Universidade Federal do Paraná
Setor de Ciências Exatas
Departamento de Estatística
Programa de Especialização em *Data Science* e *Big Data*

Jeferson Derenevick

O processo e as etapas de um gerenciamento de metadados

Curitiba
2022

Jeferson Derenevick

O processo e as etapas de um gerenciamento de metadados

Monografia apresentada ao Programa de Especialização em Data Science e Big Data da Universidade Federal do Paraná como requisito parcial para a obtenção do grau de especialista.

Orientador: Eduardo Cunha De Almeida

Curitiba
2022

Especialização em Data Science e Big Data
Universidade Federal do Paraná
dsbd.leg.ufpr.br

O processo e as etapas de um gerenciamento de metadados

Jeferson Derenevick[1]
Eduardo Cunha De Almeida[2]

Resumo

Com o crescimento exponencial do volume de dados utilizados nas organizações, aliado às demandas relacionadas às entregas das análises para as tomadas de decisões, surge a necessidade do gerenciamento dos dados de forma mais presente.

Para gerenciar os dados, inicialmente precisamos entender o que são os metadados. De uma forma simples, podemos explicar que metadados são os dados sobre dados.

Este trabalho apresenta um processo de gestão dos metadados, de forma detalhada e dividido em suas principais etapas: partindo da seleção das fontes de dados; pela etapa da criação das tabelas do dicionário de dados; até o levantamento dos principais requisitos técnicos e de negócios escolhidos para a seleção de uma aplicação de catálogo de dados, capaz de interpretar as fontes de metadados em uma interface amigável para o usuário final. Os metadados e suas tabelas utilizados no projeto foram provenientes de uma startup de tecnologia.

O resultado desse trabalho serve como suporte para a compreensão, através das etapas dedicadas, de um processo de gestão de metadados, bem como seus desafios e boas práticas.

Palavras-chave: Metadados, Dicionário de dados, Catálogo de dados.

Abstract

With the exponential growth in the volume of data used in organizations, combined with the demands related to the delivery of analysis for decision-making, the need to manage data in a more present way arises.
To manage the data, we first need to understand what metadata is. In a simple way, we can explain that metadata is data about data.
This work presents a process of managing the metadata of technical and strategic types, in a detailed way and divided into its main stages, starting from the selection of data sources, through the stage of creating the tables of the data dictionary, until the le- advantage of the main technical and business requirements chosen for the selection of a data catalog application, capable of interpreting the metadata sources in a friendly interface for the end user. The metadata and its tables used in the project came from a technology startup
The result of this work serves as support for the understanding, through the dedicated steps, of a metadata management process, as well as its challenges and good practices.

***Keywrods**: Metadata, Data Dictionary, Data Catalog.*

1. Introdução

Existem vários conceitos para explicar o que são os metadados, mas o significado mais comum e objetivo é que eles são os dados sobre dados. Mesmo assim, essa definição pode gerar certa confusão para algumas pessoas. Muitas vezes existe um conflito se determinada informação é um dado ou um metadado.

Um exemplo didático sobre a diferença do que é um dado de um metadado é imaginar-se em uma grande biblioteca, com centenas de milhares de livros e revistas, mas sem catálogo. Sem um catálogo, os leitores podem nem saber como começar a procurar um livro específico ou mesmo um tópico específico.

O catálogo não apenas fornece as informações necessárias, mas também permite que os leitores encontrem materiais usando diferentes pontos de partida. Sem o catálogo, encontrar um livro específico seria difícil, senão impossível. Uma organização sem metadados é como uma biblioteca sem um catálogo.**[1]**

Os metadados podem ser classificados em três tipos: estratégico ou de negócio; técnico e operacional. O estratégico resume-se em definições de regras de negócio, como por exemplo:

- Modelo de dados, descrições de conjuntos de dados, tabelas e colunas;
- Origem e linhagem dos dados.

Alguns exemplos do tipo técnico:

- Detalhes dos processos de E.T.L. (*Extract, transform e load*) ou E.L.T. (*Extract, load e transform*);
- Nomes e atributos das tabelas e dos campos em bancos de dados físicos.

Por fim, alguns exemplos do tipo operacional:

- Logs de execução;
- Relatórios de requisição às consultas, frequência e tempo de execução.

Para este artigo irmos abordar os metadados dos tipos técnicos e estratégicos.

Atrelado ao entendimento, definição dos tipos e descoberta dos metadados de uma organização, surge a necessidade do gerenciamento dos dados e seus metadados.

