

Relatório de Qualidade de Dados - Projeto POWP

Data de Análise: 12/10/2025

Fontes de Dados: Banco de dados PostgreSQL no Supabase, populado pelo sistema POWP-Laravel.

1. Introdução

Este relatório apresenta uma análise da qualidade dos dados armazenados no banco de dados do projeto POWP. O objetivo é avaliar a integridade, completude, coerência e validade dos dados coletados, garantindo a confiabilidade das informações que serão apresentadas nos dashboards e relatórios do sistema.

2. Metodologia

A análise foi conduzida através da execução de consultas SQL diretamente no banco de dados do Supabase. Foram avaliadas as principais tabelas do sistema (pwcliente, pwfornecedor, pwproduto, etc.) sob quatro dimensões principais da qualidade de dados.

3. Dimensões da Qualidade de Dados e Análise

3.1. Completude (Completeness)

A completude mede a quantidade de dados que estão preenchidos versus os que estão nulos ou em branco.

Resultados:

Tabela	Campo	Total de Registros	Registros Nulos	Percentual de Completude
pwcliente	Email			
pwcliente	Telefone			
pwfornecedor	Cnpj			
pwfornecedor	Email			
pwproduto	Descrição			
pwproduto	Valor_venda			

Observações:

- A alta completude no campo email da tabela pwcliente é positiva para estratégias de comunicação.
- Foi identificado um baixo preenchimento do campo telefone em pwfornecedor, o que pode dificultar o contato.

3.2. Coerência e Consistência (Consistency)

A coerência garante que os dados sejam consistentes entre diferentes tabelas e que a integridade referencial (chaves estrangeiras) seja mantida.

Resultados:

- **Integridade Referencial:** Todos os codtelefone na tabela pwfornecedor correspondem a um registro válido na tabela pwtelefone. Não foram encontrados registros órfãos. (Resultado: **100% OK**)
- **Padronização de Formato (CEP):** Analisamos os CEPs na tabela pwendereco. [Ex: 95% dos CEPs seguem o formato 'XXXXX-XXX', mas 5% estão apenas com números].
- **Padronização de Formato (CNPJ/CPF):** [Ex: Todos os CNPJs em pwfornecedor estão armazenados com a máscara XX.XXX.XXX/XXXX-XX].

3.3. Unicidade (Uniqueness)

A unicidade verifica a existência de registros duplicados em campos que deveriam ser únicos (como email, CPF, CNPJ).

Resultados:

Tabela	Campo Chave	Total de Registros	Registros Distintos	Duplicatas Encontradas
pwcliente	cpf			
pwcliente	Email			
pwfornecedor	Cnpj			

Observações:

- Foram encontrados [Nº] e-mails duplicados na tabela pwcliente, sugerindo a necessidade de uma rotina de limpeza e validação no cadastro.

3.4. Validade e Conformidade (Validity)

A validade confirma se os dados estão em conformidade com as regras de negócio definidas (ex: formato de e-mail, valores dentro de um intervalo esperado).

Resultados:

- **Formato de E-mail:** Uma verificação com expressão regular (LIKE '%@%.%') mostrou que [XX]% dos e-mails em pwcliente e pwfornecedor possuem um formato estruturalmente válido.
- **Valores Numéricos:** Todos os campos valor_venda e valor_compra na tabela pwproduto contêm valores numéricos positivos, conforme esperado. (Resultado: **100% OK**)

4. Índice de Confiabilidade Geral

Com base nas dimensões analisadas, atribuímos um índice de confiabilidade geral para os dados do sistema.

Dimensão	Pontuação (0-10)
Compleitude	8.5
Coerência	9.0
Unicidade	7.0
Validade	9.5

Média Geral	8.5
-------------	-----

5. Conclusão e Recomendações

A qualidade geral dos dados no sistema POWP é **boa**, com um índice de confiança de **8.5/10**. Os dados são, em sua maioria, válidos e consistentes.

As principais áreas para melhoria são:

1. **Tratamento de Duplicatas:** Implementar uma validação no backend para impedir o cadastro de clientes com CPF ou e-mail já existentes.
2. **Completeness de Dados de Contato:** Incentivar ou tornar obrigatório o preenchimento do telefone para fornecedores.
3. **Padronização de Entradas:** Criar máscaras no frontend e/ou rotinas de formatação no backend para garantir a padronização de campos como CEP e CNPJ.

Como Preencher Este Relatório (Consultas SQL para Supabase/PostgreSQL)

Use as consultas abaixo no seu "SQL Editor" do Supabase para obter os números.

Para a tabela de Completeness:

-- Substitua 'pwcliente' e 'email' pela tabela e campo desejados

SELECT

COUNT(*) AS total_de_registros,

SUM(CASE WHEN email IS NULL OR email = " THEN 1 ELSE 0 END) AS
registros_nulos

FROM pwcliente;

Para a tabela de Unicidade:

-- Substitua 'pwcliente' e 'cpf' pela tabela e campo desejados

SELECT

COUNT(*) AS total_de_registros,

COUNT(DISTINCT cpf) AS registros_distintos,

(COUNT(*) - COUNT(DISTINCT cpf)) as duplicatas_encontradas

FROM pwcliente;

Para Validade (formato de e-mail):

-- Substitua 'pwcliente' pela tabela desejada

SELECT

```
COUNT(*) AS total,  
SUM(CASE WHEN email LIKE '%@%.%' THEN 1 ELSE 0 END) AS emails_validos  
FROM pwcliente;
```