

Inteligência Artificial 2

Aula 12 – Parte 2: Experimento de Chunking e Avaliação de Recuperação

Prof. Aléssio Miranda Júnior
alessio@cefetmg.br

CEFET-MG - Campus Timóteo
Dep. Engenharia de Computação

Inteligência Artificial 2

- 1 Contexto e Hipótese Experimental
- 2 Tokenização e Fronteiras de Corte
- 3 As Cinco Estratégias de Fatiamento
- 4 Protocolo Científico sem Vazamento
- 5 As Quatro Métricas de Recuperação
- 6 Diagnóstico com Clusters de Embeddings
- 7 Recapitulação e Prática Laboratorial

Onde Paramos? (Reconexão com a Parte 1)

Na primeira sessão da Aula 12, construímos os alicerces da ingestão documental:

- **Ingestão de Documentos Brutos:** Extração de strings a partir de PDFs corporativos nativos e escaneados (OCR).
- **Governança Estrita:** Criação de `document-manifest.jsonl` com metadados obrigatórios (`owner`, `license`, `contains_pii`, `access_scope`).
- **Proveniência Auditável:** Cada texto extraído foi associado ao arquivo, hash SHA-256, versão e página física de origem.

O Abismo Aberto para a Parte 2

Temos os textos auditados na memória. Porém, relatórios e manuais possuem centenas de páginas. Um vetor único para o documento inteiro dilui o sinal semântico e estoura orçamentos de geração.

Como fatiar esse texto em unidades recuperáveis e provar cientificamente qual corte funciona melhor?

Objetivos da Sessão (Parte 2)

Ao final desta aula expositiva e prática, você será capaz de:

1. **Formular** a escolha de fatiamento (*chunking*) como uma hipótese experimental mensurável.
2. **Dominar** as 5 estratégias algorítmicas de corte: Fixo, Recursivo, Estrutural, Semântico e Hierárquico.
3. **Blindar** a esteira contra *Data Leakage* via separação estrita entre conjuntos de Desenvolvimento e Held-Out.
4. **Calcular e interpretar** as 4 métricas essenciais de recuperação: Recall@K, MRR, Precisão Contextual e Redundância.
5. **Construir** matrizes multicritério ponderando precisão, custo computacional e diversidade informacional.
6. **Diagnosticar** anomalias e duplicatas do corpus via clusters de embeddings sem cair em circularidade.

Por que o Chunking Define o Sucesso do RAG?

Na arquitetura RAG (*Retrieval-Augmented Generation*), o gerador só responde com precisão se o recuperador entregar o contexto exato:

Chunks Grandes Demais (> 1000 tok)

- **Diluição Semântica:** O vetor captura o tema geral, mas perde detalhes específicos.
- **Custo Elevado:** Envia texto irrelevante no prompt, inflando a fatura de tokens.
- **Lost in the Middle:** O LLM ignora evidências no meio de contextos longos.

Chunks Pequenos Demais (< 100 tok)

- **Perda de Contexto:** A frase fica isolada sem o sujeito ou sem a condição de exceção.
- **Fragmentação Excessiva:** Multiplica o número de vetores e a latência de busca.
- **Falso Casamento:** Palavras-chave isoladas geram falsos positivos semânticos.

A Formulação Científica do Experimento

Não existe tamanho de chunk “universal” (ex: a lenda urbana de que “512 tokens sempre funciona”).

- Textos jurídicos têm orações longas e estruturas hierárquicas rígidas (Artigos, Parágrafos, Incisos).
- Manuais técnicos possuem tabelas numéricas e listas curtas de passos operacionais.
- Noticiários e artigos científicos exigem granularidades completamente distintas.

A Hipótese Experimental

“Para este corpus específico e este conjunto de perguntas de negócio, qual estratégia de corte recupera evidência citável com máximo contexto útil e mínima redundância?”

Variáveis de Controle

Mantenha estritamente congelados: o corpus de documentos, o modelo de embedding, o índice vetorial e o Top- K ($K = 3$).