[1] Jeferson Derenevick, jefderenevick@gmail.com.
[2]Eduardo Cunha De Almeida - DINF/UFPR, eduardo@inf.ufpr.br

Gerenciar os metadados é fundamental para o controle, manutenção, atualização e segurança dos dados, ou seja, é um processo contínuo e de extrema importância para as empresas que utilizam da cultura orientada a dados.
Dentre os benefícios de uma gestão dos metadados podemos elencar alguns, como:

- Evoluir o valor das informações estratégicas;
- Aumentar a confiabilidade dos dados;
- Bloquear o uso de dados incorretos ou desatualizados;
- Reduzir o tempo de pesquisas orientadas a dados.

1.1 Objetivo e etapas do projeto

O objetivo deste projeto é apresentar, através de etapas, um processo de gerenciamento dos metadados.

Antes de iniciar o mapeamento dos processos de gestão de metadados e para se aproximar de um caso real corporativo, foi realizada uma consulta, junto às células técnicas e de negócio de uma startup de tecnologia que recebe dados dos principais *marketplaces* brasileiros, que teve como objetivo identificar as principais necessidades e oportunidades. Como resultado, retornaram os seguintes tópicos:

- A ausência de uma gestão de metadados;
- A inexistência de uma documentação atualizada e confiável sobre os dados mais utilizados;
- Disponibilizar aos usuários uma única fonte da verdade sobre as principais métricas e indicadores da organização;
- Eliminar a duplicidade das informações provenientes de fontes de dados distintas;
- Dar visibilidade para os times técnicos e de negócio das principais tabelas, conjuntos de dados e suas descrições.

Para explicar como podemos aplicar um processo de gestão de metadados, a contextualização e exemplos serão abordados através das seguintes etapas:

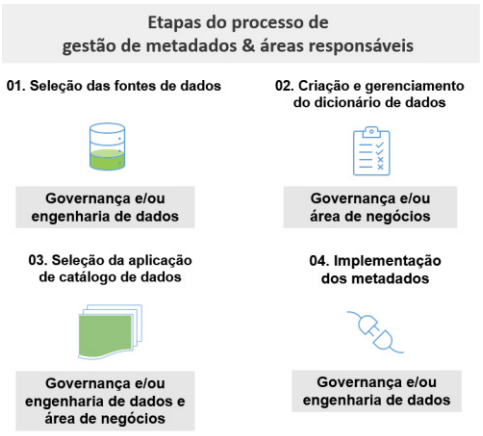

Figura 1: Resumo do projeto e etapas do processo de gestão de metadados. **01)** Iniciando com a seleção da(s) fonte(s) de dados mais consumida(s) na organização; **02)** Mapeamento, criação, atualização e manutenção do dicionário de dados; **03)** Levantamento dos requisitos técnicos e de negócio aderentes para a seleção de uma aplicação de catálogo de dados; **04)** Implementação dos metadados.

Por fim, que o desdobramento final das etapas desse processo contribua para dar suporte ao acesso das informações, permitindo que tanto as áreas técnicas quanto as áreas de negócio se tornem cada vez mais autônomas nas consultas dos metadados.

2 Seleção das fontes de dados

Para iniciar o processo de gestão dos metadados, a primeira etapa é identificar qual tabela ou fonte de dados é aderente para este fim. Desta forma, elencamos alguns pré-requisitos, estes que nortearam a seleção da fonte de dados do projeto:

- Identificar a fonte de dados com maior quantidade de acessos e consumo pelas áreas de negócio;
- Tabela(s) com as principais métricas e indicadores da organização;
- Conjunto de dados que já passaram pelo processo de E.T.L. ou E.L.T;
- Tabela(s) com os dados validados e certificados para uso em análises pelos times de negócio;
- Não possuem total ou parcialmente os descritivos dos dados ou nenhuma gestão específica dos seus metadados.

Como ressalva, é importante salientar que para este artigo não foi possível o acesso a uma aplicação ou consulta que retornasse quais tabelas eram as mais consumidas pelos usuários da organização. Dessa maneira, o levantamento foi realizado diretamente com a área de engenharia de dados. Em outros processos similares, a possibilidade de consultar o consumo dos dados nas principais tabelas, através de uma aplicação ou consulta de tabela de banco de dados, torna-o um processo mais autônomo.

Considerando todos os pontos mapeados, foram selecionadas as seguintes tabelas para fazer parte do processo de gerenciamento de metadados:

Column Name	Data Type	Length	Not Null	Description
seller_order_item_id	string	255	primary_key	
order_id	string	255	FALSE	
product_id	string	255	foreign_key	
seller_id	string	255	FALSE	
customer_id	string	255	FALSE	
payer_id	string	255	FALSE	
shipment_id	string	255	FALSE	
purchase_timestamp	timestamp	29	FALSE	
approved_at	timestamp	29	FALSE	
channel_slug	string	255	FALSE	
channel_store	string	255	FALSE	

branded_store_slug	string	255	FALSE	
region	string	255	FALSE	
quantity	bigint	19	FALSE	
price	decimal	10	FALSE	
freight_value	decimal	10	FALSE	
price_discount	decimal	12	FALSE	
price_freight_shift	string	255	FALSE	
created_at	timestamp	29	FALSE	
updated_at	timestamp	29	FALSE	
status_seller_order	string	255	FALSE	
cancelation_status	string	255	FALSE	
cancelation_reason	string	255	FALSE	
gmv	decimal	35	FALSE	
campaign_id	string	255	FALSE	
logistic_type	string	255	FALSE	
canonical_sku	string	255	FALSE	

