
PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO

# Identificação de datasets relacionados, em contexto com metadados ausentes ou parciais

**Sérgio Bernardelli Netto**

PROJETO FINAL DE GRADUAÇÃO

CENTRO TÉCNICO CIENTÍFICO - CTC
DEPARTAMENTO DE INFORMÁTICA
Curso de Graduação em Ciência da Computação

Rio de Janeiro, Novembro de 2025


**Sérgio Bernardelli Netto**

# Identificação de datasets relacionados, em contexto com metadados ausentes ou parciais

Relatório de Projeto Final, apresentado ao programa Ciência da Computação da PUC-Rio como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Marcos Vianna Villas

Rio de Janeiro, Novembro de 2025

**Agradecimentos**

Agradeço primeiramente ao meu orientador, Professor Marcos Vianna Villas, pela orientação indispensável, paciência e pelas discussões técnicas que guiaram este trabalho desde a sua concepção até o protótipo final. Aos especialistas da área de dados que participaram das entrevistas, cuja "visão de mercado"foi crucial para direcionar o foco do projeto da teoria para uma aplicação prática e relevante. Aos meus pais, Paulo Marcos Gotardelo Fraga Moreira e Silvia Carla Bernardelli Fraga Moreira, pelo suporte incondicional que me forneceram e a quem dedico todo meu esforço. Aos amigos que conheci ao longo da faculdade, especialmente Lucas Caputo Bello, Isabella Carmo e Paulo Vitor Libório, que de colegas se tornaram inspirações.

A minha namorada, Maria Beatriz de Oliveira Rabello, pelo amor, compaixão e cuidado durante esta etapa complexa e ocupada. “Ich liebe dich”, amor.

Por fim, agradeço a banca de professores que pôde participar desse processo, pelo tempo dedicado à análise deste trabalho, e pelas valiosas contribuições que contribuirão para a pesquisa.

## Resumo

**Bernardelli Netto, Sérgio. Vianna Villas, Marcos.** Identificação de datasets relacionados, em contexto com metadados ausentes ou parciais. Rio de Janeiro, 2025. 41 p. Relatório de Projeto Final II – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

O projeto **propõe** uma ferramenta para a análise, identificação e determinação de possíveis combinações de datasets, seja pela identificação de chaves primárias e estrangeiras entre dois datasets (o que permite a junção relacional entre datasets), seja pela similaridade entre datasets (o que permite união, interseção e diferença entre datasets). Serão utilizadas técnicas de mineração de dados e aprendizado de máquina, entre outras. Os resultados obtidos pelo projeto poderão viabilizar novas análises de dados com a utilização de datasets que anteriormente não se apresentavam como relacionados, em um determinado contexto.

**Palavras-chave:** Identificação de datasets, metadados ausentes, metadados parciais, mineração de dados, aprendizado de máquina, dependências funcionais relaxadas, data lakes.

## Abstract

**Bernardelli Netto, Sérgio. Vianna Villas, Marcos.** Identification of Related Datasets in the Context of Missing or Partial Metadata. Rio de Janeiro, 2025. 41 p. Relatório de Projeto Final II – Departamento de Informática. Pontifícia Universidade Católica do Rio de Janeiro.

This project proposes a tool for analyzing, identifying, and determining potential combinations of datasets, focusing on the discovery of primary and foreign keys between two datasets (enabling relational joins), or similarity between datasets (enabling union, intersection, and difference operations). The approach leverages data mining and machine learning techniques to automate the correlation process between tables. By doing so, the project aims to enable new forms of data analysis that were previously unattainable due to the lack of explicit relationships between datasets. The results are expected to enhance data integration and uncover insights in contexts where datasets appeared unrelated.

**Keywords:** dataset identification, missing metadata, partial metadata, data mining, machine learning, relaxed functional dependencies, data lakes.

**Sumário**

**Lista de Figuras**

# 1 Introdução

Com o aumento significativo da quantidade de dados disponíveis em diversos setores, a gestão eficiente e a correta interpretação desses dados se tornaram desafios cruciais. Em muitos cenários, os dados estão distribuídos em diferentes conjuntos de datasets, sejam eles tabelas pertencentes a um esquema de banco de dados tradicional ou arquivos armazenados em data lakes[1].

As tabelas de um esquema de banco de dados geralmente contam com chaves estrangeiras (FKs) previamente definidas, o que facilita a identificação da relação entre elas. No entanto, em ambientes de data lakes, onde os dados são frequentemente ingeridos sem metadados adequados, a situação é mais complexa. Quando esses data lakes carecem de organização e descrição de seus dados, transformam-se em verdadeiros data swamps, dificultando a identificação de ligações entre conjuntos de dados. A falta de metadados estruturados, seja em tabelas de bancos de dados com FKs ou em data lakes, representa um obstáculo crítico para a identificação de ligações entre datasets. Sem esses metadados, relacionar as informações se torna uma tarefa manual e propensa a erros, especialmente quando se lida com grandes volumes de dados distribuídos em diferentes fontes e formatos.

Portanto, a falta de metadados é um problema real que impede o aproveitamento total das potencialidades dos dados em um data lake, limitando a capacidade de realizar análises aprofundadas. A dificuldade em automatizar a descoberta de relações (sejam chaves estrangeiras ou similaridades de conteúdo) entre datasets afeta diretamente a qualidade das análises e a eficiência das operações de integração de dados.

[1]Data lakes representam uma abordagem alternativa aos armazéns de dados tradicionais, permitindo o armazenamento de dados em sua forma original e sem a necessidade de esquemas rígidos. No entanto, essa flexibilidade também traz desafios significativos, como a possibilidade de o repositório se tornar um data swamp, com dados de difícil organização e acesso sem metadados robustos (Khine & Wang, 2018).

## 2 Situação Atual

A situação atual da automação da identificação de ligações entre conjuntos de dados envolve o uso de técnicas sofisticadas para otimizar a integração de dados, especialmente onde os metadados são limitados. Por exemplo, em sistemas legados, onde tabelas frequentemente não possuem chaves estrangeiras (FKs) definidas, ferramentas como o Metanome (Papenbrock et al., 2015) aplicam algoritmos avançados de perfilamento para detectar dependências complexas, como dependências de inclusão e funcionais.

Nos data lakes, onde diversos datasets são colecionados sem uma descrição adequada, o problema se agrava. Para mitigar isso, o uso de dependências funcionais relaxadas (RFDs) estende as dependências clássicas para capturar ligações em dados não padronizados ou com inconsistências. Um exemplo clássico de RFD seria a relação entre CEP e Cidade (CEP $\rightarrow$ Cidade). Em um banco de dados perfeito, um CEP sempre determina uma única cidade. No entanto, em dados reais de um data lake, pode haver erros de digitação ou formatos diferentes (ex: "Rio de Janeiro"e "Rio de Janeiro/RJ") para o mesmo CEP. Uma RFD permitiria identificar essa dependência funcional mesmo com um certo percentual de violações ou erros nos dados (Caruccio et al., 2016).

No contexto de dados abertos, algoritmos de busca de similaridade, como o Locality Sensitive Hashing (LSH), oferecem soluções escaláveis para encontrar tabelas que podem ser combinadas (relacionáveis) em bases amplas, como as governamentais (Bhardwaj et al., 2020). A literatura existente, embora robusta, frequentemente pressupõe um nível mínimo de metadados estruturais, como a presença de cabeçalhos (headers). Ferramentas como o Metanome, por exemplo, são altamente eficazes na descoberta de dependências, mas operam com maior eficácia quando o esquema dos dados, ou pelo menos os nomes dos atributos, são conhecidos.

## 3 Objetivos

O principal objetivo deste trabalho é propor uma ferramenta que automatize a análise, identificação e determinação de possíveis combinações entre diferentes datasets, mesmo em contextos onde os metadados estejam ausentes ou sejam parciais. Especificamente, o projeto visa desenvolver soluções que permitam a identificação de um **par chave primária/chave estrangeira** entre dois datasets, o que permitirá realizar a junção relacional (join) entre elas, bem como a detecção de similaridades que possam indicar possíveis ligações entre tabelas ou arquivos de dados, o que permitirá a realização de operações de união, interseção e diferença entre eles.

É importante ressaltar que os arquivos CSV tratados neste projeto estão em formato tabular e, embora não sejam estritamente equivalentes a tabelas de um modelo relacional formal (com restrições de integridade garantidas pelo SGBD), foram utilizados os conceitos de chaves candidatas, chaves primárias e chaves estrangeiras para guiar a lógica de inferência. Conforme definido por Elmasri & Navathe [6], uma chave candidata é um atributo (ou conjunto de atributos) que identifica unicamente uma tupla em uma relação, e uma chave estrangeira é um atributo que estabelece uma relação lógica com a chave primária de outra tabela. A ferramenta desenvolvida busca identificar essas estruturas lógicas dentro dos dados brutos dos arquivos.

Espera-se que a ferramenta obtida viabilize análises de dados mais abrangentes, permitindo explorar ligações ocultas ou não evidentes entre datasets que, anteriormente, não se apresentavam como conectados, promovendo assim uma análise mais abrangente e integrada.

Este trabalho posiciona-se de forma complementar a essas soluções, focando em um nicho específico e desafiador: o data lake em seu estado mais bruto, onde até mesmo os cabeçalhos das colunas podem estar ausentes. A ferramenta desenvolvida (detalhada na Seção 4), especialmente a partir do Protótipo 4, foi projetada para ser resiliente a essa degradação de metadados. Ao validar sistematicamente a capacidade do sistema de inferir relações em arquivos sem cabeçalho (usando dados sintéticos) e, posteriormente, generalizar essa lógica para dados reais e desconhecidos do `gov.br` (Protótipo 5), este projeto oferece uma contribuição prática. A solução final atua como uma ferramenta de perfilagem e descoberta ágil na "primeira milha"da ingestão de dados, preparando o terreno para que ferramentas de análise mais profundas, como as citadas, possam ser aplicadas de forma eficaz em um momento posterior.