Varie apenas: a estratégia algorítmica de corte, o tamanho e a sobreposição (*overlap*).

O Orçamento em Tokens e o BPE

- O orçamento do fatiador deve usar o **tokenizer real do modelo de embedding**, jamais caracteres ou palavras aproximadas.
- Modelos modernos utilizam algoritmos de subwords como o **BPE (Byte-Pair Encoding)**:

Palavra Bruta:

‘‘Inconstitucionalissimamente’’ (27 letras)

Subwords BPE:

In

constitucional

issima

mente

⇒ 4 Tokens!

Impacto Prático no Português

Números, datas (“05/10/2026”), pontuações e acentos dividem palavras em múltiplos fragmentos. Um chunk de “300 caracteres” pode ter de 45 a 120 tokens dependendo do vocabulário!

A Matemática da Sobreposição (Chunk Overlap)

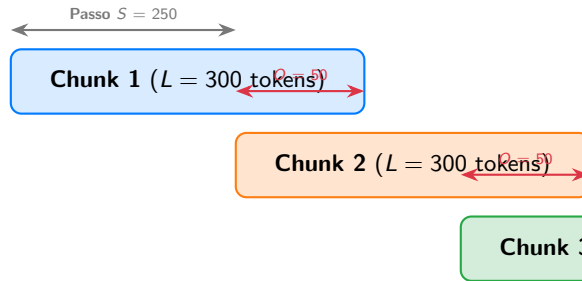
- Se o tamanho do chunk é L tokens e o overlap é O tokens, o **passo de deslocamento** (*stride*) é:

$$S = L - O$$

- O número total de chunks gerados para um documento com T tokens é aproximadamente:

$$N_{\text{chunks}} \approx \left\lceil \frac{T - O}{L - O} \right\rceil$$

- Se $O \rightarrow 0$: fronteiras rígidas, risco de quebra.
- Se $O \rightarrow L$: explosão combinatória de



Estratégia 1: Corte Fixo (Fixed Window)

- **Mecanismo:** Janela mecânica estrita a cada S tokens, sem qualquer análise sintática.
- **Vantagem:** Implementação trivial, determinística e ultrarrápida.
- **O Risco da Guilhotina:**
 - Corta números de telefone, tabelas e equações.
 - Separa o sujeito do predicado.
 - Isola o texto de uma regra da sua cláusula de exceção.

Exemplo de Guilhotina

“...o aluno será reprovado por falta se frequência $< 75\%$ **[CORTE CEGO]** salvo se apresentar atestado médico homologado...” A pergunta: “*Atestado médico abona falta?*” falhará porque o Chunk 1 diz apenas que reprova, e o Chunk 2 perdeu o contexto da reprovação!

Estratégia 2: Fatiador Recursivo (Separadores)

O algoritmo busca preservar a hierarquia natural do discurso humano antes de impor um corte



Dinâmica de Execução

- 1) Tenta manter o parágrafo inteiro; se exceder o orçamento, divide por quebras de linha.
- 2) Se a linha for grande demais, divide em sentenças completas; se a sentença for monstruosa, corta por espaços.

⇒ **É o padrão industrial mais estável para corpora genéricos!**

Estratégia 3: Chunking Estrutural (Sintaxe da Fonte)

- **Princípio:** Usa as marcações e a semântica editorial do próprio documento:
 - Cabeçalhos de Markdown (#, ##, ###).
 - Artigos de Leis (Art. 1º, Parágrafo Único).
 - Tags HTML (<section>, <article>).
 - Tabelas inteiras formatadas.
- **Proveniência Multiuso:**
 - Se uma seção ocupa 3 páginas, o metadado deve registrar explicitamente:
page_start: 3, page_end: 5.

Condição de Validade

Exige que a extração da Parte 1 tenha preservado a hierarquia visual.

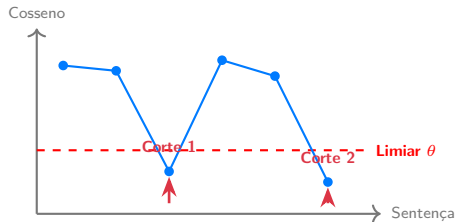
Cuidado: Em relatórios scaneados com OCR degradado, tentar inferir estrutura inventa títulos falsos!

