

## Holocron de Verificação: A Jornada do Construtor LÚMEN

**Propósito:** Este checklist é sua bússola e seu esquadro. Use-o para navegar pelas fases de construção do projeto, garantindo que as melhores práticas de DevOps, Engenharia de Dados e Análise sejam aplicadas em cada etapa.

### Fase 0: Concepção & Arquitetura (O Sonho do Arquiteto)

*Antes de escrever uma única linha de código, a visão deve ser clara.*

- [ ] 📄 **Propósito Definido:** O objetivo principal do projeto está claro? Qual pergunta fundamental queremos responder com nossos dados?
- [ ] 🌌 **Serviços Mapeados:** Os "planetas" da nossa constelação (Postgres, Metabase, Mage, etc.) foram escolhidos e justificados?
- [ ] 🗺️ **Diagrama de Arquitetura Desenhado:** Existe um rascunho, ainda que simples, mostrando como os serviços se conectam e como os dados fluirão entre eles?
- [ ] 💧 **Fonte de Dados Identificada:** A fonte de dados para o MVP foi definida? (Ex: API do INMET). Suas regras e limitações são conhecidas?

### Fase 1: Gênese do Ambiente (A Conjuração do Terreno)

*Aqui, forjamos a fundação sobre a qual nosso ecossistema irá prosperar.*

- [ ] 📁 **Estrutura de Diretórios Criada:** A estrutura de pastas (/scripts, /mage\_workspace, etc.) foi criada conforme o Roteiro?
- [ ] 🗑️ **.gitignore Configurado:** O arquivo .gitignore existe e inclui, no mínimo, .env e diretórios de cache?
- [ ] 🔑 **.env Criado e Populado:** As variáveis de ambiente (credenciais do banco) estão definidas no arquivo .env?
- [ ] 📄 **docker-compose.yml Iniciado:** A primeira versão do docker-compose.yml, contendo apenas o postgres, foi escrita?
- [ ] 🤖 **Makefile para Automação (Opcional, mas recomendado):** O arquivo Makefile com comandos básicos (up, down, logs) foi criado para facilitar a vida?

### Fase 2: Construção & Validação dos Pilares (A Prova dos Titãs)

*Cada serviço é um pilar. Testamos a força de cada um isoladamente antes de conectá-los.*

- [ ] 🏗️ **[PostgreSQL] Contêiner em Execução:** O serviço postgres sobe sem erros com docker-compose up -d postgres?
- [ ] 🛡️ **[PostgreSQL] Persistência de Dados Validada:** Ao parar (docker-compose down) e iniciar o serviço novamente, os dados (como os schemas criados) ainda existem?
- [ ] 🔗 **[PostgreSQL] Conexão Externa Funcional:** Você consegue se conectar ao banco de dados usando um cliente SQL na porta 5432?
- [ ] 🏛️ **[PostgreSQL] Schemas Criados:** Os schemas raw\_data, processed\_data e analytics foram criados com sucesso?
- [ ] 🔭 **[Metabase] Contêiner em Execução:** O serviço metabase sobe sem erros após o postgres?
- [ ] 🤝 **[Metabase] Conexão com o Templo Estabelecida:** Dentro da interface do Metabase, a conexão com o PostgreSQL (usando o nome do serviço postgres como host) foi bem-sucedida?
- [ ] 🧙 **[Mage] Contêiner em Execução:** O serviço mage sobe sem erros?
- [ ] 🔄 **[Mage] Mapeamento de Volume Funcional:** Ao criar um arquivo dentro da pasta mage\_workspace no seu computador, ele aparece na interface do Mage?

### **Fase 3: O Fluxo da Força (O Ritual da Alquimia de Dados)**

*O coração do nosso projeto. Garantimos que a energia (dados) flui de forma limpa e correta.*

- [ ] 📦 **[Extração] Pipeline Conecta à Fonte:** O bloco "Data Loader" no Mage consegue buscar os dados da API (ou arquivo) sem erros de conexão ou autenticação?
- [ ] 📦 **[Carregamento] Dados Brutos Armazenados:** O pipeline salva os dados extraídos, sem tratamento, em uma tabela no schema raw\_data?
- [ ] 🦋 **[Transformação] Script Python Executa:** O bloco "Transformer" no Mage executa o script de limpeza sem erros de sintaxe ou de lógica?
- [ ] ✨ **[Transformação] Dados Processados Gerados:** O script de

transformação salva a tabela limpa e enriquecida no schema `processed_data`?

- [ ] 🩺 **[Qualidade] Testes de Sanidade Implementados:** O script de transformação possui asserts ou blocos try-except que validam a qualidade dos dados (ex: checar se colunas importantes não são nulas, se valores estão dentro de um range esperado)?

#### Fase 4: A Narrativa Visual (O Oráculo Revela)

*Os dados foram transmutados em sabedoria. Agora, tornamos essa sabedoria visível e compreensível.*

- [ ] ? **[Metabase] Perguntas Criadas:** As "perguntas" (queries) que alimentam os gráficos foram criadas e salvam corretamente?
- [ ] 📊 **[Metabase] Dashboard Montado:** O dashboard foi criado e os gráficos estão dispostos de forma lógica e esteticamente agradável?
- [ ] 📖 **[Metabase] A História é Clara:** O dashboard conta uma história? Um usuário que não participou do projeto consegue entender os principais insights em menos de 1 minuto?
- [ ] 🔄 **[Metabase] Dashboard Atualiza Automaticamente:** Os dados no dashboard refletem as novas execuções do pipeline do Mage? (Pode ser necessário configurar o cache no Metabase).

#### Fase 5: O Refinamento Jedi (O Toque do Mestre)

*Pequenos detalhes que separam um bom projeto de um projeto profissional e impressionante.*

- [ ] 📖 **README.md Completo:** O arquivo README.md explica o que o projeto faz, quais tecnologias usa e como executá-lo com um passo a passo simples?
- [ ] 🗃️ **Logs Inteligentes:** O docker-compose.yml está configurado para que os logs de cada contêiner sejam facilmente acessíveis? (Ex: docker-compose logs -f mage).
- [ ] 🧠 **Gerenciamento de Recursos (Avançado):** O docker-compose.yml possui limites de memória e CPU para os serviços, evitando que um deles consuma todos os recursos da máquina?
- [ ] 🧹 **Limpeza do Ambiente:** O Makefile (ou um script) possui um comando

para limpar completamente o ambiente, incluindo volumes (docker-compose down -v), para recomeçar do zero se necessário?