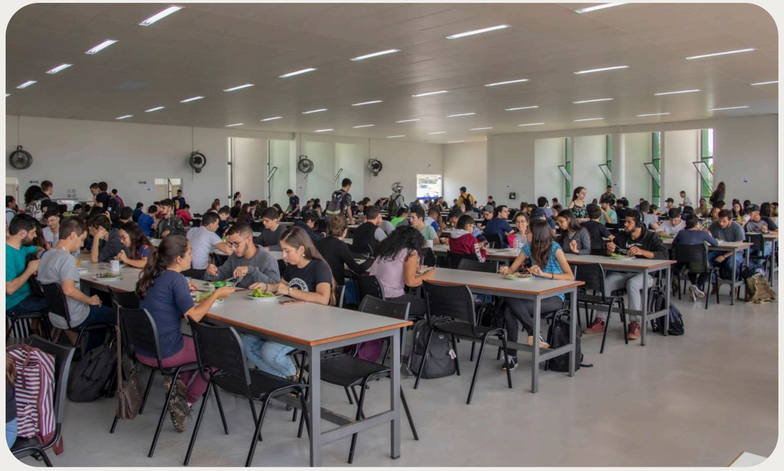
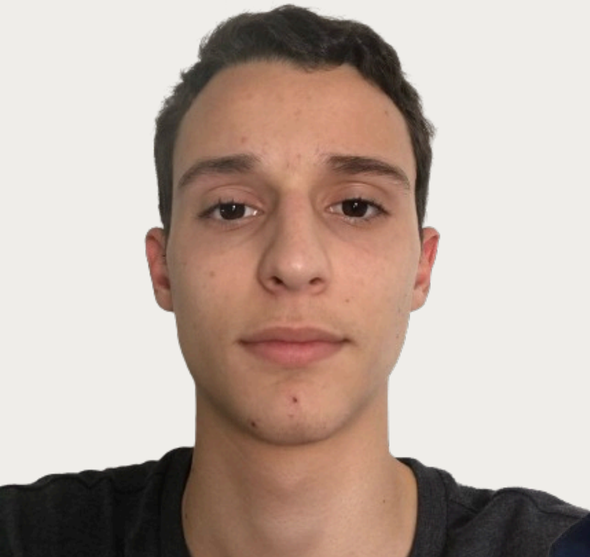
Projeto Final

