OtimizaIA

Integrates: Caio Viana de Azeredo 252035715,
Felipe Santos Araujo 252033443,
Robson Soares dos Santos 252027240,
Thiago Souza Lobo Gomes 252001904
Tutor: Marcio Vinicius Da Silva Guimarães

Projeto: Otimização de prompt de IA visando diminuir o Consumo de Água e Energia, conscientizando o usuário

1. Introdução

Com o crescente uso das "inteligências artificiais", ou os Large Language Models – LLM, verificadas no dia a dia da população, seja para consultas simples feitas via celular ou para estudos acadêmicos e geração de imagens e gráficos científicos, foi possível verificar a popularização desse tipo de uso.

Tendo em vista que o sucesso na resposta dos LLMs, seja em veracidade ou em exatidão da resposta, está diretamente relacionada com a formulação das sentenças no prompt de entrada. Faz-se necessária dominar a formulação dos prompts de forma a serem eficazes.

A engenharia de prompt que remonta dos anos de 2010 com os estudos do processamento da linguagem natural (Natural Language Processing – NLP), teve entre os anos de 2020 e 2021 destaque renovado pela popularização das LLMs através do GPT3.

Estima-se que são feitas atualmente mais de 1 bilhão de interações com o chat GPT por dia e que isso representa um consumo de 10 a 25 milhões de litros d´água diariamente, segundo o Exploding Topics 2025 visto na página https://www.ufsm.br/2025/09/04/como-o-uso-de-inteligencias-artificiais-consome-agua, em 30/10/2025.

É nesse contexto que o projeto está inserido apresentando como objetivo conscientizar os usuários sobre o consumo de recursos naturais, como água e energia, durante interações com modelos de linguagem (LLMs).

A aplicação propõe uma análise prévia do prompt escrito pelo usuário visando três pilares fundamentais da engenharia de prompts: clareza, contexto e direção. Como resultado da

análise do prompt serão identificadas partes desnecessárias no texto e estimado o impacto ambiental causados por elas.

2. Objetivo Central

Criar uma aplicação que permita que o usuário insira um prompt e, com base em uma base de dados e regras pré-definidas, a aplicação identifique saudações, frases redundantes e trechos supérfluos para uma pesquisa em LLMs.

Essas partes são associadas a um consumo estimado de água e energia, e o sistema mostra o impacto total gerado.

3. Visão Geral da Implementação

A implementação da aplicação proposta baseia-se em uma arquitetura de três camadas: **banco de dados**, **back-end** e **front-end**, integradas de forma a permitir que o usuário insira um prompt, o sistema processe as informações e retorne resultados visuais e numéricos sobre o consumo estimado de água e energia.

Cada camada possui responsabilidades específicas e se comunica por meio de uma **API RESTful**, garantindo modularidade, escalabilidade e facilidade de manutenção do sistema.

A aplicação é composta pelos seguintes componentes:

Componente	Tecnologia sugerida	Função
Front-end	Angular	Interface para digitar e visualizar resultados
Back-end	Python com FastAPI	API REST para análise e cálculo
Banco de dados	SQLite ou PostgreSQL	Armazena frases e valores médios de consumo
Visualização	Chart.js / Recharts	Exibe consumo em gráficos (Talvez)

Exemplo de prompt: "Olá, tudo bem? Me explique detalhadamente o que é um átomo, por favor." Partes desnecessárias: 'Olá, tudo bem?', 'detalhadamente', 'por favor'.
Consumo desnecessário estimado: 1,2 litros de água e 0,7 Wh de energia.
Versão otimizada: 'Explique o que é um átomo.'

4. Banco de Dados

O banco de dados será responsável por armazenar as frases consideradas desnecessárias, suas classificações, os valores médios de consumo de água e energia associados a cada uma, além de guardar o histórico das análises realizadas.

O modelo de dados adotado será **relacional**, utilizando **SQLite** em ambiente local e **PostgreSQL** em produção.

Tabela Principal

- **Frases Desnecessárias:** armazena os textos redundantes, saudações ou encerramentos e seus respectivos valores de consumo.

Cada registro de frase desnecessária contém campos como *tipo*, *texto*, *consumo de água (ml)* e *consumo de energia (Wh)*.

Essas informações serão utilizadas pelo back-end para compor os relatórios e calcular a economia gerada após a otimização do prompt.

