

Relatório Final - Sistema de Recomendação de Xícaras para Café

1. Introdução

Contexto e Importância do Projeto

A cadeia produtiva do café abrange uma série de processos que vão desde a plantação até a comercialização do produto. Um aspecto importante e que muitas vezes é negligenciado é o impacto da experiência de consumo, que inclui, entre outros fatores, a escolha da xícara ideal para o tipo de café servido. Dessa forma, um [estudo científico](#) revela que a cor e o formato da xícara podem influenciar a percepção do sabor e a satisfação do consumidor.

De acordo com a Associação Brasileira da Indústria do Café (ABIC) aquisição do produto no país registrou um aumento de 1,64%; já a consumo per capita aumentou 7,47%. Assim, com o crescimento do mercado de cafés especiais, a personalização da experiência do consumidor tem se tornado cada vez mais importante. Assim, este projeto visa desenvolver uma solução para recomendar a melhor cor de xícara com base nas características do café, a fim de proporcionar uma experiência mais agradável e alinhada com as preferências sensoriais dos consumidores.

Problema Identificado e Relevância para a Cadeia Produtiva do Café

O problema identificado está relacionado à falta de personalização nas recomendações para a escolha da xícara de café. Tradicionalmente, a escolha da xícara é feita de forma aleatória ou sem considerar características específicas do café. A relevância para a cadeia produtiva do café reside na melhoria da experiência do consumidor, que pode gerar maior satisfação e fidelização, impactando positivamente a imagem do produto e a competitividade do mercado.

Objetivos da Solução

O objetivo deste projeto é desenvolver uma API de recomendação que, ao receber informações sobre o tipo de café e suas características sensoriais, recomende a cor de xícara mais adequada para a preparação do café. O sistema será baseado em agentes inteligentes que irão processar os dados de entrada e fornecer respostas personalizadas para os usuários.

2. Desenvolvimento

Documentação da Implementação

API REST: Tecnologias Utilizadas e Endpoints Principais

A implementação da solução foi realizada utilizando a arquitetura de API RESTful, desenvolvida com o framework **FastAPI**. A API possui dois endpoints principais:

1. **/recomendar-xicara:**

- **Método:** POST
- **Descrição:** Recebe as características do café e retorna a cor de xícara recomendada com base em um modelo de linguagem inteligente.
- **Tecnologias:** Utilização do modelo **Ollama Phi3** para a análise das características do café e recomendação da cor de xícara.

2. **/enviar-dados:**

- **Método:** POST
- **Descrição:** Recebe as características do café e faz uma chamada para a API de recomendação, retornando uma mensagem gerada via Bard (inteligência artificial que gera respostas personalizadas).
- **Tecnologias:** **FastAPI**, **Requests** (para comunicação com a outra API) e **BardAPI** (para gerar a resposta com base na cor recomendada).

A API foi configurada para permitir conexões CORS (Cross-Origin Resource Sharing), facilitando a interação entre diferentes origens, como por exemplo, entre front-end e back-end.

Agentes Inteligentes: Função de Cada Agente e Interação com a API

O sistema é composto por dois agentes inteligentes principais:

1. **Agente de Recomendação (Main Service):**

- Responsável por analisar as características do café (acidez, corpo, doçura e tipo de café) e determinar a cor de xícara mais adequada. Para isso, utiliza o modelo **Ollama Phi3** e uma base de dados com exemplos de combinações de café e cor de xícara.

2. **Agente de Mensagem (Client Service):**

- Recebe a cor recomendada do Agente de Recomendação e gera uma mensagem explicativa utilizando o modelo **BardAPI**, que elabora uma resposta personalizada, dizendo ao usuário que a cor da xícara é ideal para o café.

Containerização: Configuração e Organização dos Containers Docker

A aplicação foi containerizada utilizando o Docker, que permite a execução dos serviços de forma isolada e portátil. A estrutura do Docker foi organizada da seguinte maneira:

- **Container main:** Contém a API de recomendação, que utiliza o modelo de linguagem para determinar a cor de xícara.
- **Container client:** Contém a API de geração de mensagens, que elabora a resposta final ao usuário.

Os containers se comunicam entre si por meio de uma rede Docker (cafe_network), garantindo a interação fluida entre os componentes da aplicação.

O Relatório RIPD, assim como o código completo do projeto, pode ser encontrado no repositório da aplicação : <https://github.com/RibeiroEduardo8/Sistemas-Distribuidos-Final>.

3. Considerações Finais

Resultados e Conclusões

Comparação entre Expectativas e Resultados Obtidos

O projeto foi capaz de atingir os objetivos propostos, recomendando com precisão a cor da xícara para diferentes tipos de café com base nas suas características sensoriais. As expectativas de personalização da experiência do usuário foram atendidas, com a criação de respostas únicas para cada entrada.

Desafios Enfrentados e Aprendizados

Durante a implementação, enfrentamos desafios relacionados à integração entre os containers Docker e à configuração das APIs externas (Ollama e Bard). No entanto, esses obstáculos foram superados com ajustes finos na arquitetura dos containers e na configuração das chaves de API.

Sugestões de Melhorias Futuras

- **Expansão da base de dados:** Incluir mais tipos de café e cores de xícaras para melhorar a precisão das recomendações.
- **Otimização da performance:** Melhorar a latência das chamadas para as APIs externas, implementando cache de respostas para tipos de café mais comuns.
- **Aprimoramento da interface de usuário:** Melhorar a usabilidade do front-end com opções de seleção de características do café, ao invés de campos de texto livre

Fontes:

<https://hubdocafe.cooxupe.com.br/consumo-de-cafe-no-brasil/>

<https://uniquecafes.com.br/influencia-das-cores-nas-notas-sensoriais-do-cafe/#:~:text=Acreditamos%20que%20a%20cor%20da%20x%C3%ADcara,doce%20do%20que%20realmente%20%C3%A9>

<https://dnahub.com.br/corsaborcafe>