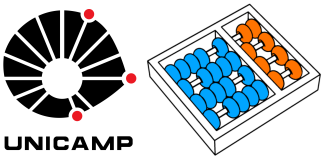

**UNIVERSIDADE ESTADUAL DE CAMPINAS**
**CURSO INF0083 - TECNOLOGIAS AVANÇADAS EM IA**
**DISCIPLINA INF0084 - Sistemas Inteligentes e Técnicas Avançadas em IA**

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

**Monitor:** Fillipe dos Santos Silva [f212148@dac.unicamp.br]

**Apoio Extraoficial:** Seyed Jamalaldin Haddadi [seyed@unicamp.br]

# Tarefa 02: RAG Multimodal

## Objetivo

Relatórios e documentações frequentemente incluem imagens ricas em informações incluindo textos, gráficos e diagramas contendo múltiplos pontos de interesse e detalhes contextuais relevantes que podem ser extraídos. Um pipeline que você desenvolver é relevante capturar e interpretar com precisão essas nuances sobre multimodalidades para incorporar as informações de forma eficaz. Um aspecto fundamental é garantir a consistência entre diferentes modalidades. Por exemplo, ao trabalhar com um documento, a representação semântica de um gráfico deve estar alinhada com o texto correspondente para manter a coerência e a precisão.

**RAG multimodal (MRAG)** é um framework avançado de IA que aprimora as capacidades dos sistemas tradicionais de RAG (*Retrieval-Augmented Generation*) ao incorporar múltiplas modalidades de dados, como texto, imagens, áudio e vídeo. Essa integração permite que o sistema recupere e gere informações em diversos formatos, resultando em respostas mais completas e contextualmente ricas.

Em uma pipeline de RAG multimodal, diferentes tipos de dados são transformados em um formato estruturado, como vetores, que o modelo pode processar. Esses vetores podem ser armazenados em um espaço vetorial unificado, permitindo que o modelo recupere informações relevantes independentemente do tipo de dado original de entrada. Por exemplo, uma consulta do usuário pode estar em formato de texto, mas o sistema pode recuperar informações pertinentes de imagens ou áudio para gerar uma resposta mais informada e relevante ao usuário.

A implementação do RAG multimodal pode ser feita com base em diferentes abordagens:

1. **Modelo Multimodal Único:** Um modelo unificado é treinado para codificar diversos tipos de dados em um espaço vetorial comum, permitindo a recuperação e geração

de informações de maneira integrada entre diferentes modalidades. Esta abordagem está sendo usada nesta tarefa.
2. **Modalidade Baseada em Texto:** Todos os tipos de dados são primeiramente convertidos em descrições textuais antes da codificação vetorial e armazenamento da mesma. Esse método aproveita os pontos fortes dos modelos baseados em texto, mas pode envolver alguma perda de informação durante o processo de conversão.
3. **Múltiplos Codificadores:** Modelos separados são usados para codificar diferentes tipos de dados, e os resultados são integrados durante o processo de recuperação. Essa abordagem permite uma codificação especializada, mas aumenta a complexidade na gestão de múltiplos modelos.

Nesta atividade, você desenvolverá um sistema **RAG Multimodal** capaz de processar e recuperar informações tanto de documentos textuais quanto de imagens. O sistema utilizará: 1) *embeddings* para indexar documentos; e 2) modelos generativos para produzir respostas com base nos documentos recuperados.

Os principais aspectos avaliados nesta atividade incluem:

1. **Compreensão dos conceitos de RAG multimodal:**
    - Avaliar a capacidade de entender e aplicar técnicas de *Retrieval-Augmented Generation* em diferentes modalidades de dados.
2. **Habilidades de implementação técnica:**
    - Verificar a competência na execução de etapas práticas, como processamento de diferentes tipos de arquivos, indexação, recuperação e geração de respostas.
3. **Análise crítica e relevância dos resultados:**
    - Avaliar a qualidade das respostas geradas pelo modelo, considerando a precisão e a contextualização das respostas às consultas dos usuários.