Estratégia 4: Chunking Semântico por Similaridade

- Em vez de contar tokens cegamente, o fatiador calcula o embedding de cada sentença individual s_i .
- Mede a similaridade de cosseno entre sentenças vizinhas:

$$\text{sim}(s_i, s_{i+1}) = \frac{\vec{s}_i \cdot \vec{s}_{i+1}}{\|\vec{s}_i\| \|\vec{s}_{i+1}\|}$$

- Quando a similaridade cai abaixo de um limiar θ (um “vale” semântico), o algoritmo inicia um novo chunk!



Custo Computacional

Alta precisão temática, porém exige gerar N embeddings na ingestão (mais caro e lento).

Estratégia 5: Chunking Hierárquico (Small-to-Big)

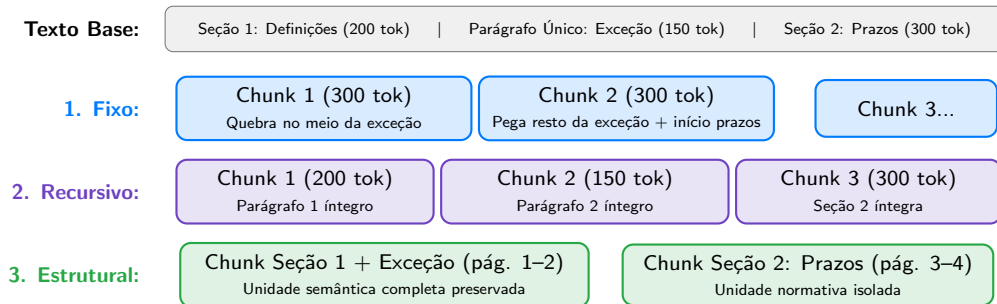
O estado da arte na engenharia industrial de RAG (*Parent Document Retrieval*):



O Segredo do Mecanismo

O ChromaDB busca no vetor do **Filho 2** (vetor pequeno, nítido e altamente discriminativo). Porém, na hora de responder, o sistema puxa do banco o **Chunk Pai** inteiro, entregando contexto farto sem alucinação!

Comparação Visual: 3 Cortes no Mesmo Texto



O Erro Clássico: Vazamento de Dados (Data Leakage)

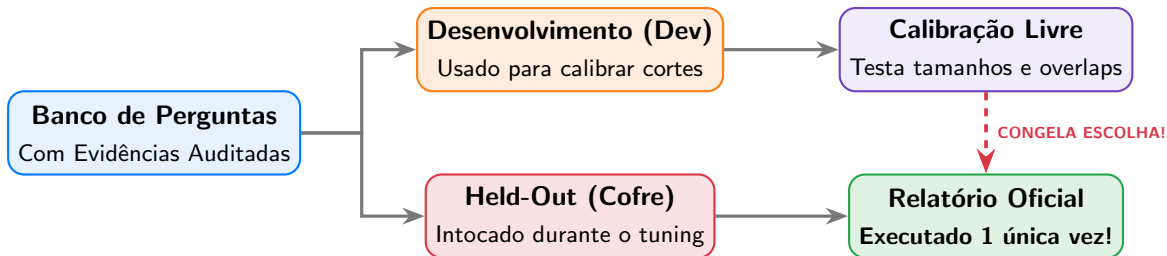
O Cenário da Ilusão

1. O desenvolvedor faz a pergunta: "*Como funciona o exame final?*"
2. O retorno falha porque o chunk cortou na metade.
3. Ele altera o código para `chunk_size=450` até a resposta aparecer no Top-1.
4. Conclusão ingênua: "Nosso RAG atingiu 100% de acurácia!".

A Realidade em Produção

- O modelo sofreu **overfitting** nas perguntas conhecidas de teste.
- Quando o usuário final fizer uma pergunta inédita, o sistema falhará silenciosamente.
- Ajustar hiperparâmetros na mesma amostra usada para validar o modelo é **fraude metodológica em IA**.

