



Relatório do Projeto - Robô Pet

Desenvolvimento de um Robô Social em I.A. com
Interação Humana Utilizando o Modelo
Arquitetônico Sentir-Pensar-Agir

Discentes: Maria Eduarda Couto Grossi
Raissa Rezende Oliveira
Thomas Anderson Lopes

Docente: André de Lima Salgado

Turma: 10A

1. Introdução

O projeto "Robô Pet" busca a implementação de um robô social que simula um animal de estimação inteligente, capaz de interagir com seres humanos de maneira natural e adaptativa. Este relatório técnico documenta o desenvolvimento do projeto, destacando seus desafios, soluções e tecnologias utilizadas.

2. Desafios Enfrentados

Durante o desenvolvimento do projeto, nossa equipe enfrentou diversos desafios, sendo eles:

2.1. Familiarização com Python: A equipe teve que superar a curva de aprendizado inicial da linguagem Python, que era nova para todos os membros.

2.2. Configuração do Ambiente: A configuração da biblioteca Eleven Labs para reconhecimento de fala no Windows exigiu etapas adicionais, incluindo download direto do site, configuração de variáveis de ambiente e execução de comandos no PowerShell, o que atrasou boa parte do desenvolvimento, pois demandou um grande esforço em conjunto até que o primeiro membro do grupo conseguiu configurar a biblioteca com êxito, ajudando os demais membros.

2.3. Limitações de Conta Gratuita: Para testar a funcionalidade de fala da Eleven Labs, tivemos que criar várias contas gratuitas devido ao limite de caracteres de fala.

2.4. Organização de Pacotes: Inicialmente, a equipe enfrentou dificuldades na organização dos pacotes e diretórios para seguir a arquitetura desejada.

2.5. Integração do Twilio: A integração do Twilio para possibilitar a realização de chamadas telefônicas automatizadas foi um desafio adicional, especialmente a configuração das chaves de acesso, que não podiam ser compartilhadas no GitHub. Isso exigiu um gerenciamento cuidadoso das chaves durante o desenvolvimento.

2.6. Atualização Automática da Interface Gráfica na Tela, Após Solicitação do Usuário: A principal dificuldade encontrada no projeto foi a atualização da imagem dos animais na tela da interface do usuário, quando ele escolhia qual seria seu tipo de animal na etapa de configuração. Para uma experiência mais agradável, era necessário que o robô se transformasse diretamente no animal escolhido sem a necessidade de encerrar a janela atual e abrir outra já atualizada. Inicialmente, o programa estava configurado de forma que a interface gráfica só era atualizada após o fechamento da janela, tornando a interação com o robô não contínua e frustrante para o usuário. Conseguimos resolver isso utilizando recursos de orientação a objetos do Python e funções de atualização automática da Tkinter, para que o controlador, mesmo definindo qual seria a nova imagem, pudesse atualizar por fora da thread da interface, utilizando uma variável de objeto compartilhada pelas duas funções.

2.7. Integração de Interface Gráfica e Fala/Escuta Paralelamente: Outro problema de grande impacto, foi a integração eficiente entre a exibição da interface gráfica e as interações de fala e escuta do robô. Resolver essa questão exigiu uma abordagem de paralelização das ações para permitir que a interface fosse atualizada enquanto as

interações de fala ocorriam em segundo plano. Com a ajuda da biblioteca Threading e sua implementação adequada, conseguimos melhorar significativamente a experiência do usuário. Entretanto, com isso surgiu um novo problema. Ao paralelizar as threads, criamos um problema no código onde as interações de fala e escuta só eram executadas após o encerramento da thread de exibição da interface gráfica em uma janela, que executava um main loop infinito até ser interrompido por um evento de fechamento de janela. Assim, nós precisamos rever o código para entender corretamente a maneira de paralelizar as threads, sem que houvesse essa dependência.

