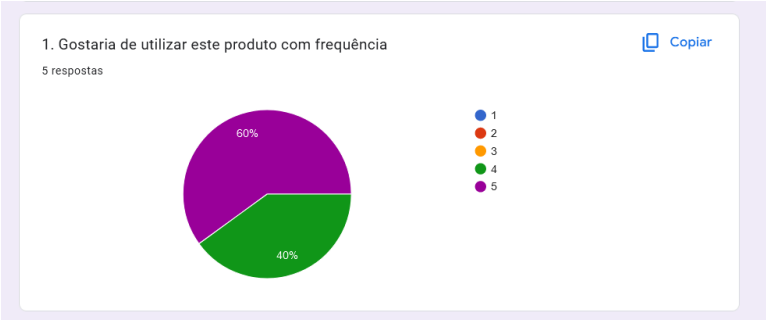


Figure 4: Valorização do produto

A esmagadora maioria dos utilizadores considerou o produto fácil ou muito fácil de utilizar.

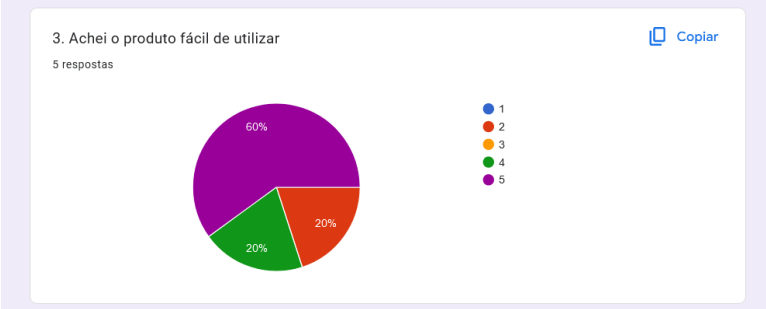


Figure 5: Facilidade de utilização/acesso

A grande maioria dos utilizadores considerou que não precisaram de experiência prévia para interagir com o sistema.

Figure 6: Necessidade de experiência prévia

Os resultados foram notoriamente positivos, demonstrando que, mesmo a nível de protótipo, o sistema é intuitivo e acessível a todo o tipo de utilizadores, indepedentemente da sua formação prévia.

5.2 Custos de Utilização

A outra vertente a considerar é o custo de utilização, visto que estamos a utilizar uma API e, como tal, cada chamada feita ao modelo tem associdado a ela um custo. Apesar deste custo ser descartável numa fase apenas de testagem, é importante que o sistema possua mecanismos de mitigação do custo caso este passe para uma fase de produção em que venha a ser aproveitado por utilizadores reais. Para além disso, temos ainda que considerar o volume de utilizadores expectável para este sistema, do qual depende o número de chamadas que seriam feitas ao modelo subjacente.

A nível de medidas de redução de custos, a prompt inicial do sistema impede-o de responder a perguntas fora da área particular a que é destinado o seu uso; assim, apesar de ter as capacidades completas do modelo GPT-4, o chatbot não responderá a prompts que difiram do seu uso pretendido como guia turístico, histórico e cultural, desincentivando a sua utilização como chatbot conversacional fora deste contexto, levando a que não sejam realizadas prompts desnecessariamente.

A API da OpenAI permite ainda monitorizar de forma detalhada todos os custos associados a calls ao sistema. Os custos associados à testagem e desenvolvimento estão na sua totalidade registados.