# 4 Desenvolvimento de Protótipos

Para atender aos objetivos do projeto, optou-se por uma metodologia de desenvolvimento iterativa, estruturada em cinco protótipos. Esta abordagem foi escolhida para permitir a validação e o refinamento progressivo da complexidade algorítmica, com cada protótipo construindo sobre as lições aprendidas na iteração anterior. O escopo de cada protótipo foi definido de forma incremental:

- **Protótipo 1 (P1):** visava validar a lógica de perfilamento de arquivo único.
- **Protótipo 2 (P2):** tinha como escopo introduzir o "segundo algoritmo"de inferência relacional.
- **Protótipo 3 (P3):** focou na "visão de mercado", implementando uma API para integração.
- **Protótipo 4 (P4):** foi desenhado para testar a robustez do "quarto algoritmo"contra dados degradados (ex: sem cabeçalho).
- **Protótipo 5 (P5):** teve como escopo validar a generalização do sistema final em um domínio de dados real e desconhecido.

Vale notar que os protótipos desenvolvidos tratam especificamente de arquivos no formato CSV. Embora o foco atual seja este formato devido à sua prevalência em dados abertos, trabalhos futuros poderão expandir a análise para outros formatos estruturados e semi-estruturados, como JSON e XML.

## 4.1 Arquitetura Inicial dos Protótipos

A solução foi especificada para ser desenvolvida integralmente na linguagem Python (versão 3.11), escolhida por sua robustez e pelo vasto ecossistema de bibliotecas para ciência de dados. A arquitetura de software foi planejada para ser modular, composta pelos seguintes componentes principais:

- **DataLoader:** Responsável pela ingestão dos dados (Utilizado em todos os protótipos; aprimorado no P4).
- **DataProfiler:** Responsável pelo perfilamento estatístico das colunas (Introduzido no P1).
- **FKFinder:** Responsável pela lógica de inferência de chaves estrangeiras (Introduzido no P2).

- **AttributeMatcher:** Responsável pela comparação de similaridade de nomes de colunas (Introduzido no P2).

Para a Ingestão de Dados, a especificação definiu o uso da biblioteca glob para a descoberta de arquivos de forma dinâmica e da biblioteca Pandas para a leitura de arquivos CSV (`pandas.read_csv`), devido à sua eficiência e capacidade de lidar com diversos formatos e codificações.

Para o Perfilamento, foi especificado o uso de operações nativas da biblioteca Pandas para o cálculo de metadados estatísticos, como `.dtype` para tipos de dados, `.nunique()` para contagem de valores únicos, e `.isnull().sum()` para valores nulos. Esta seria a base do componente `DataProfiler`.

Para o Matching de Atributos, a especificação previu o uso da biblioteca padrão `difflib`, especificamente a classe `SequenceMatcher`, para realizar a análise de similaridade textual entre nomes de colunas. Esta abordagem foi escolhida por não requerer dependências externas e ser eficaz para a detecção de similaridade de strings.

Para a Inferência e Validação de Relações, o núcleo algorítmico do FKFinder, a especificação determinou o uso de estruturas de dados `set` (conjuntos) nativas do Python. A verificação de inclusão de dados, a operação mais custosa do processo, seria implementada através do método `issubset()` (ou uma verificação percentual sobre ele), garantindo uma performance otimizada em comparação a iterações aninhadas.

Para a Integração de Sistema (escopo do P3), o micro-framework Flask foi especificado para a construção da API. Sua leveza e facilidade em expor a lógica do sistema como endpoints HTTP que manipulam JSON foram os fatores decisivos. Finalmente, para a Geração de Dados de Teste (escopo do P4), a especificação definiu a criação de um script (`gerar_dados_artificiais.py`) que utilizaria as bibliotecas Pandas e NumPy para criar datasets sintéticos com características controladas, permitindo testes de robustez em cenários de dados degradados. A biblioteca `logging` padrão do Python foi definida como a ferramenta para registro de eventos e depuração em todas as fases.

# 5 Prototipação

Esta seção detalha a execução da metodologia de desenvolvimento, documentando a especificação, construção, testes e ajustes de cada um dos cinco protótipos. (O código fonte e os datasets utilizados estão disponíveis no repositório GitHub do projeto: https://github.com/SBNetto01/tcc-dataset-relations).

## 5.1 Protótipo 1: Prova de Conceito e Perfilamento de Arquivo Único

O objetivo do Protótipo 1 foi validar a lógica de perfilamento de arquivos isolados. O sistema foi projetado para analisar arquivos .csv individualmente e identificar Chaves Candidatas (PKs candidatas).

A arquitetura consistia em um script de linha de comando (`pythonsrc/main.py`) que utilizava três componentes principais: o `FileHandler` (para encontrar os arquivos), o `DataLoader` (para carregá-los) e o `DataProfiler` (para analisá-los). A construção foi realizada em Python 3.11, usando a biblioteca Pandas como principal dependência para a ingestão e análise estatística (cálculo de métricas de unicidade e nulidade).

Os testes do P1 foram executados com um conjunto de dados real da NBA, obtido da plataforma Kaggle. Os arquivos principais foram o `Advanced.csv` (26.280 linhas) e o `Player_Play_By_Play.csv` (61.168 linhas).

### Datasets Utilizados

- `dataset_prototipo1_1.csv` (`Advanced.csv`): colunas `seas_id`, `season`, `player_id`, `player`, `birth_year`, `pos`, `age`, `experience`, `lg`, `tm`, `g`, `mp`, `pg_percent`, `sg_percent`, `sf_percent`, `pf_percent`, `c_percent`.
- `dataset_prototipo1_2.csv` (`Player_Play_By_Play.csv`): colunas `seas_id`, `season`, `player_id`, `player`, `birth_year`, `pos`, `age`, `experience`, `lg`, `tm`, `g`, `mp`, `per`, `ts_percent`, `x3p_ar`, `f_tr`, `orb_percent`, ... (etc.)

| | A | B | C | D | E |
|---|---|---|---|---|---|
| 1 | seas_id | season | player_id | player | birth_year |
| 2 | 31871 | 2025 | 5025 | A.J. Green | NA |
| 3 | 31872 | 2025 | 5026 | A.J. Lawson | NA |
| 4 | 31873 | 2025 | 5210 | AJ Johnson | NA |
| 5 | 31874 | 2025 | 5210 | AJ Johnson | NA |
| 6 | 31875 | 2025 | 5210 | AJ Johnson | NA |
| 7 | 31876 | 2025 | 4219 | Aaron Gordon | NA |
| 8 | 31877 | 2025 | 4582 | Aaron Holiday | NA |
| 9 | 31878 | 2025 | 4805 | Aaron Nesmith | NA |
| 10 | 31879 | 2025 | 4900 | Aaron Wiggins | NA |
| 11 | 31880 | 2025 | 5109 | Adam Flagler | NA |
| 12 | 31881 | 2025 | 5110 | Adama Sanogo | NA |
| 13 | 31882 | 2025 | 5211 | Adem Bona | NA |
| 14 | 31883 | 2025 | 5212 | Ajay Mitchell | NA |
| 15 | 31884 | 2025 | 3734 | Al Horford | NA |

Figura 1: Amostra das primeiras 15 linhas do dataset `Advanced.csv`.

| 1 | seas_id | season | player_id | player | birth_year |
|---|---|---|---|---|---|
| 2 | 31871 | 2025 | 5025 | A.J. Green | NA |
| 3 | 31872 | 2025 | 5026 | A.J. Lawson | NA |
| 4 | 31873 | 2025 | 5210 | AJ Johnson | NA |
| 5 | 31874 | 2025 | 5210 | AJ Johnson | NA |
| 6 | 31875 | 2025 | 5210 | AJ Johnson | NA |
| 7 | 31876 | 2025 | 4219 | Aaron Gordon | NA |
| 8 | 31877 | 2025 | 4582 | Aaron Holiday | NA |
| 9 | 31878 | 2025 | 4805 | Aaron Nesmith | NA |
| 10 | 31879 | 2025 | 4900 | Aaron Wiggins | NA |
| 11 | 31880 | 2025 | 5109 | Adam Flagler | NA |
| 12 | 31881 | 2025 | 5110 | Adama Sanogo | NA |
| 13 | 31882 | 2025 | 5211 | Adem Bona | NA |
| 14 | 31883 | 2025 | 5212 | Ajay Mitchell | NA |
| 15 | 31884 | 2025 | 3734 | Al Horford | NA |

Figura 2: Amostra das primeiras 15 linhas do segundo dataset da NBA.

O teste consistiu em executar o `main.py` e validar o relatório de saída. Os resultados foram bem-sucedidos: o sistema carregou ambos os arquivos e gerou um relatório textual (`results/attribute_suggestions.txt`) que identificava corretamente as Chaves Candidatas (como `seas_id` em `Advanced.csv`) em cada arquivo de forma isolada, cumprindo o objetivo.