Metodologia Rigorosa: Desenvolvimento vs. Held-Out



Regra de Ouro da Disciplina

Se o resultado no held-out for inferior ao esperado, **reporte a falha e analise a causa**.
Jamais volte ao código para alterar o fatiador visando o held-out!

Anatomia de uma Consulta com Ground Truth

Cada caso de teste experimental possui o texto da consulta e as **páginas ou seções verificadas previamente por humanos**:

```
# Conjunto de Desenvolvimento (Tuning de Estrategia)
desenvolvimento = [
    {"query": "qual e a carga horaria minima?", "expected_pages":
        {3}},
    {"query": "quais os pre-requisitos de IA 1?", "expected_pages":
        {5}},
]

# Conjunto Held-Out (Cofre Fechado para Avaliacao Final)
held_out = [
    {"query": "como funciona o criterio de avaliacao?", "
        expected_pages": {7, 8}},
    {"query": "qual o procedimento para trancamento?", "
        expected_pages": {12}},
]
```

Indexação Citável no ChromaDB

Para que a avaliação seja confiável, cada estratégia deve ser gravada com seus metadados de rastreabilidade:

```
def indexar(collection, chunks):  
    collection.upsert(  
        ids=[f"{c['chunking']}:{i}" for i, c in enumerate(chunks)],  
        documents=[c["text"] for c in chunks],  
        metadatas=[{  
            "source_file": c["source_file"],  
            "source_sha256": c["source_sha256"],  
            "page_start": c.get("page", c.get("page_start")),  
            "page_end": c.get("page", c.get("page_end")),  
            "chunking": c["chunking"],  
            "pipeline_version": c["pipeline_version"],  
        } for c in chunks],  
    )
```

Observe: a coleção suporta filtros vetoriais compostos preservando a linhagem do dado desde o PDF bruto.

O Cardápio de Métricas em Information Retrieval

Por que não usamos a “Acurácia” clássica de Classificação em RAG?

- O banco vetorial não dá uma resposta binária; ele devolve um **ranking ordenado** de K documentos.
- A pergunta não é apenas “se encontrou”, mas sim:
 1. **Recall@K**: A informação correta está presente nos K primeiros?
 2. **MRR**: Em qual posição do ranking ela foi entregue ao LLM?
 3. **Precisão Contextual**: O texto recuperado realmente sustenta a resposta?
 4. **Redundância**: Os trechos do Top- K são diversos ou cópias repetitivas?

O Efeito Cascata

Uma falha nessas métricas contamina toda a geração: o LLM alucina, o custo sobe e o usuário perde a confiança no sistema.

Métrica 1: Recall@K (O Filtro de Existência)

- **Pergunta:** *A evidência correta entrou no Top- K retornado pelo banco?*
- Para uma consulta q com $K = 3$:

$$\text{Recall@3}(q) = \begin{cases} 1, & \text{se } \text{Top-3} \cap \text{Evid} \neq \emptyset \\ 0, & \text{caso contrário} \end{cases}$$

- Média no corpus de N perguntas:

$$\overline{\text{Recall@3}} = \frac{1}{N} \sum_{i=1}^N \text{Recall@3}(q_i)$$

O Papel do Recall

O Recall é o teto máximo de acurácia do seu sistema RAG.

Se $\text{Recall@3} = 0$, o gerador LLM sequer terá acesso à informação verdadeira →

Alucinação inevitável!

Métrica 2: MRR (Mean Reciprocal Rank)

Por que apenas Recall@3 não é suficiente?

Estar na 1ª posição é drasticamente melhor do que estar na 3ª posição!

- LLMs sofrem de viés posicional (*Lost in the Middle*): prestam mais atenção ao início do contexto.
- Colocar a evidência no topo reduz custo de raciocínio e alucinações.