# Cenário e Instruções

Nosso objetivo final é construir um sistema **RAG multimodal** capaz de processar dados de entrada que incluam **texto e imagens**, permitindo que ele responda efetivamente a perguntas com base na base de conhecimento fornecida. Para esta tarefa, você receberá como materiais de apoio para sua execução:

- Dois arquivos HTMLs contendo texto e imagens. Você deve pré-processar esses arquivos HTMLs e gerar embbedings usando o **CLIP *embedder***. Esses embeddings serão utilizados na próxima etapa.
- Três arquivos de código.
    - Um te ajudará na preparação dos dados [*Data_Prep_MRAG_Assignment_incomplete*];
    - O outro na implementação do RAG multimodal [*MRAG_Assistant_Assinment_Incomplete*]. Voce deve **criar um Assistente RAG Multimodal** para responder perguntas sobre o conteúdo dos arquivos de entrada. Você pode utilizar o **Ollama** e o modelo **llama3.2-vision** para processar as consultas nesta etapa.

- Adicionalmente, fornecemos um **arquivo de funções auxiliares** *[functions.py]* que pode ser útil para essa tarefa. Você pode importá-lo como uma biblioteca no seu **passo de processamento de dados** para facilitar a implementação.

# Atividades

1. **Processamento de Dados [3,0 pontos]**
   - Carregar os documentos HTMLs.
   - Utilizar o modelo **CLIP** para gerar embbedings de texto e imagens para representá-los como vetores.
   - Armazenar as saídas processadas em um arquivo **JSON** para uso posterior. Um JSON com os embeedings de texto e um JSON com os embbedings de imagens.
2. **Carregar Conteúdos e *Embeddings* [2,0 pontos]**
   - Carregar os arquivos **JSON** e os embeddings relacionados aos textos e imagens.
3. **Recuperação de Documentos a partir de Consultas [3,0 pontos]**
   - Implementar um sistema de recuperação com base nos documentos vetorizados que permita aos usuários **recuperar documentos relevantes**.
4. **Geração de Respostas Baseada nos Documentos Recuperados [2,0 pontos]**
   - Usar o **Ollama** para chamar seu **MLLM** (por exemplo, **llama3.2-vision**) e gerar respostas com base nos documentos recuperados.
   - Avaliar os resultados utilizando diferentes valores de ***k*** (k=5, 7, 10) e **temperaturas**. Por exemplo, analisar quando o valor de *k* é alterado, se a geração de resposta melhora.

# Submissão

- Esta tarefa pode ser realizada individualmente ou em dupla.
- Dois arquivos devem ser entregues na submissão (por pessoa/dupla).
- Apenas um integrante da equipe deve submeter os Notebook Jupyter (**.ipynb**) contendo códigos completos e organizados para cada atividade. Deixe sua solução o mais documentada possível.
- O arquivo deve ser nomeado da seguinte forma:
  - tarefa02-rag-multimodal-dataprep<nome_dos_integrantes>.ipynb [Exemplo: tarefa02-rag-multimodal-dataprep-rafael-juliana.ipynb];

    - tarefa02-rag-multimodal-assistant<nome_dos_integrantes>.ipynb [Exemplo: tarefa02-rag-multimodal-assistant-rafael-juliana.ipynb];
- Esta entrega tem peso de **30%** da nota final do módulo.
- A entrega deve ser feita até **05/03/2025 (Quarta-feira)** às 23:59 via classroom.

## Critérios de Avaliação

- **Rigor Conceitual:** Os conceitos e técnicas esperados foram entendidos e usados adequadamente?
- **Completude:** Todas as atividades propostas foram realizadas?
- **Relevância:** Os documentos recuperados e as respostas geradas são apropriados e bem fundamentados?
- **Qualidade do Código:** O código é claro, eficiente e bem comentado?