Exemplo da tabela de frases desnecessárias:

id	tipo	texto	consumo_agua_ml	consumo_energia_wh
1	saudacao	Olá	0.5	0.02
2	saudacao	Tudo bem?	0.8	0.03
3	redundancia	Por favor, me explique detalhadamente	3.0	0.12
4	encerramento	Obrigado pela ajuda	1.2	0.05

5. Integração com o Back-end

1. **Receber o prompt do usuário:** a aplicação enviará o texto digitado através de uma requisição HTTP (método POST).

2. **Processar o prompt:** o sistema buscará no banco de dados expressões que correspondam a frases desnecessárias e calculará o impacto ambiental associado a elas.
3. **Gerar o prompt otimizado:** o sistema removerá as partes redundantes e apresentará uma versão mais objetiva do texto.
4. **Calcular o consumo:** com base em parâmetros pré-definidos (como consumo médio por interação), o back-end estimará a quantidade total de água e energia associada ao prompt original e ao prompt otimizado.
5. **Enviar a resposta ao front-end:** os resultados serão retornados em formato JSON, contendo o prompt original, o otimizado, as partes removidas e os valores de consumo calculados.

Fluxo de Comunicação

1. O front-end envia o prompt digitado pelo usuário para a API (via método POST).
2. O back-end acessa o banco de dados, identifica as frases desnecessárias e calcula os impactos correspondentes.
3. O resultado é estruturado e devolvido ao front-end, que o exibe de forma clara e visual.

O uso de **FastAPI** facilita essa comunicação, permitindo a criação de rotas bem definidas, validação automática dos dados de entrada e saída e integração simples com o banco de dados por meio de bibliotecas como **SQLAlchemy** ou **SQLModel**.

6. Integração com o Front-end

O front-end será desenvolvido em **Angular**, que oferece uma arquitetura baseada em componentes e excelente integração com APIs REST.

Ele será responsável pela interação com o usuário e pela apresentação dos resultados de maneira didática, intuitiva e visual.

Principais Componentes da Interface

- **Campo de entrada:** onde o usuário digitará o prompt a ser analisado.
- **Botão de análise:** ao clicar, o prompt será enviado para o back-end por meio de uma requisição HTTP.
- **Área de resultados:** exibe o prompt original com as partes desnecessárias destacadas, o prompt otimizado e os dados de consumo de água e energia.
- **Gráficos e indicadores visuais:** permitem visualizar o consumo total e o consumo evitado, reforçando o impacto positivo da otimização.

Fluxo de Funcionamento no Front-end

1. O usuário insere o texto no campo apropriado.
2. Ao clicar em "Analisar", o front-end envia o texto ao back-end.
3. O back-end processa o texto e retorna uma resposta estruturada.
4. O front-end interpreta a resposta e exibe:
    a. As partes desnecessárias (com destaque visual).
    b. Os valores de consumo de água e energia (totais e desnecessários).
    c. O prompt otimizado, para que o usuário compreenda a diferença.
5. O sistema pode ainda oferecer **gráficos** (com Chart.js ou Recharts) demonstrando a economia obtida e permitindo comparações entre diferentes análises.

7. Comunicação entre as Camadas

A comunicação entre o front-end e o back-end ocorrerá por meio de **requisições HTTP** do tipo **POST** e **GET**, utilizando o formato **JSON** para o envio e recebimento dos dados.

O front-end utilizará serviços responsáveis por consumir a API, tratando as respostas e repassando-as aos componentes visuais.

O back-end, por sua vez, acessará o banco de dados através de um **mapeamento objeto-relacional**, garantindo consistência nas consultas e inserções.

O fluxo completo ocorre da seguinte forma:

1. O usuário envia o prompt no front-end.
2. O front-end faz uma requisição POST para a rota **/api/analisar**.
3. O back-end processa o texto, consulta o banco e calcula os consumos.
4. O resultado (em JSON) retorna ao front-end.
5. O front-end exibe o resultado e, opcionalmente, envia o registro ao histórico do usuário.

8. Cálculo de Consumo

Os valores de consumo são baseados em estudos sobre o impacto ambiental de modelos de linguagem. Por exemplo, uma única consulta pode consumir aproximadamente 500 ml de água e 0.5 Wh de energia. O sistema distribui esse consumo proporcionalmente ao tamanho e redundância do prompt.