Definição Matemática do Reciprocal Rank (RR)

Se a primeira evidência correta aparece no rank $r \in \{1, 2, \dots, K\}$:

$$\text{RR}(q) = \frac{1}{r}, \quad \text{se não estiver no Top-}K \implies \text{RR}(q) = 0$$

O **MRR** é a média harmônica das posições em todo o conjunto de consultas:

$$\text{MRR} = \frac{1}{N} \sum_{i=1}^N \text{RR}(q_i)$$

Exemplo Numérico Passo-a-Passo de MRR

Considere um experimento com 4 consultas avaliadas no Top-3:

Consulta	Pergunta	Posição 1ª Evidência	Recall@3	Reciprocal Rank (RR)
q_1	Carga horária	Rank 1	1	$1/1 = 1,00$
q_2	Avaliação final	Rank 2	1	$1/2 = 0,50$
q_3	Trancamento	Rank 3	1	$1/3 \approx 0,33$
q_4	Intercâmbio	Não retornou (Rank > 3)	0	0,00

Cálculo do Recall@3

$$\overline{\text{Recall@3}} = \frac{1 + 1 + 1 + 0}{4} = \mathbf{0,75} \quad (75\%)$$

Cálculo do MRR

$$\text{MRR} = \frac{1,00 + 0,50 + 0,33 + 0,00}{4} = \frac{1,83}{4} = \mathbf{0,46}$$

O Perigo do Falso Positivo

Um chunk pode vir da página 7 correta, mas conter apenas o rodapé ou os agradecimentos! O script marcou “sucesso” por página, mas o conteúdo real não sustenta a resposta.

Auditoria por Leitura Humana

Dos 3 trechos retornados no Top-3, quantos realmente trazem os argumentos necessários para responder à pergunta?

$$\text{PrecCtx} = \frac{\text{Chunks Relevantes no Top-3}}{3}$$

- Se o Top-3 traz 1 chunk bom e 2 irrelevantes: $\text{PrecCtx} = 1/3 = 0,33$.
- Trechos irrelevantes desperdiçam a janela de contexto e confundem a atenção do modelo gerador.

Métrica 4: Redundância e a Armadilha do Overlap

O que acontece com overlap excessivo?

- Se o chunk tem 300 tokens com overlap de 200:
- O Top-3 retornado pelo ChromaDB pode ser:
 - Chunk 1: [Tokens 0 a 300]
 - Chunk 2: [Tokens 100 a 400]
 - Chunk 3: [Tokens 200 a 500]
- **Problema:** Os 3 chunks repetem exatamente a mesma frase!

Custo da Redundância

Você pagou por 3 chunks no prompt, mas entregou a mesma informação 3 vezes em vez de 3 perspectivas complementares!

Cálculo Prático

Calcula-se a similaridade média de cosseno entre os pares do Top-3:

$$\text{Redundância} = \frac{\cos(v_1, v_2) + \cos(v_1, v_3) + \cos(v_2, v_3)}{3}$$

Valores $> 0,85$ denunciam chunks duplicados causados por sobreposição agressiva.

A Matriz de Decisão Multicritério

Após rodar as consultas held-out, consolide a matriz experimental:

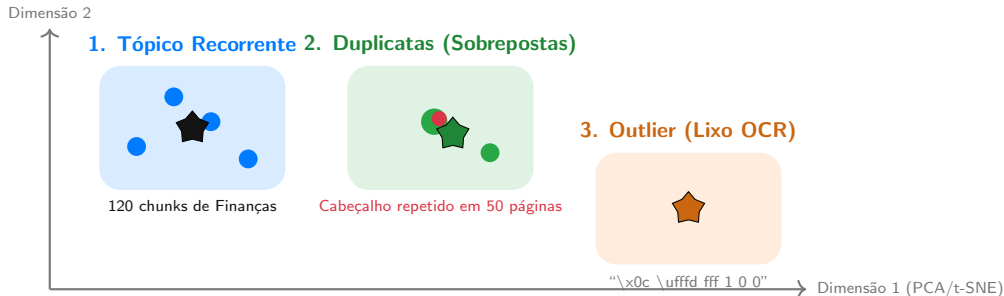
Estratégia	Recall@3	MRR	Prec. Contextual	Redundância	Diagnóstico Técnico
Fixa (300/50)	0,67	0,44	0,33	0,65	Cortes cegos quebram fronteiras
Recursiva (300/50)	0,83	0,61	0,67	0,42	Equilíbrio sólido em prosa geral
Estrutural	0,83	0,72	0,78	0,21	Campeã se a extração for limpa

