

Hackathon Tractian na UNICAMP - Documentação do Projeto

Equipe

Ana Helena Tuchiama Leite de Almeida

Enzo de Oliveira Farias

Kevin Caio Marques dos Santos

Luc Joffily Ribas

Tema: Nova Feature - Classificação de Imagens de Máquinas com ChatGPT

Contextualização

Bem-vindo ao Hackathon Tractian na UNICAMP! Este evento reúne mentes criativas e inovadoras com o objetivo de desenvolver soluções tecnológicas que possam contribuir para a melhoria dos processos de manutenção industrial. O desafio proposto envolve a criação de uma nova feature para a plataforma da Tractian, focada na **classificação de imagens de máquinas** utilizando as capacidades do GPT e outras tecnologias de IA.

Nosso desafio é desenvolver um sistema que, dado um título, tipo de modelo, e algumas imagens, utilize o GPT para gerar automaticamente uma **ficha técnica da máquina**. Esta ficha técnica deve conter informações detalhadas e relevantes, facilitando o cadastro e gerenciamento dos equipamentos na plataforma da Tractian.

Objetivo

O principal objetivo deste projeto é criar um sistema capaz de automatizar e simplificar o processo de cadastro de equipamentos industriais, utilizando tecnologias de ponta em processamento de linguagem natural e visão computacional. O sistema gerado será fundamental para fornecer dados essenciais ao time de manutenção, melhorando a eficiência operacional e a precisão dos dados cadastrados.

Descrição das Principais Features

1. Formulário de Cadastro de Equipamentos

- **Descrição:** Uma interface web simples que permite ao usuário inserir o título da máquina, escolher o tipo de modelo e fazer o upload de até 3 imagens. Este formulário é o ponto de partida para a geração automática da ficha técnica.
- **Tecnologias Utilizadas:** HTML, CSS, Flask, Python.

2. Geração Automática de Ficha Técnica com GPT

- **Descrição:** Utilizando o modelo `gpt-4o`, o sistema gera automaticamente uma ficha técnica completa baseada nas informações fornecidas pelo usuário. Esta ficha inclui campos como Nome, Tipo, Fabricante, Modelo, Identificação, Localização e Especificações Técnicas.
- **Tecnologias Utilizadas:** OpenAI API, Flask, Python.

3. Upload e Análise de Imagens

- **Descrição:** O sistema permite o upload de até 3 imagens da máquina. Futuramente, estas imagens poderão ser analisadas usando técnicas de visão computacional para extrair informações adicionais que possam enriquecer a ficha técnica.
- **Tecnologias Utilizadas:** Flask, Python, Google Vision API (em futuras iterações).

4. Pesquisa de Fichas Técnicas já cadastradas

- **Descrição:** Após o cadastro, as fichas técnicas das máquinas são armazenadas no sistema, permitindo que os usuários façam buscas eficientes pelo nome da máquina. Esta funcionalidade facilita o acesso rápido e organizado às informações já cadastradas, essencial para consultas e tomadas de decisão futuras.
- **Tecnologias Utilizadas:** HTML, CSS, Flask.

5. Interface Amigável e Interativa

- **Descrição:** A interface foi projetada para ser simples e intuitiva, permitindo uma experiência de usuário fluida e eficiente. O sistema redireciona o usuário automaticamente para a visualização da ficha técnica gerada, garantindo uma navegação fácil.
- **Tecnologias Utilizadas:** HTML, CSS, Flask.

6. Busca por Caracteres Específicos nas Fichas Técnicas

- **Descrição:** Esta funcionalidade permite que os usuários filtrem as máquinas cadastradas com base na presença ou ausência de caracteres específicos no nome da máquina. Por exemplo, é possível buscar todas as máquinas que contêm ou não a letra "a" no nome. Esta busca refinada facilita a navegação em grandes conjuntos de dados e permite consultas mais detalhadas e personalizadas. A busca funciona

utilizando a API do GPT, portanto a busca pode ser feita utilizando linguagem natural, por exemplo: “nomes sem a letra ‘a’ e do modelo ‘B’”.

- **Tecnologias Utilizadas:** HTML, CSS, Flask, GPT API.

Setup Inicial do Projeto

1. Pré-requisitos

Antes de começar a configuração do projeto, certifique-se de que você tem os seguintes softwares instalados no seu ambiente de desenvolvimento:

- **Python 3.8+:** Linguagem de programação utilizada para o desenvolvimento do backend.
- **pip:** Gerenciador de pacotes do Python, utilizado para instalar as dependências necessárias.
- **Git:** Controle de versão para clonar o repositório do projeto.
- **Editor de Código:** Recomenda-se o uso de um editor como VS Code para facilitar o desenvolvimento e a depuração.

2. Clonando o Repositório

Endereço do repositório: <https://github.com/Kvn12/HackathonTractian>

Primeiro, clone o repositório do projeto para a sua máquina local:

```
Python
git clone https://github.com/Kvn12/HackathonTractian.git
cd HackathonTractian
```

3. Configurando um Ambiente Virtual

É recomendado criar um ambiente virtual para isolar as dependências do projeto:

```
Python
python -m venv venv
#Linux source venv/bin/activate
#Windows venv\Scripts\activate
```

4. Instalando as Dependências

Com o ambiente virtual ativado, instale todas as dependências necessárias para o projeto. As dependências estão listadas no arquivo `requirements.txt`:

```
Python  
pip install -r requirements.txt
```

5. Configurando as Variáveis de Ambiente

Você precisará configurar algumas variáveis de ambiente para que o sistema funcione corretamente, especialmente a chave da API da OpenAI:

1. Crie um arquivo `.env` na raiz do projeto.
2. Adicione as seguintes linhas, substituindo com suas próprias credenciais:

```
Python  
api_key="sua-chave-api-aqui"
```

6. Executando o Projeto

Com todas as dependências instaladas e as variáveis de ambiente configuradas, você pode iniciar o servidor Flask:

```
Python  
python app.py
```

O servidor estará rodando em `http://127.0.0.1:5000/`. Acesse este endereço no seu navegador para utilizar o sistema.