**Relatório de Saída (Exemplo):**

```
[INFO] Iniciando perfilamento de Advanced.csv...
[RESULT] Coluna: seas_id | Tipo: int64 | Únicos: 73 | Nulos:
    0 (0.00%)
[RESULT] Coluna: player_id | Tipo: int64 | Únicos: 4923 |
   Nulos: 0 (0.00%)
[RESULT] Coluna: age | Tipo: int64 | Únicos: 28 | Nulos: 0
   (0.00%)
[INFO] Candidatas a PK para Advanced.csv: ['seas_id', '
   player_id']
...
[INFO] Processamento finalizado.
```

Os Ajustes necessários identificados no P1 foram arquiteturais. A avaliação confirmou a principal limitação: a ausência de análise relacional. O sistema era "míope", incapaz de comparar os perfis entre si. O ajuste proposto para o Protótipo 2 foi, portanto, a criação de um novo componente, o `FKFinder`, e a refatoração completa do `main.py` para um fluxo em duas fases (profiling global seguido de análise relacional).

## 5.2 Protótipo 2: Inferência Relacional Básica

O objetivo do Protótipo 2 foi expandir a lógica para incluir o "segundo algoritmo", focado na inferência de Chaves Estrangeiras (FKs). O sistema agora deveria comparar cada coluna de cada tabela com um catálogo global de PKs candidatas.

A arquitetura reutilizou a base do P1 (Python e Pandas) e introduziu o componente `FKFinder`. Este componente utilizou a biblioteca `difflib` (especificamente a classe `SequenceMatcher`, abstraída no `AttributeMatcher`) para a comparação de nomes e estruturas de dados `set` nativas do Python para a validação de inclusão de dados (verificação de conteúdo). O `main.py` foi reescrito para orquestrar esse novo fluxo de duas fases: primeiro, perfilar todos os arquivos, e segundo, invocar o `FKFinder` para a análise relacional.

Os testes desta fase utilizaram os mesmos datasets da NBA (em versões encurtadas, devido ao custo computacional da análise de inclusão), conforme P1.

Os resultados, gerados no artefato `output/fk_identification_results.txt`, confirmaram que o `FKFinder` conseguia inferir corretamente a relação `seas_id` nos dados da NBA.

**Relatório de Saída (Exemplo):**

```
RELATÓRIO DE IDENTIFICAÇÃO DE CHAVES ESTRANGEIRAS
================================================

[RELAÇÃO ENCONTRADA]
Tabela Origem (FK): Player Play By Play.csv
Coluna Origem: seas_id
--> REF -->
Tabela Destino (PK): Advanced.csv
Coluna Destino: seas_id

Métricas de Validação:
- Similaridade de Nome: 1.0 (100%)
- Inclusão de Dados: 1.0 (100% dos valores contidos)
- Tipo de Dado: Compatível (int64 -> int64)
------------------------------------------------
```

Os Ajustes identificados foram de duas naturezas: o desempenho da análise de inclusão foi confirmado como um gargalo em datasets maiores; e o sistema, como um script local, foi considerado de baixa usabilidade. O ajuste proposto para o Protótipo 3 foi focar na "visão de mercado", refatorando o sistema como um serviço de API e testando-o com um novo conjunto de dados.

## 5.3 Protótipo 3: Integração via API e Visão de Mercado

O objetivo do Protótipo 3 foi transformar a ferramenta de um script local em um serviço consumível, motivado pela "visão de mercado". Para validar essa necessidade, foram realizadas entrevistas qualitativas com especialistas da área.

**Entrevistas com Especialistas:** No contexto do desenvolvimento do Protótipo 3, foi conduzida uma entrevista semiestruturada com Ricardo, um profissional com 34 anos de experiência no mercado de tecnologia da informação. Atuando como proprietário de duas empresas (focadas em infraestrutura e desenvolvimento de software para saúde/educação), Ricardo possui ampla vivência prática com os desafios de gestão de dados.

Durante a entrevista, o especialista validou a relevância crítica do problema abordado pelo projeto. Ele confirmou que a identificação e manutenção de chaves primárias e estrangeiras são essenciais para evitar a redundância de dados e garantir a integridade referencial, sendo uma tarefa recorrente em análises de sistemas. Ricardo destacou que, embora bancos de dados modernos ofereçam recursos nativos, uma ferramenta de "pré-análise"seria extremamente valiosa para orientar DBAs e desenvolvedores, especialmente ao lidar com

sistemas legados ou desestruturados.

Um dos pontos altos da discussão foi sobre performance: o entrevistado citou um caso prático onde a correta indexação e definição de chaves em um banco de dados reduziu o tempo de uma consulta complexa de 50 segundos para apenas 5 segundos. Ele validou a abordagem técnica proposta (uso de unicidade e inclusão) e ressaltou que o maior valor da ferramenta estaria na economia de tempo de análise e na garantia de consistência dos dados, sugerindo que o produto deve focar em mostrar claramente os ganhos de integridade para o usuário final.

A arquitetura foi modificada para operar como uma API RESTful. A principal adição foi a biblioteca FastAPI, escolhida por sua alta performance e validação automática de dados. Um novo arquivo `api.py` foi criado para "envolver"a lógica principal em um endpoint HTTP. O "terceiro algoritmo"consistiu em implementar um endpoint capaz de receber e processar múltiplos arquivos CSV simultaneamente via upload (`multipart/form-data`), centralizando a lógica de inferência que antes rodava apenas localmente.

**Especificação da API:**

- **Endpoint:** `POST /analyze/`
- **Descrição:** Recebe múltiplos arquivos CSV, realiza o carregamento e perfilamento em memória, e retorna uma lista de possíveis relações de chave estrangeira identificadas entre eles.
- **Parâmetros:** `files` (Lista de arquivos .csv). É obrigatório o envio de pelo menos dois arquivos.

**Exemplo de Requisição (cURL):**

```
curl -X 'POST' \
  'http://localhost:8000/analyze/' \
  -H 'accept: application/json' \
  -H 'Content-Type: multipart/form-data' \
  -F 'files=@tabela_clientes.csv;type=text/csv' \
  -F 'files=@tabela_pedidos.csv;type=text/csv'
```

**Exemplo de Resposta (JSON):**


```
{
  "relations_found": [
    {
      "FK_DataFrame_Name": "tabela_pedidos",
      "FK_Column": "id_cliente",
      "FK_DataType": "int64",
      "PK_DataFrame_Name": "tabela_clientes",
      "PK_Column": "id_cliente",
      "PK_DataType": "int64",
      "Inclusion_Percentage": 100.0,
      "Name_Similarity_Score": 100.0
    }
  ]
}
```


Os testes desta fase focaram na validação da nova API usando um conjunto de dados acadêmico com três tabelas. Testes funcionais enviando requisições POST ao endpoint `/analyze` validaram que o sistema conseguia inferir as relações neste novo dataset.

O resultado foi uma ferramenta funcional e integrável. A Lógica principal da API é ilustrada na Figura 6.

### Datasets Utilizados

- `dataset_prototipo3_1.csv`: colunas `nome`, `cpf`, `matricula`.
- `dataset_prototipo3_2.csv`: colunas `rg`, `cpf`.
- `dataset_prototipo3_3.csv`: colunas `matricula`, `rg`.

| 1 | nome | cpf | matricula |
|---|---|---|---|
| 2 | João Silva | 111.222.333-44 | 2025001 |
| 3 | Maria Oliveira | 222.333.444-55 | 2025002 |
| 4 | Pedro Costa | 333.444.555-66 | 2025003 |
| 5 | Ana Souza | 444.555.666-77 | 2025004 |
| 6 | Carlos Pereira | 555.666.777-88 | 2025005 |
| 7 | Mariana Ferreira | 666.777.888-99 | 2025006 |
| 8 | Lucas Rodrigues | 777.888.999-00 | 2025007 |
| 9 | Juliana Almeida | 888.999.000-11 | 2025008 |
| 10 | Rafael Lima | 999.000.111-22 | 2025009 |
| 11 | Ana Souza | 000.111.222-33 | 2025010 |

Figura 3: Amostra do dataset acadêmico (Tabela 1: Alunos).

| 1 | rg | cpf |
|---|---|---|
| 2 | 12.345.678-9 | 111.222.333-44 |
| 3 | 23.456.789-0 | 222.333.444-55 |
| 4 | 34.567.890-1 | 333.444.555-66 |
| 5 | 45.678.901-2 | 444.555.666-77 |
| 6 | 56.789.012-3 | 555.666.777-88 |
| 7 | 67.890.123-4 | 666.777.888-99 |
| 8 | 78.901.234-5 | 777.888.999-00 |
| 9 | 89.012.345-6 | 888.999.000-11 |
| 10 | 90.123.456-7 | 999.000.111-22 |
| 11 | 01.234.567-8 | 000.111.222-33 |

Figura 4: Amostra do dataset acadêmico (Tabela 2: Documentos).

| 1 | matricula | rg |
|---|---|---|
| 2 | 2025001 | 12.345.678-9 |
| 3 | 2025002 | 23.456.789-0 |
| 4 | 2025003 | 34.567.890-1 |
| 5 | 2025004 | 45.678.901-2 |
| 6 | 2025005 | 56.789.012-3 |
| 7 | 2025006 | 67.890.123-4 |
| 8 | 2025007 | 78.901.234-5 |
| 9 | 2025008 | 89.012.345-6 |
| 10 | 2025009 | 90.123.456-7 |
| **11** | 2025010 | 01.234.567-8 |

Figura 5: Amostra do dataset acadêmico (Tabela 3: Matrículas).