3. Gráficos analíticos do rendimento da equipe:

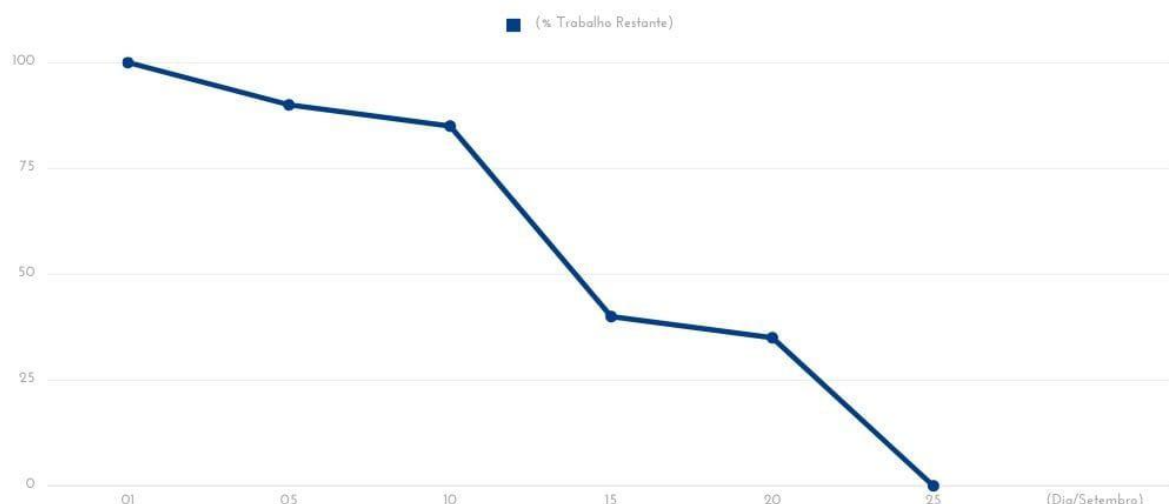
Nesta seção, apresentamos gráficos e estatísticas que demonstram o desempenho e o progresso da equipe ao longo do desenvolvimento do projeto.

3.1. Taxa de Desenvolvimento da Equipe em Burndown

Aqui, exibimos um gráfico de Burndown que ilustra visualmente a evolução da equipe no desenvolvimento do projeto ao longo do mês de setembro de 2023, período proposto para o decorrer desta atividade. Inicialmente, devido a curva de aprendizado tanto da linguagem quanto da base disponibilizada pelo professor, a equipe teve pouca produção. Com o passar do tempo, até a reta final, o rendimento direcionou-se apenas a aumentar, com exceção dos picos em que as maiores dificuldades, já listadas, já apareceram.

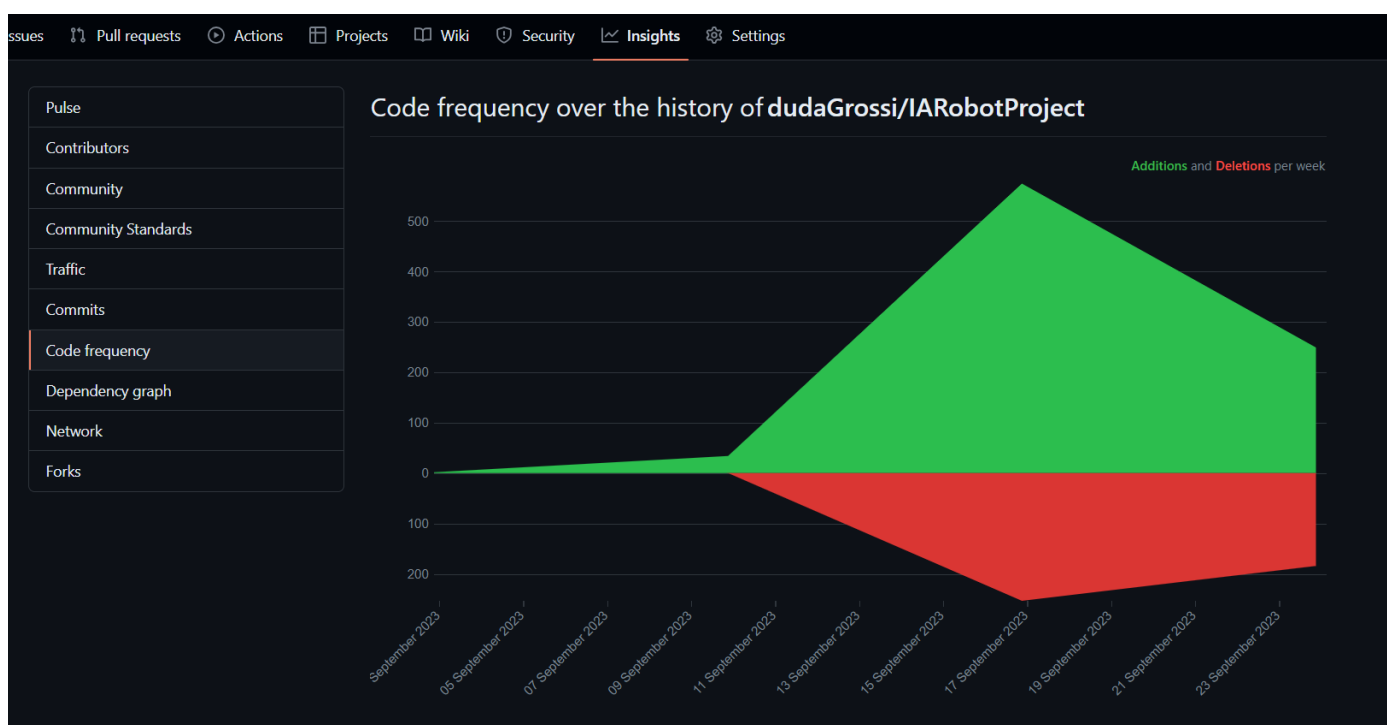
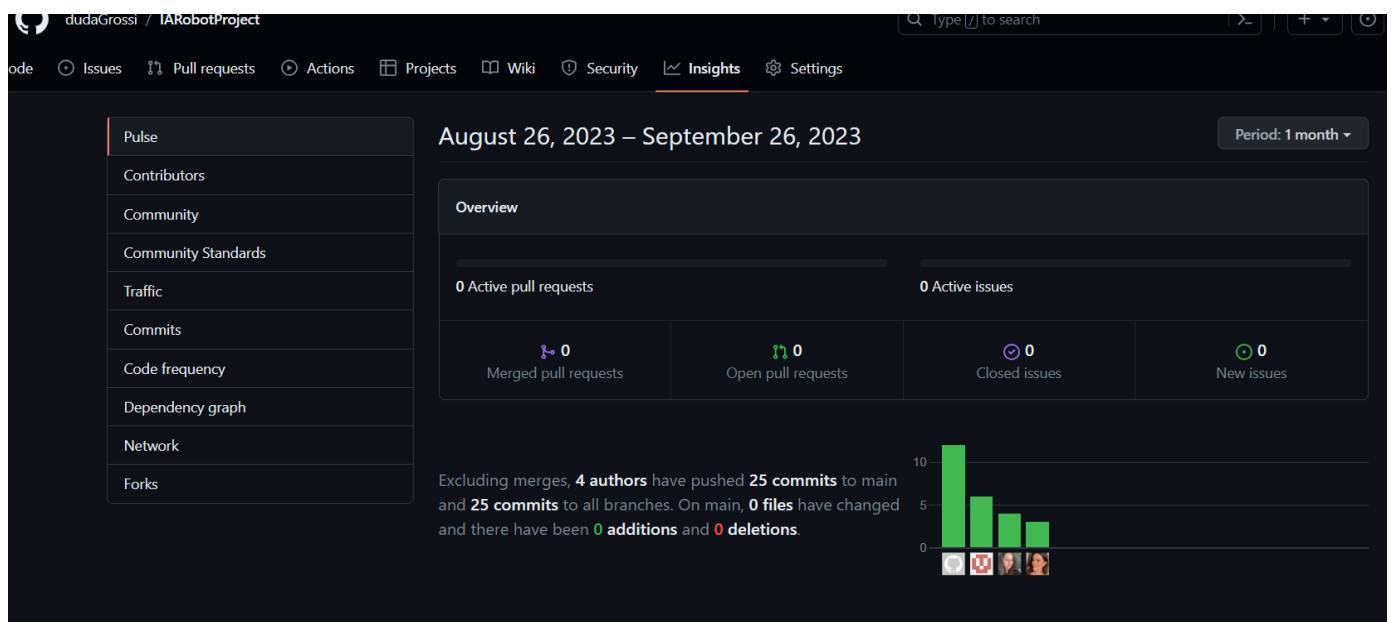


