**Documentação do Processo ETL**

**Projeto: Centralização de Dados para Dashboard Financeiro**

**Data: 22/11/2024**

**1. Introdução**

Este documento descreve detalhadamente o processo de **Extração, Transformação e Carga (ETL)** implementado para consolidar dados financeiros provenientes de diversas fontes em um Data Warehouse centralizado.

**2. Visão Geral do Processo ETL**

|  |  |
| --- | --- |
| **Aspecto** | **Descrição** |
| **Objetivo** | Centralizar dados financeiros para análise no Power BI. |
| **Fontes de Dados** | ERP Financeiro, CRM, Sistema Legado |
| **Ferramentas Utilizadas** | Python (pandas), Power Query, PostgreSQL |
| **Frequência** | Processo diário |

**3. Arquitetura do ETL**

1. **Extração:** Coleta de dados das fontes.
2. **Transformação:** Limpeza, normalização e aplicação de regras de negócio.
3. **Carga:** Inserção dos dados processados no Data Warehouse.

**4. Detalhamento por Etapas**

**4.1 Extração**

|  |  |  |  |
| --- | --- | --- | --- |
| **Fonte** | **Formato** | **Ferramenta** | **Frequência** |
| ERP Financeiro | SQL Server Database | Python (pyodbc) | Diário |
| CRM de Vendas | API REST (JSON) | Python (requests) | Diário |
| Sistema Legado | Arquivos CSV | Python (pandas) | Diário |

**Exemplo de Script (Extração do ERP):**

python

Copiar código

import pyodbc conn = pyodbc.connect("DRIVER={SQL Server};SERVER=servidor;DATABASE=ERP;UID=user;PWD=password") query = "SELECT \* FROM transacoes WHERE data >= GETDATE()-1" df\_erp = pd.read\_sql(query, conn)

|  |  |
| --- | --- |
| **Tarefa** | **Descrição** |
| **Deduplicação de Registros** | Remoção de transações duplicadas. |
| **Normalização de Moedas** | Conversão de BRL para USD com taxa de câmbio fixa. |
| **Correção de Formatos** | Padronização de datas para o formato ISO 8601. |
| **Enriquecimento de Dados** | Adição de informações de clientes vindas do CRM. |

**4.2 Transformação**

**Exemplo de Script (Transformação de Dados):**

python

Copiar código

*# Normalização de moeda* df\_erp['valor\_usd'] = df\_erp['valor\_brl'] / 5.0 *# Padronização de datas* df\_erp['data\_transacao'] = pd.to\_datetime(df\_erp['data\_transacao'], format='%Y-%m-%d')

**4.3 Carga**

|  |  |  |  |
| --- | --- | --- | --- |
| **Destino** | **Formato** | **Ferramenta** | **Frequência** |
| Data Warehouse | PostgreSQL Database | Python (psycopg2) | Diário |

**Exemplo de Script (Carga no Data Warehouse):**

python

Copiar código

import psycopg2 conn = psycopg2.connect("dbname=datawarehouse user=user password=password host=localhost") cursor = conn.cursor() for index, row in df\_erp.iterrows(): cursor.execute("INSERT INTO transacoes (id, valor\_usd, data\_transacao) VALUES (%s, %s, %s)", (row['id'], row['valor\_usd'], row['data\_transacao'])) conn.commit()

**5. Fluxo do Processo ETL**

**Fluxo Representado:**

1. Fontes de Dados (ERP, CRM, Legado) →
2. Extração (via Python) →
3. Transformação (Deduplicação, Normalização, Enriquecimento) →
4. Carga no Data Warehouse (PostgreSQL) →
5. Visualização (Power BI).

**6. Monitoramento e Logs**

1. **Monitoramento:**
   * Logs de erros gravados diariamente no arquivo etl\_logs.txt.
   * Alerta de falhas por e-mail em caso de erro crítico.
2. **Exemplo de Log:**

csharp

Copiar código

[22/11/2024 02:00] Extração iniciada. [22/11/2024 02:10] Extração concluída. 5.200 registros processados. [22/11/2024 02:15] Transformação concluída. [22/11/2024 02:30] Carga concluída. 5.200 registros inseridos no Data Warehouse.

**7. Riscos e Contingências**

|  |  |  |
| --- | --- | --- |
| **Risco** | **Impacto** | **Plano de Contingência** |
| Falha na conexão com o ERP | Alto | Backup de última extração; reprocessamento manual. |
| Dados inconsistentes | Médio | Implementação de validações automáticas. |

**8. Anexos**

1. **Script completo do ETL:** Disponível em anexo no repositório GitHub.
2. **Plano de Execução:** Cronograma e detalhes das rotinas agendadas.
3. **Manuais de Operação:** Passo a passo para ajustes no processo.

**Assinatura do Responsável pelo ETL:**

Elmesson Ribeiro