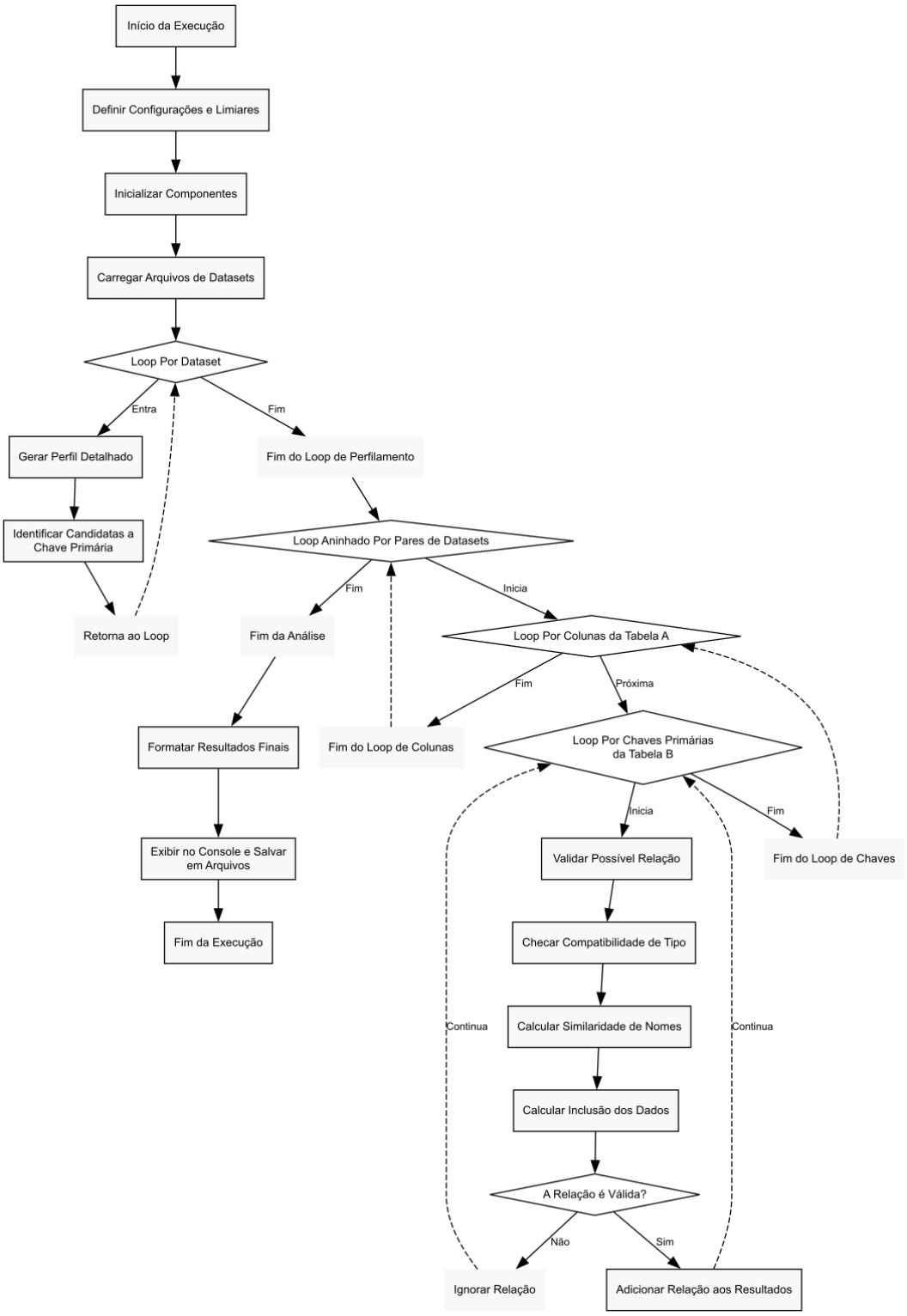


Figura 6: Lógica principal da API do Protótipo 3. Dataset A e Dataset B representam as tabelas sendo comparadas. (Documento de minha autoria.)

**Relatório de Saída (Exemplo):**

{

```
  "status": "success",
  "analysis_summary": {
    "processed_files": 2,
    "relationships_found": 1
  },
  "relationships": [
    {
      "source_table": "Player Play By Play.csv",
      "source_column": "seas_id",
      "target_table": "Advanced.csv",
      "target_column": "seas_id",
      "confidence": {
        "inclusion_score": 1.0,
        "name_similarity": 1.0
      }
    }
  ]
}
```


Os Ajustes propostos após o P3 foram focados em robustez. Os especialistas levantaram a preocupação sobre dados "sujos"e a dependência do sistema em metadados de alta qualidade (presença de cabeçalhos) para o `AttributeMatcher` funcionar. O ajuste necessário para o Protótipo 4 foi blindar o sistema contra cenários de degradação de metadados, usando dados sintéticos.

## 5.4 Protótipo 4: Robustez e Tratamento de Dados Degradados

O objetivo do Protótipo 4 foi unicamente de robustez: implementar o "quarto algoritmo", projetado para operar em cenários onde os metadados são parciais ou inexistentes (sem cabeçalhos).

A arquitetura não envolveu novas bibliotecas. As modificações foram puramente lógicas no `DataLoader` (para aceitar um parâmetro `header=None` via API e repassá-lo ao `pandas.read_csv`) e no `FKFinder`. A lógica do `FKFinder` foi alterada para que, caso a similaridade de nome falhasse, o algoritmo prosseguisse para a validação de conteúdo (análise de inclusão) como critério decisivo, em vez de descartar a relação.

Para os testes, foi construído um script (`gerar_dados_artificiais.py`) usando Pandas e NumPy para criar três conjuntos de datasets sintéticos. A validação foi dividida em dois grupos de testes:

**Grupo 1: Identificação de Chaves Candidatas.** Utilizou-se o conjunto `datasets_teste1_normais` (com cabeçalhos) para verificar a capacidade do algoritmo de identificar corretamente as chaves. Neste grupo, vale destacar o teste com a Tabela 5 (`..._normal5.PNG`), que contém múltiplas colunas identificadoras (Matrícula, Email, CPF e RG) na mesma tabela. Isso permitiu avaliar como o algoritmo lida com a identificação de pares chave candidata/chave estrangeira, não se limitando apenas à chave primária.

**Grupo 2: Robustez a Metadados Degradados.** Utilizou-se os conjuntos `datasets_teste2_genericos` (cabeçalhos genéricos como 'col1', 'col2') e `datasets_teste3_sem_cabecalho` para validar se o algoritmo conseguia inferir as relações baseando-se apenas no conteúdo (inclusão de dados), ignorando a ausência de nomes significativos nas colunas.

### Datasets Utilizados

- `..._normal1.csv`: colunas `RG`, `nome-disciplina`.
- `..._normal2.csv`: colunas `matricula-aluno`, `codigo-disciplina`.
- `..._normal3.csv`: colunas `codigo-disciplina`, `nome-disciplina`, `departamento`.
- `..._normal4.csv`: colunas `placa`, `CPF`, `data-aquisicao`.
- `..._normal5.csv`: colunas `matricula-aluno`, `nome`, `email`, `CPF`, `RG`, `data-nascimento`.
- `..._generico1.csv` a `..._generico5.csv`: colunas `A`, `B`, `C`, ... (etc.)
- `..._sem_cabecalho1.csv` a `..._sem_cabecalho5.csv`: sem nomes de colunas.

| 1 | RG | nome-disciplina |
|---|---|---|
| 2 | 361204875 | Algoritmos e Estruturas de Dados |
| 3 | 427810632 | Fisica Basica I |
| 4 | 12345866 | Algoritmos e Estruturas de Dados |
| 5 | 12345866 | Quimica Geral |
| 6 | 572306180 | Algoritmos e Estruturas de Dados |
| 7 | 410365282 | Fisica Basica I |
| 8 | 531076829 | Algoritmos e Estruturas de Dados |
| 9 | 612057835 | Algoritmos e Estruturas de Dados |
| 10 | 385624013 | Fisica Basica I |
| 11 | 758413269 | Calculo II |

Figura 7: Amostra do dataset sintético 'normal' (Tabela 1).

| 1 | matricula-aluno | codigo-disciplina |
|---|---|---|
| 2 | 20250099 | INF1001 |
| 3 | 20250067 | QUI1004 |
| 4 | 20250079 | MAT1002 |
| 5 | 20250037 | FIS1003 |
| 6 | 20250037 | MAT1002 |
| 7 | 20250038 | MAT1002 |
| 8 | 20250055 | INF1001 |
| 9 | 20250088 | FIS1003 |
| 10 | 20250051 | INF1001 |
| 11 | 20250016 | INF1001 |

Figura 8: Amostra do dataset sintético 'normal' (Tabela 2).

| 1 | codigo-disciplina | nome-disciplina | departamento |
|---|---|---|---|
| 2 | INF1001 | Algoritmos e Estruturas de Dados | Informatica |
| 3 | MAT1002 | Calculo II | Matematica |
| 4 | FIS1003 | Fisica Basica I | Fisica |
| 5 | QUI1004 | Quimica Geral | Quimica |

Figura 9: Amostra do dataset sintético 'normal' (Tabela 3).

| 1 | placa | CPF | data-aquisição |
|---|---|---|---|
| 2 | VAJ-3R18 | 256.897.031-68 | 2025-03-13 |
| 3 | ELB-4I56 | 791.506.834-20 | 2020-12-29 |
| 4 | NFD-4G64 | 578.694.201-58 | 2023-01-01 |
| 5 | SFV-5J31 | 940.375.182-79 | 2023-09-26 |
| 6 | JNF-3O34 | 256.418.937-73 | 2023-03-12 |
| 7 | VOA-6T78 | 498.023.716-96 | 2021-05-10 |
| 8 | WVQ-2O54 | 612.840.359-05 | 2025-03-11 |
| 9 | GOG-9H89 | 571.396.824-37 | 2022-08-27 |
| 10 | RCC-9V71 | 792.168.543-91 | 2023-03-31 |
| 11 | CRP-1N59 | 021.739.546-52 | 2021-09-18 |