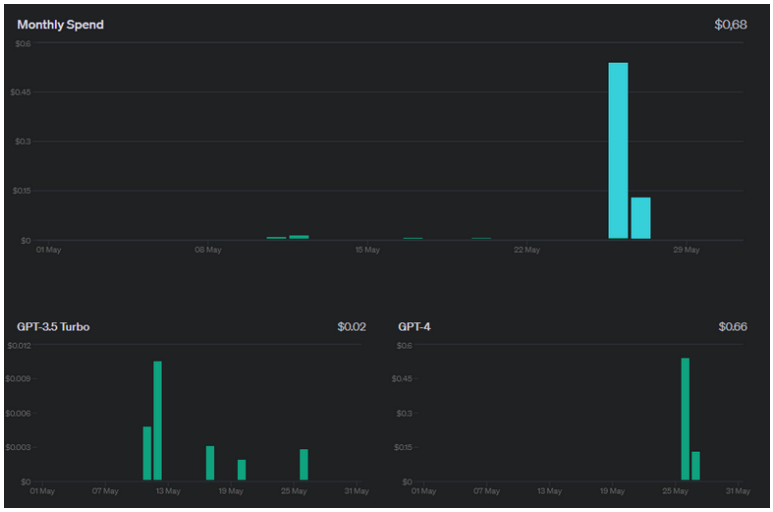


Figure 7: Custos da utilização da API para o mês de maio

No mês de maio podemos observar chamadas tanto ao GPT 3.5 Turbo como para o GPT-4, modelo que apresenta uma capacidade substancialmente melhorada, mas que, como tal, tem também associado um custo superior.

A decisão tomada foi no sentido de avançar com o GPT-4, tendo em a investigação que aponta para um conhecimento geográfico adequado[1] possuído pleos modelos GPT bem como os preços de utilização do API [2], que permitem um volume de utilização muito elevado, não compromentendo na qualidade da resposta do modelo

5.3 Conclusão

O projeto proposto explora uma área de grande potencial, tanto a nível científico como financeiro, apresentando-se como uma aplicação prática e proveitosa dos recentes avanços na tecnologia dos LLM 's.

O espaço de tempo destinado à realização do Projeto-Estágio não permitiu uma conclusão de todas as características e funcionalidades de que o sistema necessitaria; porém, o protótipo desenvolvido até ao momento representa um produto promissor, tendo obtido um feedback excecional junto dos consumidores nos testes realizados, bem como uma fundação sólida, com uma missão claramente definida à qual pode ser dada continuidade com base no trabalho até ao momento desenvolvido.

Este projeto permitiu ainda explorar conceitos nas áreas do desenvolvi-

mento web e Interação Pessoa-Máquina, unindo-os de forma única e creativa com conhecimentos técnicos e específicos da área da Inteligência Artificial, nomeadamente no que toca ao processamento de linguagem natural. Para além disso, foi-me dado a conhecer um ambiente dinâmico de trabalho em equipa com ênfase no feedback e no trabalho autónomo.

5.4 Trabalho Futuro

A continuação deste projeto passaria principalmente pela implementação de novas funcionalidades, nomeadamente a nível de manipulação da resposta do modelo.

O modelo deve ser capaz de: devolver as coordenadas de múltiplas localizações, enquanto que atualmente devolve apenas um único conjunto de coordenadas; definir polígonos e áreas, para além de localizações únicas, por exemplo no caso de se referir a espaços à volta de um monumento/atração (isto seria conseguido referindo-se à localização e definindo um raio à sua volta ou definindo uma série de pontos que constituiriam o polígono).

6 Apêndices

References

[1] Peter Mooney, Wencong Cui, Boyuan Guan, and Levente Juhász. Towards understanding the geospatial skills of chatgpt: Taking a geographic information systems (gis) exam. In *Proceedings of the 6th ACM SIGSPATIAL International Workshop on AI for Geographic Knowledge Discovery*, pages 85–94, 2023.

[2] OpenAI. Openai api pricing. `https://openai.com/api/pricing/`, 2024. Accessed: 2024-07-01.

[3] OpenAI. Openai gpt-4 api. `https://platform.openai.com/docs/guides/gpt`, 2024.

[4] David Scarin. Repositório do servidor alivebot. `https://github.com/davidmscarin/alive-bot`, 2024.

[5] David Scarin. Testes de usabilidade alivebot. `https://docs.google.com/forms/d/1p1zqPtGMEFbYYr-on6vy1_WNYFfJ_BHZq0qlvtVI10c/viewanalytics`, 2024.

