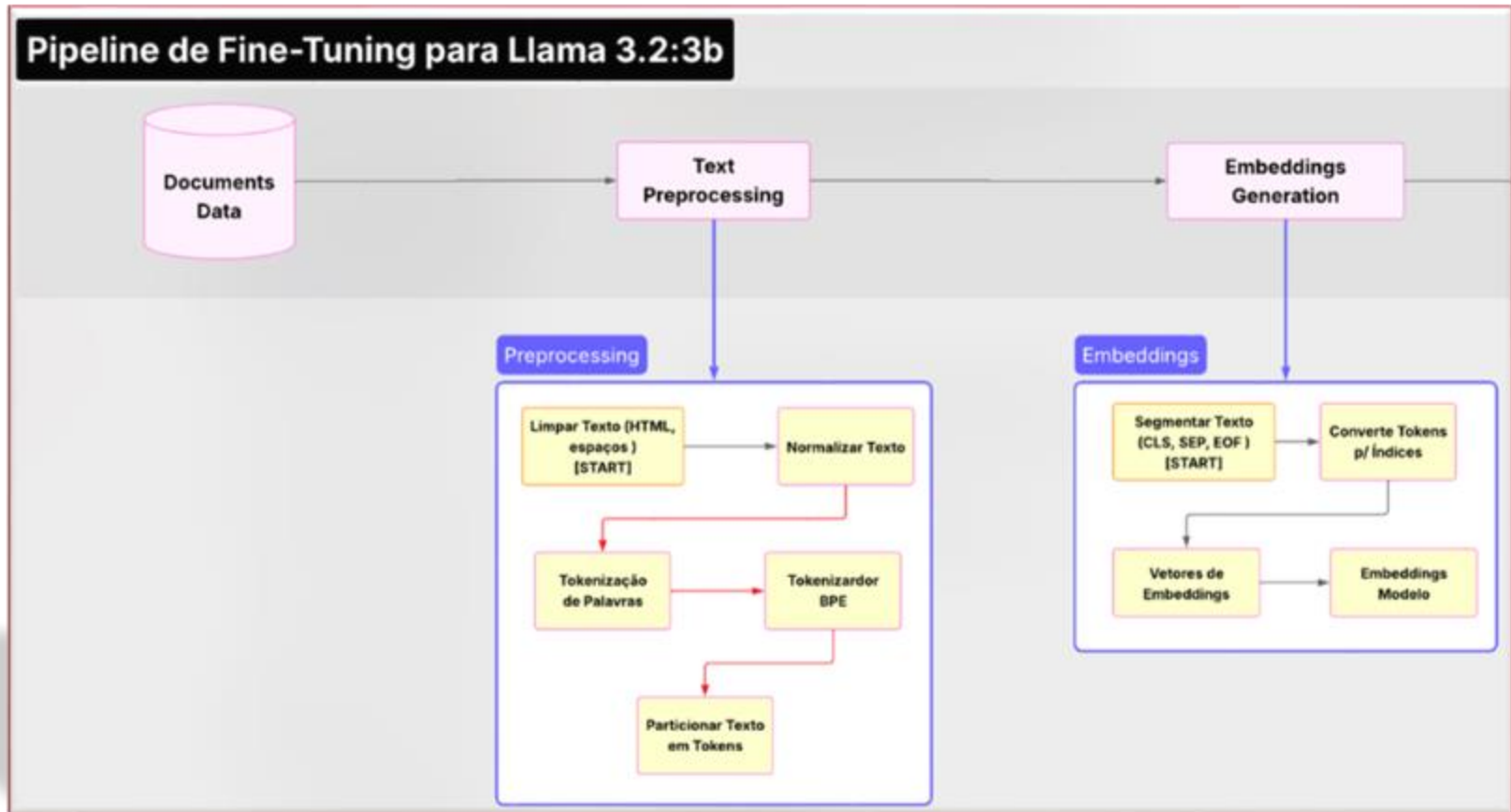




Grafo de Prioridade para Pré-Processamento de Texto

Desenvolvido por: *Gustavo Alexandre*

Motivação e Problema



Motivação e Problema


Pré-processamento de Texto exige:

- Limpeza textual e normalização linguística
- Tokenização e preparação para embeddings
- Pipelines sequenciais não escalam bem



Proposta da Solução

Arquitetura baseada em:

- Grafo de dependência acíclico (DAG)
 - Fila de prioridade para escalonamento
 - Execução paralela com múltiplas threads
- 

Metodologia e Arquitetura

Cinco módulos principais:

- Leitura e parsing dos dados
- Geração do grafo de tarefas
- Priorização com base em criticidade
- Execução paralela com sincronização
- Gerenciamento do pipeline

Resultados e Análise

Comparação de cenários:

- Particionamento: 5,10x speedup, 53.641 docs/s
- Scheduler: desempenho inferior ao sequencial
- Confirmação da Lei de Amdahl

Conclusões e Contribuições



- PARTICIONAMENTO DE DADOS É EFICAZ



- ARQUITETURA ESCALÁVEL E ROBUSTA



- GRAPH PRIORITY NO GITHUB