Figura 10: Amostra do dataset sintético 'normal' (Tabela 4).

| 1 | matricula-aluno | nome | email | CPF |
|---|---|---|---|---|
| 2 | 20250001 | Srta. Alana Casa Grande | mendoncaevelyn@example.net | 270.413.598-32 |
| 3 | 20250002 | Lorena Monteiro | danielaragao@example.net | 453.912.876-00 |
| 4 | 20250003 | Maria Isis Oliveira | caua75@example.org | 326.978.051-68 |
| 5 | 20250004 | Mateus Lopes | gustavo-henrique23@example.net | 549.630.128-98 |
| 6 | 20250005 | Zoe Costa | eduardorezende@example.net | 940.375.182-79 |
| 7 | 20250006 | Ágatha Brito | da-luzjosue@example.net | 520.498.731-23 |
| 8 | 20250007 | Aylla Barbosa | macedonatalia@example.com | 096.412.587-02 |
| 9 | 20250008 | Rebeca Vasconcelos | camila12@example.net | 201.587.943-97 |
| 10 | 20250009 | Beatriz Ferreira | aragaotheodoro@example.org | 076.145.389-00 |
| 11 | 20250010 | Theo Aparecida | vinicius66@example.com | 791.506.834-20 |
| 12 | 20250011 | Dra. Maria Sophia Jesus | isaque80@example.com | 274.698.150-58 |
| 13 | 20250012 | Isis Barros | benicioda-paz@example.org | 792.168.543-91 |
| 14 | 20250013 | João Moreira | ysiqueira@example.org | 165.023.974-25 |
| 15 | 20250014 | Luigi Sousa | gomesarthur@example.org | 326.081.475-26 |

Figura 11: Amostra do dataset sintético 'normal' (Tabela 5).

| 1 | A | B |
|---|---|---|
| 2 | 361204875 | Algoritmos e Estruturas de Dados |
| 3 | 427810632 | Fisica Basica I |
| 4 | 12345866 | Algoritmos e Estruturas de Dados |
| 5 | 12345866 | Quimica Geral |
| 6 | 572306180 | Algoritmos e Estruturas de Dados |
| 7 | 410365282 | Fisica Basica I |
| 8 | 531076829 | Algoritmos e Estruturas de Dados |
| 9 | 612057835 | Algoritmos e Estruturas de Dados |
| 10 | 385624013 | Fisica Basica I |
| 11 | 758413269 | Calculo II |

Figura 12: Amostra do dataset sintético 'genérico' (Tabela 1).

| 1 | A | B |
|---|---|---|
| 2 | 20250099 | INF1001 |
| 3 | 20250067 | QUI1004 |
| 4 | 20250079 | MAT1002 |
| 5 | 20250037 | FIS1003 |
| 6 | 20250037 | MAT1002 |
| 7 | 20250038 | MAT1002 |
| 8 | 20250055 | INF1001 |
| 9 | 20250088 | FIS1003 |
| 10 | 20250051 | INF1001 |
| 11 | 20250016 | INF1001 |

Figura 13: Amostra do dataset sintético 'genérico' (Tabela 2).

| 1 | A | B | C |
|---|---|---|---|
| 2 | INF1001 | Algoritmos e Est | Informatica |
| 3 | MAT1002 | Calculo II | Matematica |
| 4 | FIS1003 | Fisica Basica I | Fisica |
| 5 | QUI1004 | Quimica Geral | Quimica |

Figura 14: Amostra do dataset sintético 'genérico' (Tabela 3).

| 1 | A | B | C |
|---|---|---|---|
| 2 | VAJ-3R18 | 256.897.031-68 | 2025-03-13 |
| 3 | ELB-4I56 | 791.506.834-20 | 2020-12-29 |
| 4 | NFD-4G64 | 578.694.201-58 | 2023-01-01 |
| 5 | SFV-5J31 | 940.375.182-79 | 2023-09-26 |
| 6 | JNF-3O34 | 256.418.937-73 | 2023-03-12 |
| 7 | VOA-6T78 | 498.023.716-96 | 2021-05-10 |
| 8 | WVQ-2O54 | 612.840.359-05 | 2025-03-11 |
| 9 | GOG-9H89 | 571.396.824-37 | 2022-08-27 |
| 10 | RCC-9V71 | 792.168.543-91 | 2023-03-31 |
| 11 | CRP-1N59 | 021.739.546-52 | 2021-09-18 |

Figura 15: Amostra do dataset sintético 'genérico' (Tabela 4).

| 1 | A | B | C | D |
|---|---|---|---|---|
| 2 | 20250001 | Srta. Alana Casa Grande | mendoncaevelyn@example.net | 270.413.598-32 |
| 3 | 20250002 | Lorena Monteiro | danielaragao@example.net | 453.912.876-00 |
| 4 | 20250003 | Maria Isis Oliveira | caua75@example.org | 326.978.051-68 |
| 5 | 20250004 | Mateus Lopes | gustavo-henrique23@example.net | 549.630.128-98 |
| 6 | 20250005 | Zoe Costa | eduardorezende@example.net | 940.375.182-79 |
| 7 | 20250006 | Ágatha Brito | da-luzjosue@example.net | 520.498.731-23 |
| 8 | 20250007 | Aylla Barbosa | macedonatalia@example.com | 096.412.587-02 |
| 9 | 20250008 | Rebeca Vasconcelos | camila12@example.net | 201.587.943-97 |
| 10 | 20250009 | Beatriz Ferreira | aragaotheodoro@example.org | 076.145.389-00 |
| 11 | 20250010 | Theo Aparecida | vinicius66@example.com | 791.506.834-20 |
| 12 | 20250011 | Dra. Maria Sophia Jesus | isaque80@example.com | 274.698.150-58 |
| 13 | 20250012 | Isis Barros | benicioda-paz@example.org | 792.168.543-91 |
| 14 | 20250013 | João Moreira | ysiqueira@example.org | 165.023.974-25 |
| 15 | 20250014 | Luigi Sousa | gomesarthur@example.org | 326.081.475-26 |

Figura 16: Amostra do dataset sintético 'genérico' (Tabela 5).

| | | |
|---|---|---|
| 1 | 361204875 | Algoritmos e Estruturas de Dados |
| 2 | 427810632 | Fisica Basica I |
| 3 | 12345866 | Algoritmos e Estruturas de Dados |
| 4 | 12345866 | Quimica Geral |
| 5 | 572306180 | Algoritmos e Estruturas de Dados |
| 6 | 410365282 | Fisica Basica I |
| 7 | 531076829 | Algoritmos e Estruturas de Dados |
| 8 | 612057835 | Algoritmos e Estruturas de Dados |
| 9 | 385624013 | Fisica Basica I |
| 10 | 758413269 | Calculo II |
| 11 | 251684738 | Algoritmos e Estruturas de Dados |

Figura 17: Amostra do dataset sintético 'sem cabeçalho' (Tabela 1).

| | | |
|---|---|---|
| 1 | 20250099 | INF1001 |
| 2 | 20250067 | QUI1004 |
| 3 | 20250079 | MAT1002 |
| 4 | 20250037 | FIS1003 |
| 5 | 20250037 | MAT1002 |
| 6 | 20250038 | MAT1002 |
| 7 | 20250055 | INF1001 |
| 8 | 20250088 | FIS1003 |
| 9 | 20250051 | INF1001 |
| 10 | 20250016 | INF1001 |
| 11 | 20250010 | MAT1002 |

Figura 18: Amostra do dataset sintético 'sem cabeçalho' (Tabela 2).

| | | | |
|---|---|---|---|
| 1 | INF1001 | Algoritmos e Estruturas de Dados | Informatica |
| 2 | MAT1002 | Calculo II | Matematica |
| 3 | FIS1003 | Fisica Basica I | Fisica |
| 4 | QUI1004 | Quimica Geral | Quimica |

Figura 19: Amostra do dataset sintético 'sem cabeçalho' (Tabela 3).

| | | | |
|---|---|---|---|
| 1 | VAJ-3R18 | 256.897.031-68 | 2025-03-13 |
| 2 | ELB-4I56 | 791.506.834-20 | 2020-12-29 |
| 3 | NFD-4G64 | 578.694.201-58 | 2023-01-01 |
| 4 | SFV-5J31 | 940.375.182-79 | 2023-09-26 |
| 5 | JNF-3O34 | 256.418.937-73 | 2023-03-12 |
| 6 | VOA-6T78 | 498.023.716-96 | 2021-05-10 |
| 7 | WVQ-2O54 | 612.840.359-05 | 2025-03-11 |
| 8 | GOG-9H89 | 571.396.824-37 | 2022-08-27 |
| 9 | RCC-9V71 | 792.168.543-91 | 2023-03-31 |
| 10 | CRP-1N59 | 021.739.546-52 | 2021-09-18 |
| 11 | PNS-5N46 | 876.409.325-56 | 2025-05-28 |

Figura 20: Amostra do dataset sintético 'sem cabeçalho' (Tabela 4).

