

UNIVERSIDADE ESTADUAL DE CAMPINAS  
INSTITUTO DE COMPUTAÇÃO

Docente: Prof. Dr. Julio C. dos Reis [dosreis@unicamp.br]

## DESAFIO AGENTES CONVERSACIONAIS

### Objetivo

Sistemas multi-agentes podem ser aplicados para diversos contextos de recomendação. Nesta tarefa, você utilizará agentes inteligentes para processar avaliações de restaurantes, extraindo e atribuindo pontuações automaticamente. O objetivo é construir um sistema baseado em agentes para recuperar, processar e classificar avaliações de forma coordenada e eficiente. Isso desempenha um papel chave para recomendações aos usuários.

Os principais aspectos avaliados incluem:

- **Modelagem e uso de agentes na análise de texto:** Estruturar um sistema multi-agente para recuperar e processar avaliações.
- **Implementação técnica:** Desenvolver um pipeline que utilize agentes para processar dados não estruturados.
- **Qualidade e precisão:** Avaliar se as pontuações geradas pelos agentes são coerentes com as avaliações.

### Cenário

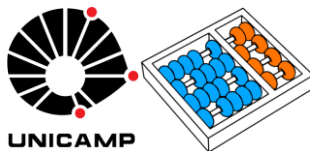
O arquivo **restaurantes.txt** contém avaliações qualitativas de diferentes restaurantes. Cada linha segue o formato:

***"<nome\_do\_restaurante>. <avaliação>"***

Exemplo:

***"Santo Pão. Sanduíches e sopas de boa qualidade, com ingredientes frescos. Atendimento eficiente e satisfatório."***

Seu sistema deve processar essas avaliações e responder automaticamente perguntas como: "Quão bom é o Santo Pão como restaurante?".



# Atividades

## 1. Recuperação de Avaliações

- Implementação da função **fetch\_restaurant\_data**
  - Recuperar avaliações associadas a um restaurante.
  - Retornar um dicionário com o nome do restaurante como chave e uma lista de avaliações como valor.

**Exemplo** **de** **saída:**  
**{ "Santo Pão": ["Sanduíches e sopas de boa qualidade, com ingredientes frescos.", "Atendimento eficiente e satisfatório."] }**

## 2. Cálculo da Pontuação Geral

- Implementação da função **calculate\_overall\_score**
  - Calcular a pontuação do restaurante com base nos escores extraídos.
  - Utilizar a fórmula:
    - $SUM(\sqrt{food\_scores[i]^2 * customer\_service\_scores[i]}) * 1/(N * \sqrt{125})) * 10$
  - A pontuação final deve ter exatamente três casas decimais.

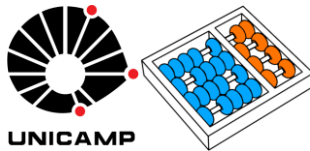
**Exemplo** **de** **saída:**  
**{ "Santo Pão": 7.542 }**

## 3. Análise e Extração de Pontuações com Agentes

- Desenvolvimento de agentes para análise das avaliações. Sugerimos que três agentes são suficientes para resolver o problema, mas sinta-se à vontade para criar quantos forem necessários. Considere o seguinte:
  1. Criar o agente **data\_fetch\_agent** que seja responsável por recuperar avaliações e estar ligado à função **fetch\_restaurant\_data**.
  2. Criar um agente **review\_analysis\_agent** que analisa as avaliações e converte adjetivos em escores conforme a seguinte escala (**não modificar esta escala, pois ela afeta o cálculo final da pontuação**):
    - a. **1/5**: horrível, nojento, terrível.
    - b. **2/5**: ruim, desagradável, ofensivo.
    - c. **3/5**: mediano, sem graça, irrelevante.
    - d. **4/5**: bom, agradável, satisfatório.
    - e. **5/5**: incrível, impressionante, surpreendente.
  3. Criar um agente **score\_agent** que seja responsável pelo cálculo final da pontuação, vinculado à função **calculate\_overall\_score**.

## 4. Implementação do Fluxo de Conversação com Agentes

- Criação do pipeline completo de comunicação entre os agentes



- Configurar o fluxo de execução entre os agentes usando **initiate\_chats**.
- Garantir que os agentes sejam chamados/orquestrados na sequência correta:
  1. O usuário fornece uma consulta sobre um restaurante.
  2. O **data\_fetch\_agent** obtém as avaliações.
  3. O **review\_analysis\_agent** analisa e converte avaliações em scores numéricos.
  4. O **score\_agent** gera a pontuação final.
  5. O sistema retorna a resposta ao usuário.

## Arquivos Fornecidos

1. **Restaurantes.txt**: Este arquivo é o conjunto de dados.
2. **main.py**: Este arquivo deve ser utilizado como template de base para implementar a solução.
3. **teste.py**: Contém testes pré-definidos que validam a implementação do sistema. A execução deste arquivo deve gerar a seguinte saída esperada:

**Teste 1 Passou. Esperado:** 3.79 Consulta: Qual é a avaliação média do Bob's?  
**Teste 2 Passou. Esperado:** 6.19 Consulta: Qual é a avaliação média do Paris 6?  
**Teste 3 Passou. Esperado:** 4.64 Consulta: Quão bom é o restaurante KFC?  
**Teste 4 Passou. Esperado:** 4.64 Consulta: Qual é a avaliação média do China in Box?  
4/4 Testes Passaram

## Exemplo de Execução

Para ilustrar o funcionamento do sistema, considere a seguinte execução:

1. **python main.py "Qual é a avaliação média do Bob's"?**
2. O **data\_fetch\_agent** recupera as avaliações do Bob's a partir do arquivo **restaurantes.txt**.
3. O **review\_analysis\_agent** converte os adjetivos das avaliações em scores numéricos com base na escala fornecida (veja atividade 3).
4. O **score\_agent** calcula a pontuação final do Bob's e retorna a resposta ao usuário.
5. O sistema exibe e apresenta a resposta final formatada: **A avaliação média do Bob's é 3.79.**

## Submissão do Desafio

- Suba os arquivos da solução construída em um repositório público gitlab ou github;
- No formulário de candidatura, insira a URL do seu repositório público contendo a solução construída.