

Projeto LÚMEN: Laboratório Unificado de Modelagem, Engenharia e Narrativas

Versão: 1.0 - "A Centelha da Gênese"

Autor: Eric Pimentel

Data de Criação: 24 de junho de 2025



"Onde há dados, há padrão. Onde há padrão, há informação. Onde há informação, há narrativa. E onde há narrativa, há o poder de transformar a realidade. LÚMEN é a nossa forja de narrativas."

Resumo Executivo (A Visão em Um Olhar)

O Projeto LÚMEN é um ecossistema de dados ponta-a-ponta, totalmente open-source e containerizado, projetado para ser uma plataforma robusta, escalável e reproduzível para análise de dados, engenharia de dados e experimentação com Machine Learning. Ele nasce da filosofia de que ferramentas de nível profissional não devem ser um privilégio corporativo, mas um poder acessível a todos os estudiosos e criadores.

Esta arquitetura integra Python, PostgreSQL, Metabase, Mage.ai e MLflow, orquestrados por Docker, para criar um ambiente que simula uma stack de dados moderna, ideal para portfólio, estudos avançados e desenvolvimento de projetos de impacto.

1. Visão e Missão (A Estrela Guia)

-  **Visão:** Ser um ecossistema de referência, modular e evolutivo, capaz de transformar dados brutos de qualquer fonte em sabedoria acionável, desde dashboards interativos a modelos preditivos inteligentes.
-  **Missão:** Construir e documentar um MVP (Minimum Viable Product) da plataforma LÚMEN, demonstrando um fluxo de dados completo: ingestão automatizada, armazenamento estruturado, transformação alquímica e visualização eloquente.

2. Objetivos do MVP (Os Primeiros Feixes de Luz)

Para que nosso MVP seja considerado um sucesso, os seguintes objetivos devem ser alcançados:

- **O1: [Infraestrutura como Código]** Ter toda a infraestrutura (PostgreSQL, Metabase, Mage) definida e orquestrada em um único arquivo `docker-compose.yml`, permitindo que todo o ambiente seja iniciado com um único comando.
- **O2: [Pipeline de Dados Funcional]** Implementar um pipeline de dados inicial no Mage que extraia dados de uma fonte pública (ex: dados meteorológicos da estação de Belém via API do INMET), realize uma limpeza básica com Python (Pandas/Polars) e armazene os dados brutos e processados em schemas distintos no PostgreSQL.
- **O3: [Narrativa Visual Inicial]** Criar um dashboard inicial no Metabase com pelo menos 3 a 4 visualizações (ex: série temporal de temperatura, medidor de umidade atual, gráfico de barras de precipitação) que consuma os dados da tabela processada.
- **O4: [Documentação exemplar]** Manter este documento como a fonte central da verdade e garantir que o repositório do projeto tenha um `README.md` claro, conciso e que guie o usuário na execução do projeto.

3. Arquitetura da Constelação LÚMEN (O Mapa Estelar)

A arquitetura é baseada em serviços desacoplados e especializados, comunicando-se através de uma rede Docker interna.

Fluxo de Dados Lógico do MVP:

[Fonte Externa: API do INMET]

|
v

[1. Mage.ai (Container)] --(Aciona o Pipeline Agendado)-->

| 1.1 Extrai os dados

|
v

[2. PostgreSQL (Container)] --(Armazena em schema `raw_data`)-->

|
v

[3. Script Python (Invocado pelo Mage)] --(Lê de `raw_data`)-->

| 3.1 Limpa, transforma, enriquece

|
v

[2. PostgreSQL (Container)] --(Salva em schema `processed_data`)-->

|
v

[4. Metabase (Container)] --(Usuário acessa o Dashboard)-->

| 4.1 Consulta `processed_data`

|
v

[Navegador do Usuário Final] --(Visualiza a narrativa de dados)-->

4. Plano de Ação para o MVP (A Jornada do Construtor)

Dividiremos nossa jornada em fases claras e sequenciais.

- **Fase 0: Gênese (Duração: 2 horas)**

- [] Criar um repositório Git (no GitHub, GitLab, etc.).
- [] Criar a estrutura inicial de pastas (/data, /notebooks, /scripts).
- [] Escrever a primeira versão do docker-compose.yml com os serviços

postgres e metabase.

- [] Criar o arquivo Makefile para os comandos de automação.
- **Fase 1: O Templo (Duração: 3 horas)**
 - [] Iniciar o serviço do PostgreSQL via Docker.
 - [] Conectar-se ao banco usando uma ferramenta como DBeaver ou psql e criar os schemas: raw_data, processed_data, analytics.
 - [] Validar a persistência dos dados reiniciando o contêiner.
- **Fase 2: O Ritual da Ingestão (Duração: 4 horas)**
 - [] Adicionar o serviço mage ao docker-compose.yml.
 - [] Criar um novo projeto Mage e desenvolver o pipeline de extração da API do INMET.
 - [] Testar o bloco de "data loader" para garantir que os dados brutos são salvos na tabela correta no schema raw_data.
- **Fase 3: A Alquimia (Duração: 4 horas)**
 - [] Desenvolver o script Python de transformação em um bloco "transformer" no Mage.
 - [] O script deve ler da raw_data, tratar valores nulos, converter tipos de dados e salvar o resultado em processed_data.
 - [] Adicionar testes de qualidade de dados básicos no script (ex: checar se a temperatura está em uma faixa plausível).
- **Fase 4: O Oráculo (Duração: 3 horas)**
 - [] Configurar a conexão do Metabase com o nosso PostgreSQL.
 - [] Criar as "perguntas" (queries) no Metabase apontando para a tabela processada.
 - [] Montar o dashboard final do MVP.

5. O "Algo a Mais" (Inspirando Confiança e Profissionalismo)

Para elevar nosso MVP de "funcional" para "impressionante", incluiremos:

- **Makefile de Automação:** Um Makefile na raiz do projeto com alvos simples e poderosos:
 - make up: Inicia toda a stack com docker-compose up.
 - make down: Para todos os serviços.

- make logs: Exibe os logs de todos os serviços.
- make clean: Remove volumes e redes para um reinício limpo.
- **Testes de Qualidade de Dados (Data Quality Checks):** Dentro do nosso script Python de transformação, implementaremos funções explícitas que validam os dados. Se um dado inválido for encontrado (ex: umidade acima de 100%), o pipeline deve falhar e registrar um erro claro. Isso demonstra uma mentalidade de engenharia robusta.
- **Variáveis de Ambiente:** Não haverá senhas ou chaves de API "hardcoded". Usaremos um arquivo .env (incluído no .gitignore) para gerenciar as credenciais, que serão lidas pelo docker-compose.yml.

6. Próximos Passos (Além do Horizonte do MVP)

Uma vez que o LÚMEN esteja estável, a jornada continua:

- **Integração do MLflow:** Adicionar o serviço do MLflow para começar a registrar experimentos de Machine Learning.
- **Pipeline de Treinamento:** Criar um novo pipeline no Mage para treinar um modelo simples (ex: prever a temperatura da próxima hora) e logá-lo no MLflow.
- **Integração com dbt:** Para transformações mais complexas e testes de dados mais avançados, o dbt pode ser integrado como a principal ferramenta de transformação.
- **Alerta:** Configurar alertas no Metabase ou via uma ferramenta externa quando certos limiares nos dados forem atingidos.