| | | | | |
|---|---|---|---|---|
| 1 | 20250001 | Srta. Alana Casa Grande | mendoncaevelyn@example.net | 270.413.598-32 |
| 2 | 20250002 | Lorena Monteiro | danielaragao@example.net | 453.912.876-00 |
| 3 | 20250003 | Maria Isis Oliveira | caua75@example.org | 326.978.051-68 |
| 4 | 20250004 | Mateus Lopes | gustavo-henrique23@example.net | 549.630.128-98 |
| 5 | 20250005 | Zoe Costa | eduardorezende@example.net | 940.375.182-79 |
| 6 | 20250006 | Ágatha Brito | da-luzjosue@example.net | 520.498.731-23 |
| 7 | 20250007 | Aylla Barbosa | macedonatalia@example.com | 096.412.587-02 |
| 8 | 20250008 | Rebeca Vasconcelos | camila12@example.net | 201.587.943-97 |
| 9 | 20250009 | Beatriz Ferreira | aragaotheodoro@example.org | 076.145.389-00 |
| 10 | 20250010 | Theo Aparecida | vinicius66@example.com | 791.506.834-20 |
| 11 | 20250011 | Dra. Maria Sophia Jesus | isaque80@example.com | 274.698.150-58 |
| 12 | 20250012 | Isis Barros | benicioda-paz@example.org | 792.168.543-91 |
| 13 | 20250013 | João Moreira | ysiqueira@example.org | 165.023.974-25 |
| 14 | 20250014 | Luigi Sousa | gomesarthur@example.org | 326.081.475-26 |
| 15 | 20250015 | Liz Novais | camargomaria-alice@example.org | 021.739.546-52 |

Figura 21: Amostra do dataset sintético 'sem cabeçalho' (Tabela 5).

Os resultados foram um sucesso. Os testes sintéticos provaram que o "quarto algoritmo"era robusto: o Teste 2 forçou o sistema a ignorar o `AttributeMatcher` e funcionar apenas com a inclusão de conteúdo, e o Teste 3 validou a capacidade de operar por índices de coluna.

**Relatório de Saída (Exemplo):**

```
[INFO] Configuração recebida: header=None
[INFO] Carregando ALUNO.csv (sem cabeçalho)...
```

```
[INFO] Carregando MATRICULA.csv (sem cabeçalho)...
[WARN] Similaridade de nome baixa para 'col_0' vs 'col_1'.
[INFO] Iniciando validação de conteúdo profunda...
[RESULT] Relação Identificada (Baseada em Conteúdo):
  Origem: MATRICULA.csv [col_1]
  Destino: ALUNO.csv [col_0]
  Inclusão: 100%
```

Os Ajustes identificados foram mínimos, pois os testes controlados provaram a robustez. A limitação final era que esta robustez ainda não havia sido testada em um ambiente real e desconhecido. O ajuste proposto para o Protótipo 5 foi a validação final da generalização do sistema.

## 5.5 Protótipo 5: Validação Final e Generalização

O objetivo do Protótipo 5 foi a validação final: aplicar o sistema robusto do P4, sem modificações de código, a um conjunto real de datasets e de um domínio totalmente desconhecido. O "quinto algoritmo"era, portanto, o mesmo do P4.

A arquitetura foi a implementação estável do P4, sem desenvolvimento de novas funcionalidades.

Os testes utilizaram um novo conjunto de 10 datasets reais do portal `dados.gov.br`, focados na área de Pesca (ex: `BA_Pescadores(in).csv`, `Base_de_Dados_da_Sardinha-verdadeira(in).csv`). Estes arquivos continham desafios do mundo real, como delimitadores de ponto e vírgula (;) e diferentes encodings de texto.

**Datasets Utilizados**

- `..._5_1.csv`: colunas `CPF`, `NomedoPescador`, `DtNascimento`, `NivelEscolaridade`, `TipoAlfabetizacao`, `Sexo`, `Categoria`, `FormadeAtuac`, `UF`, `Municipio`.
- `..._5_2.csv`: colunas `idMapaDeBordo`, `modeloMapaDeBordo`, `portoSaida`, `portoChegada`, `dataSaida`, `dataChegada`, `nomeEmbarcacao`, `numRegEmbarcacao`, `codUfEmbarcacao`, `UfEmbarcacao`, `compEmbarcacao`, `numTrcEmbarcacao`.
- `..._5_3.csv`: colunas `Nome/RazaoSocial`, `VinculocomoRGP`, `NumeronoRGP`, `cidade`, `UF`, `TipodePessoa`.
- `..._5_4.csv`: colunas `Nome/RazaoSocial`, `VinculocomoRGP`, `NumeronoRGP`, `cidade`, `UF`, `TipodePessoa`.

- ..._5_5.csv: colunas Nome/RazaoSocial, NumeronoRGP, Sistemadecultivo, Tipodeatividade, Cidade/UF, UF, TipodeRegistro, Genero.
- ..._5_6.csv: colunas IdEntradaEmEmpresa, Datadorecebimento, Pesodetainharecebido(kg), Tipodeprodutor.
- ..._5_7.csv: colunas idMapaDeBordo, NomedaEmbarcacao, DatadeSaida, DatadeChegada, Ano, CodigoIN, Especie, Producao(t).
- ..._5_8.csv: colunas NomedaEmbarcacao, NumerodeInscricaonoRGP, PPPouHAEP, NumerodeInscricaonaMarinhadoBrasil, AB, Comprimento, Propulsao, HP, CapacidadedePorao, VolumedoTanque, Tripulacao, NumerodeConves, Status, UF, NumerodaFrota, CodigoIN, Petrecho.
- ..._5_9.csv: colunas Codigo, Datadorelatorio, Especie, Captura(Kg), ValorR$/Kg.
- ..._5_10.csv: colunas CPF, NomedoPescador, DtNascimento, NivelEscolaridade, TipoAlfabetizacao, Sexo, Categoria, FormadeAtuac, UF, Municipio.

| 1 | CPF | Nome do Pescador | Dt Nascimento | Nivel Escolaridade |
|---|---|---|---|---|
| 2 | ***.835.157-** | MARIA ********** | 1953 | ENSINO MEDIO COMPLETO |
| 3 | ***.837.228-** | HERMES ********** | 1957 | ENSINO MEDIO INCOMPLETO |
| 4 | ***.865.893-** | JOSE ********** | 1983 | ENSINO MEDIO INCOMPLETO |
| 5 | ***.908.883-** | JACIANA ********** | 1974 | PRIMEIRO QUARTO ANO COMPLETO |
| 6 | ***.919.745-** | CARLOS ********** | 1981 | ENSINO MEDIO COMPLETO |
| 7 | ***.937.793-** | RAIMUNDO ********** | 1980 | ENSINO MEDIO COMPLETO |
| 8 | ***.947.217-** | SEVERINO ********** | 1969 | ENSINO MEDIO COMPLETO |
| 9 | ***.962.826-** | LAURIDE ********** | 1967 | PRIMEIRO QUARTO ANO COMPLETO |
| 10 | ***.967.435-** | ORLEY ********** | 1979 | PRIMEIRO QUARTO ANO INCOMPLETO |
| 11 | ***.987.745-** | LUCIA ********** | 1982 | ENSINO MEDIO COMPLETO |
| 12 | ***.000.375-** | COSME ********** | 1980 | ENSINO MEDIO COMPLETO |
| 13 | ***.035.567-** | SERGIO ********** | 1970 | QUINTO NONO ANO COMPLETO |
| 14 | ***.042.293-** | MARIA ********** | 1967 | QUINTO NONO ANO INCOMPLETO |
| 15 | ***.101.533-** | THALLES ********** | 1985 | ENSINO MEDIO COMPLETO |

Figura 22: Amostra do dataset real gov.br (Tabela 1).

| 1 | idMapaDeBordo | modeloMapaDeBordo | portoSaida | portoChegada |
|---|---|---|---|---|
| 2 | 4 | Armadilha | Pontas de pedra goiana PE | Pontas de Pedra goiana pe |
| 3 | 5 | Armadilha | Pontas de pedra goiana PE | Pontas de Pedra goiana pe |
| 4 | 13 | Armadilha | URUAU - BEBERIBE - CE | URUAU - BEBERIBE - CE |
| 5 | 22 | Armadilha | Pontas de pedra goiana PE | Pontas de Pedra goiana pe |
| 6 | 23 | Armadilha | Pontas de pedra goiana PE | Pontas de Pedra goiana pe |
| 7 | 24 | Armadilha | URUAU - BEBERIBE | URUAU - BEBERIBE |
| 8 | 46 | Armadilha | uruau - beberibe | URUAU - BEBERIBE |
| 9 | 82 | Armadilha | URUAU - BEBERIBE | URUAU - BEBERIBE |
| 10 | 100 | Armadilha | URUAU - BEBERIBE | URUAU - BEBERIBE |
| 11 | 106 | Arrasto Camarões | ANGRA DOS REIS | ANGRA DOS REIS |
| 12 | 108 | Armadilha | URUAU - BEBERIBE | URUAU - BEBERIBE |
| 13 | 135 | Rede de Cerco - Sardinha | Itajaí | Itajaí |
| 14 | 136 | Armadilha | BALEIA - ITAPIPOCA | BALEIA - ITAPIPOCA |
| 15 | 139 | Armadilha | uruau - beberibe | uruau - beberibe |

Figura 23: Amostra do dataset real `gov.br` (Tabela 2).