Análise de Consumo no Restaurante Universitário

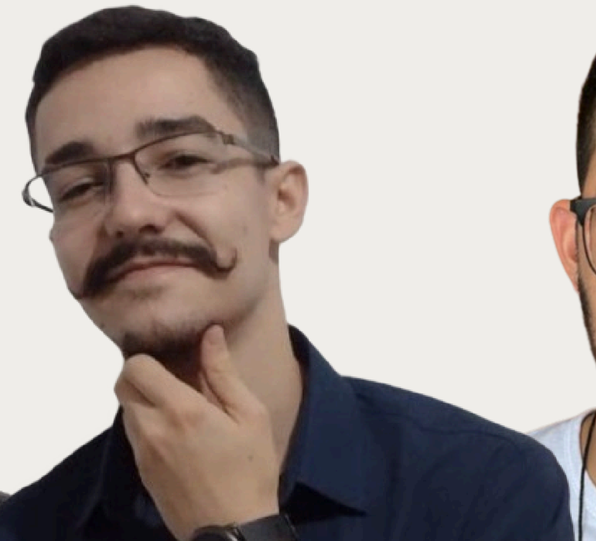


Integrantes

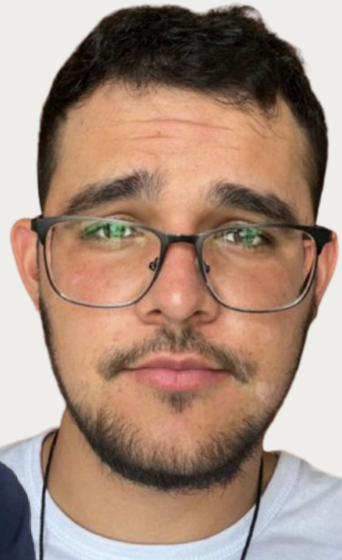
Diogo Carrer



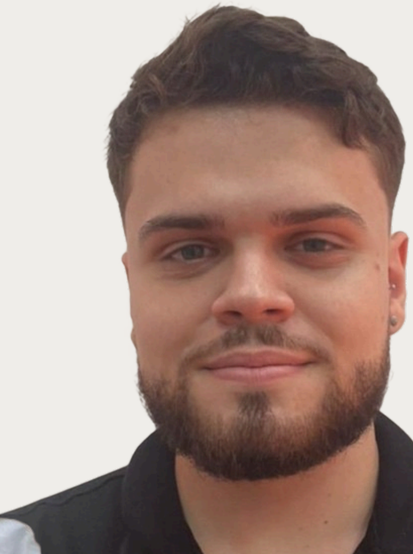
Jean Diniz



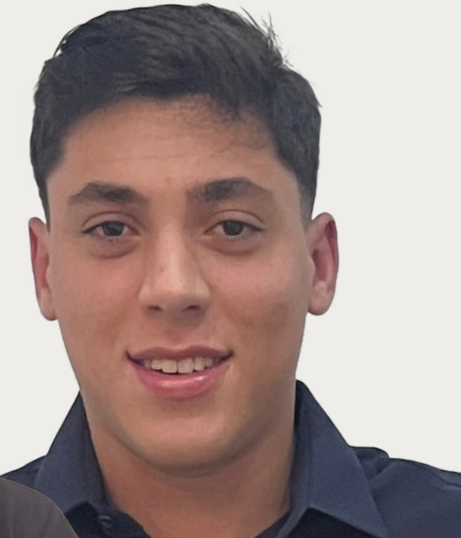
João Victor Carvalho



Rafael Rodrigues



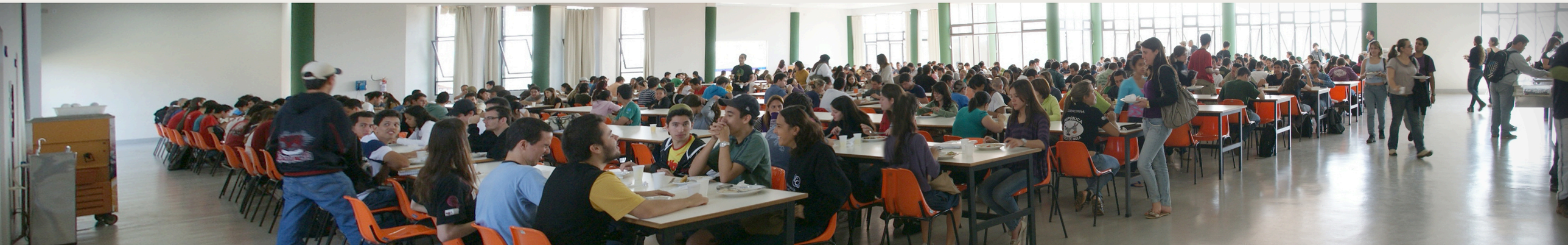
Vitor Pereira



1 Objetivos

Objetivos

- Identificar padrões temporais de consumo.
- Perfilar usuários por curso e vulnerabilidade social.
- Detectar comunidades de consumo similares.
- Gerar métricas para otimização do serviço.



2 Arquitetura Geral

Arquitetura

- Cluster Apache Spark (Master + workers).
- Container Docker com a aplicação analytics.
- Armazenamento em volumes: dados, resultados, métricas e logs.

3 Estrutura de Dados

Estrutura de Dados

- **Formato JSON por registro:**
  - documento, data_consumo, tipo_consumo, tipo_usuario, etc.

- **Diretórios principais:**
  - datasample/ (amostra).
  - misc/data/, misc/results/, misc/metrics/, misc/logs/.

4 Módulos Chave

Módulos Chave

- **config.py:** paths e tuning do Spark.
- **logging_config.py:** Loguru (rotacionamento, níveis).
- **data_analysis.py:** RUAnalyzer (estatísticas, grafo, comunidades).
- **experiments.py:** testes de performance e escalabilidade.
- **spark_measure_utils.py:** coleta de métricas via sparkMeasure.

5

Fluxo de Processo

Fluxo de Processo

1. Inicialização Docker → Spark Master/Workers.
2. Download (se preciso) e carregamento dos dados.
3. Filtragem / estatísticas / criação de grafo.
4. Detecção de comunidades (Louvain).
5. Armazenamento de resultados e métricas.

6 Execução & Deploy

Execução & Deploy

- **Local (Docker Compose):**
  - ./bin/run_project.sh up
  - ./bin/run_project.sh analyze sample|complete
  - ./bin/run_project.sh experiments complete performance

- **Spark Master UI:** (localhost:8080), **Spark App UI:** (localhost:4040).