Critério de Desempate da Engenharia

Em caso de empate em Recall@3 e MRR, escolha a estratégia que: **1)** Minimizar redundância, **2)** Preservar proveniência de seções e **3)** Gere menor custo de embedding.

Diagnóstico 1: Tópicos, Duplicatas e Anomalias

Auditoria Não-Supervisionada: O k -Means revela a estrutura interna da base documental:



Valor para o Engenheiro

Permite expor documentos repetitivos e falhas grotescas de extração **antes** de gastar dinheiro indexando no banco.

Diagnóstico 2: Inspeção de Centróides e Nomes Provisórios

- **Centróide é um Vetor Médio:**

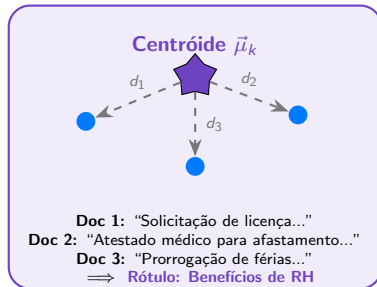
$$\vec{\mu}_k = \frac{1}{|C_k|} \sum_{\vec{v} \in C_k} \vec{v}$$

Ele não é um texto real, mas um ponto de gravidade no espaço $\mathbb{R}^d$.

- **Como Inspeccionar:** Buscamos os trechos reais **mais próximos do centróide**:

$$\arg \min_{\vec{v}_i \in C_k} \|\vec{v}_i - \vec{\mu}_k\|$$

- Lemos os 3 primeiros trechos e damos um **nome provisório** ao cluster.



Diagnóstico 3: Hiperparâmetros (K e Normalização L2)

O resultado do clustering depende criticamente de dois fatores técnicos:

1. A Escolha do Número de Grupos (K)

- **K pequeno demais ($K = 2$):** Mistura assuntos distintos (ex: contratos e manuais de TI viram o mesmo grupo).
- **K grande demais ($K = 30$):** Quebra o mesmo tópico em micro-grupos irrelevantes.
- **Solução:** Inspeção o Coeficiente de Silhueta (*Silhouette Score*) e a inércia do cotovelo (*Elbow Method*).

2. A Normalização L2 ($\|\vec{v}\| = 1$)

O k -Means minimiza a **distância Euclidiana**. Para que ele respeite a geometria de **coseno**:

$$\|\vec{u} - \vec{v}\|^2 = 2 - 2 \cos(\vec{u}, \vec{v})$$

Obrigatório: Sempre normalize os embeddings com `L2_norm` antes do k -Means! Sem isso, o tamanho do texto distorce o cluster.

O que um Cluster NÃO é (Cuidado Epistemológico!)

Cluster NÃO é Ground Truth!

Um cluster agrupa pontos por proximidade matemática na geometria do espaço vetorial, não por verdade de negócio ou intenção do usuário.

- **Proibição de Circularidade:** Jamais forneça as perguntas de teste ou as páginas esperadas como entrada para o algoritmo de clustering!
- **Supervisão é Sagrada:** Rótulos de teste pertencem exclusivamente à avaliação de métricas (Recall, MRR).
- **Cuidado Estético:** Não escolha uma estratégia de chunking apenas porque o gráfico t-SNE “parece mais bonito e colorido”. A decisão pertence ao Held-Out!