GRAFICO DE BURNDOWN (ROBÔ PET)



3.2. Estatísticas do GitHub

Esta parte oferece um resumo das estatísticas relevantes do GitHub, destacando o envolvimento da equipe no repositório do projeto, incluindo commits, branches e contribuições individuais.



4. Estratégias de Codificação e Tecnologias Utilizadas

Para superar esses desafios e desenvolver o projeto, adotamos várias estratégias:

4.1. Padronização de Nomes de Variáveis: Implementamos convenções de nomenclatura significativas para tornar o código mais compreensível.

4.2. Comentários Descritivos: Adicionamos comentários ao código para explicar trechos complexos ou detalhes importantes.

4.3. Documentação do Código: Criamos documentação abrangente para descrever a estrutura do projeto, os principais módulos e suas funcionalidades.

4.4. Utilização de Bibliotecas e APIs: Fizemos uso de várias bibliotecas e APIs essenciais para o projeto, incluindo Pygame, Eleven Labs, Speech Recognition, Speech To Text e Twilio, mais detalhadas sobre seus respectivos usos na documentação técnica.

5. Conclusão

O projeto "Robô Pet" representou um passo inicial importante no nosso aprendizado prático de sistemas de IA com interações humanas. Através dessa proposta, pudemos superar desafios iniciais, incluindo a configuração do ambiente e a familiarização com Python. A arquitetura "Sentir-Pensar-Agir" permitiu a interação natural com o usuário, enquanto a estrutura MVC organizou eficazmente o código.

Embora o projeto tenha alcançado resultados que entusiasmaram muito o grupo a aprender mais sobre Python e IAs, reconhecemos que trata-se de uma experiência inicial, altamente simplificada dentro do escopo proposto para o projeto e os limites de recursos e tempos, e por isso, existem oportunidades para melhorias futuras, como a expansão das capacidades de reconhecimento de voz e a adição de novas funcionalidades.