| 1 | Nome / Razão Social | Vínculo com o RGP | Número no RGP | cidade |
|---|---|---|---|---|
| 2 | 41.956.777 ********** | Indústria de pesca | PAI00042010 | Salinópolis,PA,Brasil |
| 3 | 47.219.981 ********** | Indústria de pesca | CEI00041216 | Fortaleza,CE,Brasil |
| 4 | 52.141.288 ********** | Indústria de pesca | SPI00043004 | Severínia,SP,Brasil |
| 5 | 53.577.455 ********** | Indústria de pesca | SPI00042946 | Igaratá,SP,Brasil |
| 6 | 56.051.950 ********** | Indústria de pesca | PAI00043288 | Abaetetuba,PA,Brasil |
| 7 | 57.815.782 **** | Indústria de pesca | PAI00042978 | Belém,PA,Brasil |
| 8 | 60.524.571 ********** | Indústria de pesca | PRI00043886 | Cambé,PR,Brasil |
| 9 | A. ********** | Indústria de pesca | BAI00034537 | Porto Seguro,BA,Brasil |
| 10 | A ********** | Indústria de pesca | BAI00026844 | Prado,BA,Brasil |
| 11 | A ********** | Indústria de pesca | BAI00026844 | Prado,BA,Brasil |
| 12 | A.C.P. ******** | Indústria de pesca | AMI00040938 | Juruá,AM,Brasil |
| 13 | ACQUA ********** | Indústria de pesca | SPI00043054 | São Paulo,SP,Brasil |
| 14 | ACQUA ********** | Indústria de pesca | RJI00044000 | Guapimirim,RJ,Brasil |
| 15 | ACQUA ********** | Indústria de pesca | SPI00033937 | Bauru,SP,Brasil |

Figura 24: Amostra do dataset real `gov.br` (Tabela 3).

| 1 | Nome / Razão Social | Vínculo com o RGP | Número no RGP | cidade |
|---|---|---|---|---|
| 2 | 58.555.075 ********** | Armador de pesca | PAA00043676 | Belém,PA,Brasil |
| 3 | A.B. ********** | Armador de pesca | PAA00025258 | Bragança,PA,Brasil |
| 4 | ABELARDO ********** | Armador de pesca | SCA00020614 | Bombinhas,SC,Brasil |
| 5 | ABEL ******** | Armador de pesca | SCA00023746 | Balneário Camboriú,SC,Brasil |
| 6 | Abel ******* | Armador de pesca | SCA00019961 | Florianópolis,SC,Brasil |
| 7 | ABENIR ********** | Armador de pesca, Proprietário de embarcação | SCP10302638,SCA00013491 | Florianópolis,SC,Brasil |
| 8 | Abercio ********** | Armador de pesca | SCP10320587,SCA00022458 | Navegantes,SC,Brasil |
| 9 | ACACIO ********** | Armador de pesca, Proprietário de embarcação | SCP05657381,SCA00022038 | Porto Belo,SC,Brasil |
| 10 | ACACIO ********** | Armador de pesca, Proprietário de embarcação | SCP10340038,SCA00011307 | Bombinhas,SC,Brasil |
| 11 | Acacio ********** | Armador de pesca | SCP10284239,SCA00039575 | Florianópolis,SC,Brasil |
| 12 | ACACIO ********** | Armador de pesca, Proprietário de embarcação | SCA00014961 | Florianópolis,SC,Brasil |
| 13 | Acari ********** | Armador de pesca, Proprietário de embarcação | SPA00023068 | Santos,SP,Brasil |
| 14 | A ********** | Armador de pesca | PAA00031411 | Belém,PA,Brasil |
| 15 | ACILDO ********** | Armador de pesca, Proprietário de embarcação | CEA00027050 | Acaraú,CE,Brasil |

Figura 25: Amostra do dataset real `gov.br` (Tabela 4).

| 1 | Nome / Razão Social | Número no RGP | Sistema de cultivo | Tipo de atividade |
|---|---|---|---|---|
| 2 | VANDERSON ********** | SP-U1158710-8 | Semi-intensivo | Cultivo de Espécies Ornamentais (Produção de Ornamentais) |
| 3 | CLAUDIO ********** | SP-U1158471-1 | Intensivo | Cultivo de Formas Jovens (Produção de Formas Jovens) |
| 4 | LAYSA ********** | SP-U1158719-7 | Intensivo | Cultivo de Espécies Ornamentais (Produção de Ornamentais) |
| 5 | ANDERSON ********** | RN-U1158042-4 | Semi-intensivo | Cultivo de Espécies Ornamentais (Produção de Ornamentais) |
| 6 | MARCELO ********** | RJ-U1158676-8 | Semi-intensivo | Cultivo de Camarão Marinho (Carcinicultura Marinha) |
| 7 | DANIELLE ********** | RN-U1158698-4 | Semi-intensivo | Cultivo de Espécies Ornamentais (Produção de Ornamentais) |
| 8 | ACQUADOCE ********** | CE-U1155829-1 | Extensivo | Cultivo de Moluscos (Malacocultura) |
| 9 | AcquaFarm **** | ES-U1158354-4 | Semi-intensivo | Cultivo de Camarão Marinho (Carcinicultura Marinha) |
| 10 | ADALVO **** | RO-U1154204-0 | Semi-intensivo | Cultivo de peixes em Tanque escavado (Piscicultura em Tanque escavado / Edificado) |
| 11 | Adelson ********** | RN-U1158273-7 | Semi-intensivo | Cultivo de peixes em Tanque escavado (Piscicultura em Tanque escavado / Edificado) |
| 12 | ADEMAR ********** | RO-U1150677-1 | Semi-intensivo | Cultivo de peixes em Tanque escavado (Piscicultura em Tanque escavado / Edificado) |
| 13 | Ademar ******* | MA-U0002861-5 | Semi-intensivo | Cultivo de peixes em Tanque escavado (Piscicultura em Tanque escavado / Edificado) |
| 14 | ADEMIR ********** | SP-U0001659-7 | Semi-intensivo | Cultivo de peixes em Tanque escavado (Piscicultura em Tanque escavado / Edificado) |
| 15 | ADEMIR ********** | PR-U1157550-8 | Semi-intensivo | Cultivo de peixes em Tanque escavado (Piscicultura em Tanque escavado / Edificado) |

Figura 26: Amostra do dataset real `gov.br` (Tabela 5).

| 1 | IdEntradaEmEmpresa | Data do recebimento | Peso de tainha recebido (kg) | Tipo de produtor |
|---|---|---|---|---|
| 2 | 0 | 26/07/2019 | 12680 | Cerco/Traineira |
| 3 | 1 | 26/07/2019 | 10980 | Cerco/Traineira |
| 4 | 2 | 26/07/2019 | 10420 | Cerco/Traineira |
| 5 | 3 | 27/07/2019 | 8120 | Cerco/Traineira |
| 6 | 4 | 16/07/2019 | 20840 | Cerco/Traineira |
| 7 | 5 | 17/07/2019 | 10220 | Cerco/Traineira |
| 8 | 6 | 31/07/2019 | 10640 | Cerco/Traineira |
| 9 | 7 | 11/07/2019 | 38000 | Cerco/Traineira |
| 10 | 8 | 11/07/2019 | 5920 | Cerco/Traineira |
| 11 | 9 | 11/07/2019 | 240 | Cerco/Traineira |
| 12 | 10 | 19/07/2019 | 18000 | Cerco/Traineira |
| 13 | 11 | 27/07/2019 | 8960 | Cerco/Traineira |
| 14 | 12 | 02/08/2019 | 6000 | Cerco/Traineira |
| 15 | 13 | 10/07/2019 | 49140 | Cerco/Traineira |

Figura 27: Amostra do dataset real `gov.br` (Tabela 6).

| 1 | idMapaDeBordo | Nome da Embarcação | Data de Saída | Data de Chegada |
|---|---|---|---|---|
| 2 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 3 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 4 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 5 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 6 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 7 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 8 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 9 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 10 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 11 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 12 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 13 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 14 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |
| 15 | 1 | ALYSSON | 01/05/2022 | 20/05/2022 |

Figura 28: Amostra do dataset real `gov.br` (Tabela 7).

| 1 | Nome da Embarcação | Número de Inscrição no RGP (PPP ou RAEP) | Número de Inscrição na Marinha do Brasil | AB |
|---|---|---|---|---|
| 2 | 7 | RJ00070751 | 3870058803 | 7 |
| 3 | 7 | CE00253345 | 161M2013001190 | |
| 4 | 02 DE JULHO | CE00033717 | 1620022150 | 0,9 |
| 5 | 02 GUERREIROS | PA00161265 | 230095631 | 2 |
| 6 | 03 IRMAOS | AP00244798 | 220088276 | 4,53 |
| 7 | 03 IRMÃOS | BA00002978 | 2810275254 | 12,3 |
| 8 | 03 MAIO II | ES00131160 | 3430045851 | 4 |
| 9 | 03 NETINHOS | SC00180187 | 4420222727 | 14,5 |
| 10 | 04 DE ABRIL | CE00251427 | 161M2010000851 | |
| 11 | 04 DE JULHO | CE00013379 | 1620014009 | 5 |
| 12 | 11 DE NOVEMBRO | RJ00130718 | 3877041710 | 0,7 |
| 13 | 12 APÓSTOLOS | CE00248690 | 1630046035 | 31 |
| 14 | 12 DE AGOSTO | MA00273719 | 1210141124 | 1 |
| 15 | 13 DE JUNHO | BA00210171 | 2930018321 | 1,9 |

Figura 29: Amostra do dataset real `gov.br` (Tabela 8).

