Relatório do Trabalho Prático

Desenvolvimento de Inteligência Artificial Distribuída

LUCAS SILVA BICALHO

MARCELO RODRIGUES JÚNIOR

MARIANA GABRIELE FERREIRA SALES

RENAN RIBEIRO PEREIRA

LAVRAS - MG

2025

SUMÁRIO

1 INTRODUÇÃO

O desenvolvimento de soluções baseadas em Inteligência Artificial (IA) tem se tornado uma tendência crescente em diversas áreas, proporcionando avanços significativos na automação de tarefas e na melhoria da experiência do usuário. Com o crescimento exponencial do uso da IA, surgem também desafios relacionados à segurança, proteção de dados pessoais e eficiência da infraestrutura computacional. Neste contexto, este trabalho tem como objetivo o desenvolvimento de uma solução de Inteligência Artificial Distribuída, utilizando duas redes neurais distintas para a geração de imagens e GIFs (Graphics Interchange Format) a partir de textos fornecidos pelos usuários.

A escolha deste problema foi baseada na crescente demanda por sistemas de geração de conteúdo multimodal, especialmente no campo do *design*, *marketing* e comunicação visual. Sistemas que permitem a conversão automática de textos em imagens ou animações podem facilitar a criação de conteúdo digital dinâmico e acessível. No entanto, esse tipo de aplicação também apresenta desafios técnicos e de segurança, especialmente no que se refere à privacidade e ao processamento distribuído das requisições.

2 DESENVOLVIMENTO

O sistema desenvolvido consiste em uma interface *web* que permite ao usuário inserir um texto e selecionar se deseja gerar uma imagem ou um GIF correspondente. A partir dessa entrada, o sistema direciona a requisição para um dos dois modelos de IA, cada um responsável por uma das tarefas. O processamento ocorre de forma distribuída, garantindo escalabilidade e eficiência na execução.

A solução foi implementada utilizando containers Docker para cada um dos agentes de IA, garantindo isolamento e facilidade na implantação. Além disso, uma API (Application Programming Interface) REST (Representational State Transfer) foi desenvolvida para intermediar a comunicação entre o front-end e os agentes de IA, permitindo integração flexível com outros sistemas. O uso de tecnologias modernas, como Flask e FastAPI, garantiu uma arquitetura leve e eficiente.

Durante o desenvolvimento, foi identificada a necessidade de garantir a compatibilidade do Docker com imagens que ofereçam suporte à tecnologia CUDA (Compute Unified Device Architecture). Para isso, foi necessário realizar a instalação do NVIDIA Container Toolkit, uma solução disponibilizada pela própria NVIDIA, que permite a utilização de GPUs (Graphics Processing Unit) dentro de contêineres Docker. Com essa solução, conseguimos habilitar a aceleração por *hardware* dentro dos contêineres, garantindo o funcionamento adequado das aplicações que dependem de CUDA.

Em termos de segurança, foi elaborado um Relatório de Impacto à Proteção de Dados Pessoais (RIPD), que identifica e avalia os principais riscos associados ao processamento de informações no sistema. Utilizando a metodologia STRIDE, foram identificadas ameaças potenciais, como *spoofing*, *tampering* e *denial of service*, e foram propostas estratégias de mitigação, incluindo criptografia TLS, autenticação baseada em *tokens* e monitoramento de atividades suspeitas.

O código-fonte completo do projeto está disponível no repositório do GitHub, contendo toda a documentação necessária para implantação e uso do sistema. O repositório inclui instruções detalhadas para execução dos containers, descrição da arquitetura do sistema e exemplos de uso.

2.1 Arquitetura do Sistema

O sistema foi projetado para ser modular e distribuído, garantindo eficiência no processamento das requisições. A arquitetura geral está dividida nos seguintes componentes:

- **Front-end:** Interface *web* responsiva onde o usuário insere um texto e escolhe entre gerar uma imagem ou um GIF.
- **Back-end:** Responsável pelo processamento das requisições, utilizando APIs REST desenvolvidas com FastAPI.
- **Agentes de IA:** Dois modelos de rede neural especializados para cada tipo de geração (imagem e GIF).
- **Infraestrutura:** Contêinerização com Docker e orquestração via Docker Compose.