Testes de Usabilidade - SUS

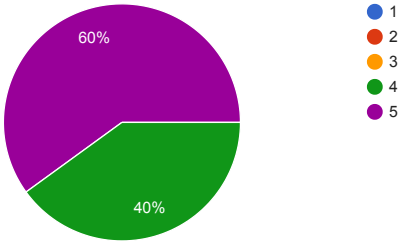
Testes de Usabilidade AliveBot

5 respostas

1. Gostaria de utilizar este produto com frequência

Copiar

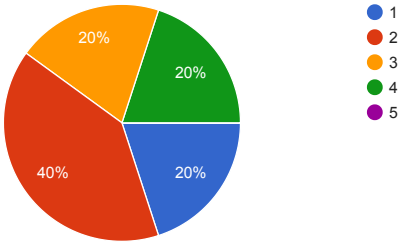
5 respostas



2. Considerei o produto mais complexo do que necessário

Copiar

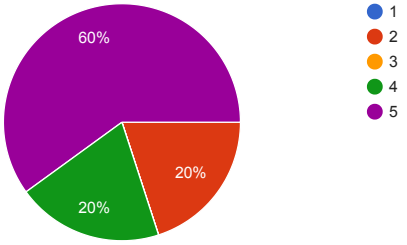
5 respostas



3. Achei o produto fácil de utilizar

Copiar

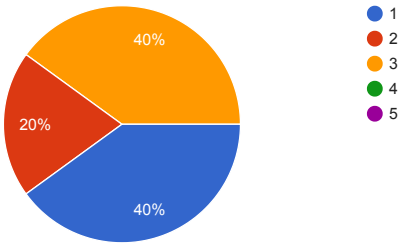
5 respostas



4. Acho que necessitaria de ajuda de um técnico para conseguir utilizar este produto

Copiar

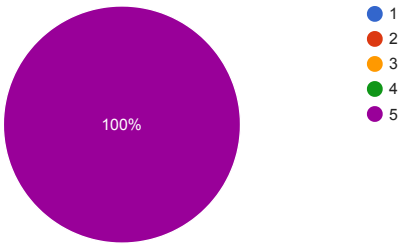
5 respostas



5. Considerei que as várias funcionalidades deste produto estavam bem integradas

Copiar

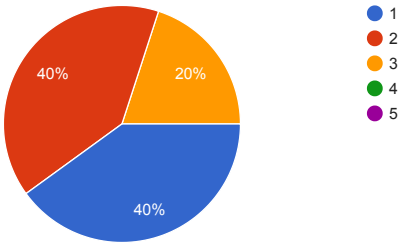
5 respostas



6. Achei que este produto tinhas muitas inconsistências

Copiar

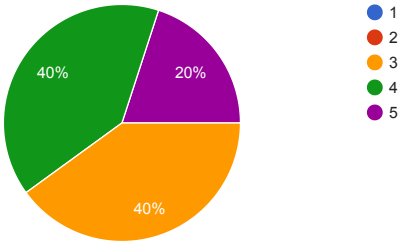
5 respostas



7. Suponho que a maioria das pessoas aprenderia a utilizar rapidamente este produto

Copiar

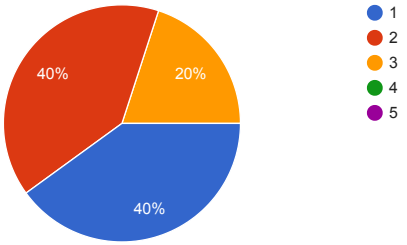
5 respostas



8. Considerei o produto muito complicado de utilizar

Copiar

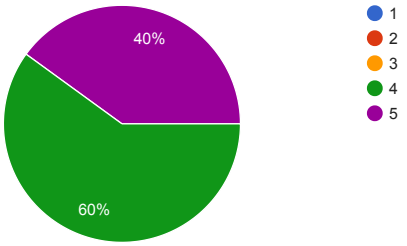
5 respostas



9. Senti-me muito confiante a utilizar este produto

Copiar

5 respostas



10. Tive que aprender muito antes de conseguir lidar com este produto

Copiar

5 respostas