| 1 | Código | Data do relatório | Espécie | Captura (Kg) |
|---|---|---|---|---|
| 2 | 1967320 | 02/03/2022 | Sardinha-verdadeira | 5.000,00 |
| 3 | 1790234 | 02/03/2022 | Sardinha-verdadeira | 46.756,00 |
| 4 | 1786734 | 02/03/2022 | Sardinha-verdadeira | 34.000,00 |
| 5 | 2143667 | 04/03/2022 | Sardinha-verdadeira | 42.000,00 |
| 6 | 1916667 | 04/03/2022 | Sardinha-verdadeira | 58.356,00 |
| 7 | 2143667 | 05/03/2022 | Sardinha-verdadeira | 20.000,00 |
| 8 | 1967320 | 05/03/2022 | Sardinha-verdadeira | 30.000,00 |
| 9 | 1967320 | 05/03/2022 | Sardinha-verdadeira | 65.000,00 |
| 10 | 1916667 | 05/03/2022 | Sardinha-verdadeira | 135.458,00 |
| 11 | 1916667 | 05/03/2022 | Sardinha-verdadeira | 64.275,00 |
| 12 | 1790234 | 05/03/2022 | Sardinha-verdadeira | 27.838,00 |
| 13 | 1790234 | 05/03/2022 | Sardinha-verdadeira | 130.814,00 |
| 14 | 1786734 | 05/03/2022 | Sardinha-verdadeira | 60.000,00 |
| 15 | 1786734 | 06/03/2022 | Sardinha-verdadeira | 145.000,00 |

Figura 30: Amostra do dataset real `gov.br` (Tabela 9).

| 1 | CPF | Nome do Pescador | Dt Nascimento | Nivel Escolaridade |
|---|---|---|---|---|
| 2 | ***.754.765-** | BENEDITO ********** | 1976 | PRIMEIRO QUARTO ANO INCOMPLETO |
| 3 | ***.755.115-** | EDIVANIA ********** | 1982 | ENSINO MEDIO COMPLETO |
| 4 | ***.755.865-** | EDIRAN ********** | 1976 | QUINTO NONO ANO INCOMPLETO |
| 5 | ***.757.845-** | ROSELI ********** | 1971 | PRIMEIRO QUARTO ANO INCOMPLETO |
| 6 | ***.758.751-** | NIVALDO ********** | 1975 | ENSINO MEDIO INCOMPLETO |
| 7 | ***.760.295-** | CERIALVA ********** | 1981 | QUINTO NONO ANO COMPLETO |
| 8 | ***.761.125-** | OSVALDO ********** | 1980 | PRIMEIRO QUARTO ANO INCOMPLETO |
| 9 | ***.761.135-** | IRECI ********** | 1984 | QUINTO NONO ANO INCOMPLETO |
| 10 | ***.761.865-** | SILVANA ********** | 1979 | ENSINO MEDIO COMPLETO |
| 11 | ***.761.975-** | GENAILDE ********** | 1977 | PRIMEIRO QUARTO ANO COMPLETO |
| 12 | ***.761.985-** | MARIA ********** | 1973 | PRIMEIRO QUARTO ANO COMPLETO |
| 13 | ***.762.015-** | ERISVALDO ********** | 1978 | PRIMEIRO QUARTO ANO INCOMPLETO |
| 14 | ***.762.915-** | TATIANE ***** | 1982 | ENSINO MEDIO INCOMPLETO |
| 15 | ***.763.135-** | VILMAR ********** | 1975 | PRIMEIRO QUARTO ANO COMPLETO |

Figura 31: Amostra do dataset real `gov.br` (Tabela 10).

O sistema identificou com sucesso as relações esperadas no conjunto real de datasets oriundos do `gov.br`. A API do P5, já configurável, foi capaz de lidar com os novos delimitadores através de um parâmetro na requisição.

**Relatório de Saída (Exemplo):**

```
RELATÓRIO FINAL DE EXECUÇÃO - DADOS REAIS (GOV.BR)
================================================

[Relação 1 - Identificação de Embarcação]
```

```
Tabela Origem: Base de dados de captura da especie Pargo(in).
   csv
Coluna Origem: ID_EMBARCACAO
--> REF -->
Tabela Destino: Base de Dados das autorizações das Embarcaçõ
   es de Pesca.csv
Coluna Destino: ID_EMBARCACAO
Métricas: Inclusão 98.5% | Similaridade Nome 1.0

[Relação 2 - Registro Geral de Pesca]
Tabela Origem: BA Pescadores(in).csv
Coluna Origem: NR_RGP
--> REF -->
Tabela Destino: Base de dados de Registros de Aquicultores(
   in).csv
Coluna Destino: NR_RGP
Métricas: Inclusão 92.0% | Similaridade Nome 1.0
```

Esta fase provou que a arquitetura final do sistema atende a todos os objetivos do projeto (Seção 3): é uma ferramenta funcional (P2), integrável (P3), robusta (P4) e, finalmente, generalizável (P5). Na execução desta fase, a ferramenta demonstrou ser capaz de propor uma ferramenta (o próprio fluxo de execução), "identificar chaves estrangeiras"(o núcleo do `FKFinder`) e, crucialmente, fazê-lo em "contextos com metadados ausentes ou parciais", conforme comprovado pela robustez do P4 e validado em um cenário real no P5.

# 6 Considerações Finais

Este trabalho abordou o desafio central da descoberta de conhecimento em data lakes, onde a ausência de metadados explícitos impede a integração de datasets e, consequentemente, a realização de análises aprofundadas, por não indicarem que datasets podem potencialmente serem combinados para uma análise mais abrangente. O objetivo principal, de propor e implementar uma ferramenta para a identificação automática de relações em tais contextos, foi plenamente alcançado.

A principal contribuição deste projeto não é apenas um algoritmo singular, mas sim um framework de software robusto, validado através de uma metodologia de desenvolvimento iterativa de cinco protótipos. Esta abordagem permitiu que o sistema evoluísse de uma simples prova de conceito (Protótipo 1) para uma ferramenta de inferência relacional (Protótipo 2), que foi então validada por requisitos de mercado (Protótipo 3). Mais importante, o sistema foi metodicamente "blindado"contra os desafios do mundo real, provando sua capacidade de lidar com dados de baixa qualidade, como cabeçalhos genéricos ou ausentes (Protótipo 4).

Finalmente, o Protótipo 5 demonstrou a generalização e a eficácia da solução ao aplicá-la com sucesso a um domínio de dados real (`gov.br`), completamente distinto do domínio original, provando que a ferramenta é agnóstica ao contexto. Como resultado, o projeto entrega uma solução funcional que atende a todos os requisitos de engenharia propostos. O "quinto algoritmo"é capaz de ingerir múltiplos datasets, perfilá-los, identificar chaves candidatas e inferir relações de chave estrangeira com base em uma análise heurística de conteúdo e metadados, mesmo quando estes são parciais ou inexistentes.

Naturalmente, o trabalho apresenta limitações que abrem caminhos claros para futuras pesquisas. A principal limitação é a escalabilidade computacional; a análise de inclusão de conteúdo (`FKFinder`) é eficaz, mas computacionalmente intensiva, tornando-se um gargalo para cenários de big data com bilhões de registros. Além disso, a atual lógica de correspondência de atributos (`AttributeMatcher`) baseia-se em similaridade textual, não semântica, sendo incapaz de inferir que `cod_cli` e `id_comprador` representam o mesmo conceito. O sistema também está focado na identificação de chaves simples, não abordando o desafio de chaves primárias ou estrangeiras compostas.

Como trabalhos futuros, sugerem-se três caminhos principais:

- **Escalabilidade:** A adaptação da arquitetura para ambientes de processamento distribuído, como Spark ou Dask, para permitir a análise em volumes de dados massivos.

- **Análise Semântica:** A substituição da comparação textual por embeddings de linguagem (como BERT ou Word2Vec) para que o sistema possa comparar colunas pelo seu significado semântico, e não apenas pela ortografia.
- **Expansão da Lógica:** O desenvolvimento de heurísticas mais complexas no `DataProfiler` e no `FKFinder` para permitir a sugestão e validação de chaves compostas.
- **Novos Formatos:** Expandir a análise para outros formatos estruturados e semi-estruturados, como JSON e XML.

Conclui-se que este projeto cumpre integralmente seus objetivos, entregando uma metodologia de desenvolvimento e uma ferramenta de software validada, capaz de destravar o potencial analítico de dados previamente isolados em data lakes, viabilizando novas análises e gerando valor tangível a partir de informações desconexas.

# 7 Referências

[1] CARUCCIO, Loredana; DEUFEMIA, Vincenzo; POLESE, Giuseppe. Relaxed functional dependencies—a survey of approaches. *IEEE Transactions on Knowledge and Data Engineering*, v. 28, n. 1, p. 147-165, 2015.

[2] KHINE, Pwint Phyu; WANG, Zhao Shun. Data lake: a new ideology in big data era. In: *ITM web of conferences*. EDP Sciences, 2018. p. 03025.

[3] MILLER, Renée J. et al. Making Open Data Transparent: Data Discovery on Open Data. *IEEE Data Eng. Bull.*, v. 41, n. 2, p. 59-70, 2018.

[4] PAGANELLI, Matteo et al. Automated machine learning for entity matching tasks. In: *Advances in Database Technology-EDBT 2021, 24th International Conference on Extending Database Technology, Nicosia, Cyprus, March 23-26, Proceedings*. OpenProceedings. org, 2021. p. 325-330.

[5] PAPENBROCK, Thorsten et al. Data profiling with metanome. *Proceedings of the VLDB Endowment*, v. 8, n. 12, p. 1860-1863, 2015.

[6] ELMASRI, Ramez; NAVATHE, Shamkant B. *Fundamentals of Database Systems*. 6. ed. Boston: Addison-Wesley, 2010.