Auditoria de Anomalias com Clustering em Python

Como o engenheiro utiliza o clustering na vida real para depurar a base de dados:

```
from sklearn.cluster import KMeans
from sklearn.preprocessing import normalize
import numpy as np

# 1. Normalizacao L2 obrigatoria para simular distancia angular de
    cosseno
X_norm = normalize(embeddings, norm='l2')
kmeans = KMeans(n_clusters=5, random_state=42).fit(X_norm)

# 2. Identifica chunks com maior distancia euclidiana para seus
    centroides
distancias = np.min(kmeans.transform(X_norm), axis=1)
outliers_idx = np.argsort(distancias)[-3:] # Top-3 anomalias

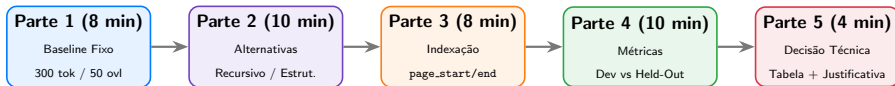
for idx in outliers_idx:
    print(f"ANOMALIA [Doc {idx}]: {chunks[idx]['text'][:80]}...")
```

Recapitulação — Os 7 Princípios da Ingestão de Dados

1. **Chunking é Experimental:** O corte ótimo depende do corpus e das perguntas de negócio.
2. **Tokens Reais Mandam:** O orçamento do splitter obedece ao tokenizer do modelo, não a caracteres.
3. **Filosofias de Corte:** Fixo (mecânico), Recursivo (linguístico), Estrutural (seções) e Hierárquico.
4. **Blindagem Metodológica:** Dev Set calibra hiperparâmetros; Held-Out mede a performance final uma única vez.
5. **Quatro Métricas Irmãs:** Recall@K (teto), MRR (posição), Precisão Contextual (sustentação) e Redundância (diversidade).
6. **Trade-offs Reais:** Overlap excessivo gasta prompt e infla custos sem agregar valor semântico.
7. **Clusters são Diagnóstico:** Úteis para expor anomalias de OCR, nunca para substituir avaliação supervisionada.

Roteiro do Laboratório 12B (40 Minutos)

Missão: Implementar os 3 splitters, indexar com proveniência e medir o Top-3.



Mão no Código!

Vocês implementarão o cálculo do MRR e do Recall@3 em Python sobre o ChromaDB real.
Vamos ao laboratório!

Dica de Implementação para o Laboratório

Estrutura do laço avaliador que vocês implementarão no 12-lab-parte2.py:

```
def avaliar_recuperacao(collection, casos, k=3):
    recalls, reciprocal_ranks = [], []
    for caso in casos:
        resp = collection.query(query_texts=[caso["query"]],
                                n_results=k)
        paginas = [m["page_start"] for m in resp["metadatas"][0]]

        # Encontra em qual posicao a pagina esperada apareceu
        ranks = [i + 1 for i, pag in enumerate(paginas)
                  if pag in caso["expected_pages"]]

        recalls.append(bool(ranks))
        reciprocal_ranks.append(1.0 / ranks[0] if ranks else 0.0)

    return {"Recall_Top3": np.mean(recalls), "MRR": np.mean(
        reciprocal_ranks)}
```

O que Construimos no Módulo 2

Com a Aula 12, completamos toda a esteira da **Recuperação Semântica**:

- **Representação**: Palavras e textos projetados em espaços densos $\mathbb{R}^d$ (Embeddings).
- **Geometria**: Cosseno e buscas aproximadas em milissegundos (HNSW e ChromaDB).
- **Engenharia de Dados**: Extração auditada, fatiamento experimental e avaliação com métricas.

O Gancho Motivacional

Nosso banco vetorial agora recupera com maestria os melhores trechos de qualquer livro ou contrato corporativo.

Mas quem lê esses trechos recuperados e formula uma resposta brilhante com raciocínio contextual?

→ Na próxima aula: **A Revolução dos Transformers e os Mecanismos de Atenção!**

Dúvidas?

Email: alessio@cefetmg.br

Aula: Segunda e Quarta-Feira, 7h as 8:40h (Sala 39)

WhatsApp: Dúvidas rápidas

Lembrete

Confira sua Issue no GitLab!

Próxima aula: [Arquitetura Transformer e Mecanismos de Atenção](#)

100% da Aula 12 Concluída!