2.2 Implementação dos Agentes de IA

Os agentes de IA foram implementados utilizando a biblioteca ***diffusers***, que permite trabalhar com modelos de difusão pré-treinados para geração de conteúdo. Os modelos utilizados foram:

- **Stable Diffusion v1-5** para conversão de texto em imagem.
- **AnimateDiff-Lightning** para conversão de texto em GIF.

Os agentes foram implantados em contêineres que suportam **CUDA**, utilizando a imagem base nvidia/cuda:12.8.0-runtime-ubuntu24.04 para otimizar o processamento em GPUs NVIDIA.

2.3 Integração Front-end e Back-end

Para o desenvolvimento do ***front-end***, utilizamos **HTML, JavaScript puro e Bootstrap** para a estilização da interface. Além disso, utilizamos o **Nginx**, um servidor *web* de código aberto, para hospedar a aplicação no cliente.

A integração com o *back-end* foi feita utilizando a **Fetch API do JavaScript**, garantindo comunicação assíncrona eficiente entre cliente e servidor.

O ***back-end***, desenvolvido utilizando **Python** com **FastAPI**, expõe *endpoints* REST para interação com os agentes de IA e o *front-end*. A API foi projetada para ser leve e de fácil expansão, permitindo futuras melhorias e integrações.

2.4 Contêinerização e Orquestração

Para garantir um ambiente isolado e reprodutível, utilizamos **Docker** para criar contêineres separados para cada agente de IA. O **Nginx** foi utilizado como servidor *web* para hospedar o *front-end*, garantindo desempenho otimizado.

A orquestração foi feita com **Docker Compose**, onde definimos:

- Contêiner para o servidor web (Nginx-Alpine);
- Contêineres para cada agente de IA:
    - Stable Diffusion v1-5 (Text-to-image);
    - AnimateDiff-Lightning (Text-to-gif).
- Configuração de dependência entre os serviços para garantir funcionamento correto.

3 CONSIDERAÇÕES FINAIS

A implementação deste projeto demonstrou a viabilidade de um sistema distribuído para geração de conteúdo visual a partir de textos, destacando os benefícios da modularização por meio de containers Docker e da utilização de APIs REST. Durante o desenvolvimento, desafios como a escalabilidade do sistema e a segurança das requisições foram abordados, resultando em soluções eficazes para garantir um serviço seguro e eficiente.

A abordagem baseada na metodologia STRIDE permitiu uma avaliação detalhada dos riscos e ameaças, contribuindo para a implementação de medidas de segurança robustas. No futuro, melhorias podem ser realizadas na autenticação dos usuários e na otimização do desempenho dos modelos de IA.

Em relação a geração tanto de imagens quanto de GIFs a partir de um determinado *prompt*, alcançamos o resultado esperado, comprovando a viabilidade da proposta. No entanto, identificamos algumas limitações relacionadas tanto aos modelos utilizados quanto ao *hardware* disponível, o que pode afetar a qualidade das mídias geradas. Ou seja, observamos que, em algumas situações, os resultados não são totalmente precisos. Contudo, verificamos que aumentar o número de passos no processo de inferência melhora a qualidade das imagens e GIFs gerados, mas, ao mesmo tempo, demanda um maior uso de recursos computacionais, podendo afetar o desempenho do sistema.

De modo geral, este trabalho forneceu uma solução inovadora para a geração de imagens e GIFs a partir de texto, explorando as vantagens dos sistemas distribuídos e destacando a importância da segurança e da proteção de dados em aplicações baseadas em IA.

4 LINKS

Repositório no GitHub: https://github.com/celo-rod/vision-flow

Relatório de Impacto à Proteção de Dados Pessoais(RIPD): https://docs.google.com/document/d/1UBU22Yd8k46be-Fr8VdLBWtH2ul7dGY0fOhyd6Wvt3c/edit?pli=1&tab=t.0