

Roteiro de Construção: Ecossistema LÚMEN (MVP)

Propósito: Este documento é o guia prático e sequencial para a construção da infraestrutura base do Projeto LÚMEN. Siga os passos na ordem apresentada para garantir a integridade e funcionalidade do sistema.

Pré-requisitos:

Docker e Docker Compose: Instalados e funcionando em sua máquina.

Git: Instalado para controle de versão.

Editor de Código: VS Code, Sublime Text, ou qualquer outro de sua preferência.

Um Cliente SQL (Opcional, mas recomendado): DBeaver, Postico ou pgAdmin para inspecionar o banco de dados.

Fase 0: Gênese - A Fundação do Repositório

Nesta fase, preparamos o terreno sagrado. Criamos a estrutura de pastas e os arquivos de configuração que darão forma ao nosso universo.

Passo 1: Criar a Estrutura de Diretórios

Abra seu terminal e execute os seguintes comandos para criar a estrutura do projeto:

Cria o diretório principal do projeto e entra nele

```
mkdir projeto_lumen
```

```
cd projeto_lumen
```

Cria os subdiretórios essenciais

```
mkdir scripts
```

```
mkdir mage_workspace
```

```
touch docker-compose.yml Makefile .env .gitignore README.md
```

Passo 2: Configurar o Controle de Versão (Git)

Ainda no terminal, na pasta projeto_lumen, inicialize o repositório:

```
git init
```

```
git branch -M main
```

Passo 3: Criar o Arquivo .gitignore

Abra o arquivo .gitignore e adicione o seguinte conteúdo. Ele evitará que arquivos sensíveis ou desnecessários sejam enviados ao seu repositório.

```
# Arquivos de ambiente
.env
```

```
# Arquivos de sistema
.DS_Store
Thumbs.db
```

```
# Diretórios de dependências Python
__pycache__/
*.pyc
venv/
.venv/
```

```
# Volumes do Docker (se mapeados localmente de forma diferente)
postgres_data/
```

Passo 4: Criar o Arquivo .env

Este arquivo guardará nossas "chaves secretas". Abra o .env e adicione:

```
# Credenciais do Banco de Dados PostgreSQL
POSTGRES_USER=padawan
POSTGRES_PASSWORD=aforcadoconhecimento
POSTGRES_DB=universo_dados
```

Fase 1: O Templo - Erguendo o PostgreSQL

Agora, vamos erguer o pilar central da nossa arquitetura: o banco de dados. Ele será o nosso templo, o guardião de todo o conhecimento.

Passo 5: Configurar o docker-compose.yml para o PostgreSQL

Abra o arquivo docker-compose.yml e adicione a definição do serviço do nosso banco de dados:

```
version: '3.8'
```

```
services:
```

```
  # O Templo do Saber
```

```
  postgres:
```

```
    image: postgres:15-alpine
```

```
    container_name: templo_postgres
```

```
environment:
  POSTGRES_USER: ${POSTGRES_USER}
  POSTGRES_PASSWORD: ${POSTGRES_PASSWORD}
  POSTGRES_DB: ${POSTGRES_DB}
volumes:
  - postgres_data:/var/lib/postgresql/data
ports:
  - "5432:5432"
restart: unless-stopped
```

```
volumes:
  postgres_data:
    name: lumen_postgres_data
```

Note que estamos usando as variáveis do arquivo .env para as credenciais, uma excelente prática de segurança.

Passo 6: Iniciar o Templo

No terminal, execute o comando para iniciar apenas o banco de dados em modo "detached" (em segundo plano):
`docker-compose up -d postgres`

Verifique se o contêiner está rodando com `docker ps`. Você deverá ver o `templo_postgres`.

Passo 7: Criar os Schemas Sagrados

Conecte-se ao banco de dados (usando seu cliente SQL ou o próprio Docker) e execute os seguintes comandos SQL para criar as alas do nosso templo:

```
CREATE SCHEMA IF NOT EXISTS raw_data;
CREATE SCHEMA IF NOT EXISTS processed_data;
CREATE SCHEMA IF NOT EXISTS analytics;
```

Fase 2: O Oráculo - Despertando o Metabase

Com o templo erguido, é hora de construir nosso observatório. O Metabase será nossa janela para a alma dos dados.

Passo 8: Adicionar o Metabase ao `docker-compose.yml`

Modifique seu docker-compose.yml para incluir o serviço do Metabase.

... (conteúdo anterior do postgres) ...

O Observatório Cósmico

metabase:

image: metabase/metabase:latest

container_name: observatorio_metabase

depends_on:

- postgres

ports:

- "3000:3000"

environment:

MB_DB_TYPE: postgres

MB_DB_DBNAME: \${POSTGRES_DB}

MB_DB_PORT: 5432

MB_DB_USER: \${POSTGRES_USER}

MB_DB_PASS: \${POSTGRES_PASSWORD}

MB_DB_HOST: postgres # Nome do serviço postgres no docker-compose

restart: unless-stopped

... (volumes) ...

Note que depends_on garante que o Metabase só iniciará depois do PostgreSQL. E o MB_DB_HOST é postgres porque os contêineres na mesma rede Docker podem se encontrar pelo nome do serviço.

Passo 9: Iniciar o Oráculo

No terminal, execute o comando up novamente. O Docker Compose é inteligente e só irá criar/atualizar o que for novo ou modificado.

docker-compose up -d

Aguarde alguns minutos para o Metabase inicializar completamente. Acesse <http://localhost:3000> no seu navegador para configurar sua conta de administrador e conectar o Metabase à sua fonte de dados (nosso PostgreSQL).

Fase 3: O Ritual da Ingestão - Conjurando o Mage

Agora que temos onde guardar e onde ver, precisamos de um meio para buscar o

conhecimento. O Mage será nosso coletor de sabedoria cósmica.

Passo 10: Adicionar o Mage ao docker-compose.yml

Adicione o serviço final do nosso MVP ao docker-compose.yml:

... (conteúdo anterior do postgres e metabase) ...

```
# Os Portais de Coleta e Orquestração
```

```
mage:
```

```
  image: mageai/mageai:latest
```

```
  container_name: portais_mage
```

```
  command: mage start projeto_lumen
```

```
  depends_on:
```

```
    - postgres
```

```
  ports:
```

```
    - "6789:6789"
```

```
  volumes:
```

```
    - ./mage_workspace:/home/src/
```

```
  restart: unless-stopped
```

O volumes aqui é crucial: ele mapeia a pasta mage_workspace do seu computador para dentro do contêiner, permitindo que você edite os pipelines localmente e eles sejam refletidos instantaneamente.

Passo 11: Iniciar a Constelação Completa

Execute o comando up mais uma vez para erguer toda a nossa infraestrutura.

```
docker-compose up -d
```

Acesse <http://localhost:6789> no seu navegador. Você verá a interface do Mage, pronta para que você crie seu primeiro pipeline de dados!

Parabéns, Arquiteto de Dados!

Você acaba de conjurar a infraestrutura completa do LÚMEN. Cada serviço está em sua órbita, comunicando-se e pronto para cumprir seu propósito.

Seu próximo passo, conforme nosso plano, é entrar na interface do Mage e iniciar a **Fase da Alquimia**: construir o pipeline que busca os dados, os transforma com a magia do Python e os deposita em nosso templo, para que o Metabase possa

finalmente nos contar suas histórias.

A fundação está pronta. A Força da Engenharia está com você.