Tabela 1: ***gold__orderitem__delta.orderitem*****.** Esta tabela contém os dados de registros de pedidos diários, de vários canais de *marketplaces*. Nesta etapa a tabela ainda não possui descritivos ou qualquer processo de gestão dos metadados:

Column Name	Data Type	Length	Not Null	Description
product_id	string	255	primary_key	
seller_id	string	255	FALSE	
canonical_sku	string	255	FALSE	
seller_product_sku	string	255	FALSE	
gtin	string	255	FALSE	
full_name	string	255	FALSE	
created_at	timestamp	29	FALSE	
updated_at	timestamp	29	FALSE	
brand	string	255	FALSE	
price_seller	decimal	10	FALSE	
offer	decimal	10	FALSE	
type_catalog	string	255	FALSE	
category_catalog	string	255	FALSE	
stock	int	10	FALSE	
product_status	string	255	FALSE	
product_approved	boolean	1	FALSE	
product_active	boolean	1	FALSE	
availability_days	int	10	FALSE	
waiting_invoice	boolean	1	FALSE	
has_rejection	string	255	FALSE	
region	string	255	FALSE	

Tabela 2: gold__product__delta.product. Esta tabela contém os dados de cadastro de produtos, categorias, subcategorias, códigos e descrições. Nesta etapa a tabela ainda não possui descritivos ou qualquer processo de gestão dos metadados:

Área(s) responsável(is) por esta etapa: Governança e/ou Engenharia de dados

Após a etapa de seleção das fontes de dados, a compreensão de alguns conceitos apresentados a seguir auxiliam no entendimento dos passos dedicados neste projeto.

3 Conceito de metamodelo

Um metamodelo é o modelo de um modelo. É uma definição precisa das regras e construções necessárias para a criação de modelos semânticos, **[2]**e que pode ser organizado por qualquer uma das seguintes perspectivas: pela origem, pela classificação ou pelo uso dos metadados.

3.1 Modelo entidade relacionamento (MER)

O Modelo Entidade Relacionamento (também chamado modelo ER, ou MER) é um modelo conceitual utilizado para descrever os objetos (entidades) envolvidos em um domínio de negócios, com suas características (atributos) e como elas se relacionam entre si (relacionamentos).**[3]** Vale mencionar que algumas organizações utilizam o diagrama de classes da UML (*Unified Modeling Language*) como uma alternativa modelo ER. Mesmo que, por um lado, o diagrama de classes tenha tido inspiração no modelo ER e consiga capturar os requisitos de dados do mundo real de uma maneira simples e significativa, produzindo um modelo inteligível.**[4]**

Na construção do MER existem algumas características que o definem em um modelo abstrato de um aspecto do mundo real, seguindo uma estrutura que representará o banco de dados:

- **Entidades:** Representam classes de objetos (abstratos ou reais) com existência independente de seus atributos, exemplo: Produto, cliente, professor, aluno etc. São representados graficamente por um retângulo;
- **Atributos:** Descrevem as propriedades ou características dentro de um tipo de entidade, exemplo: Entidade "Aluno" tem os atributos nome, idade, matrícula etc.
- **Relacionamentos:** São os acontecimentos que ligam as entidades. Dentre as características dos tipos de relacionamentos, existe o conceito de cardinalidade, que basicamente pode ser explicado como o número máximo e mínimo de ocorrências de uma entidade que estão associadas às ocorrências de outra entidade que participa do

relacionamento. Dentre os tipos de cardinalidades, podemos citar: um-para-um (1:1); um-para-muitos (1:N) e muitos-para-muitos (N:M) São representadas graficamente por um losango.

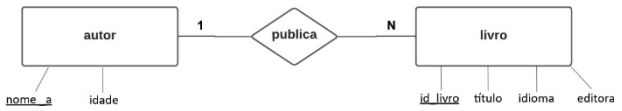

Figura 2: Exemplo de um diagrama ER, onde o "autor" (entidade) pode publicar(relacionamento) muitos livros(entidade), mas, nesse contexto apresentado, um livro pode ser publicado por apenas um autor. Observamos que os atributos chave são sublinhados para identificação.

3.1.1 Diagrama entidade relacionamento do projeto

Neste projeto utilizamos do modelo conceitual através da ferramenta diagrama entidade relacionamento. O principal papel é demonstrar as características que compõem as tabelas selecionadas, seus atributos e relacionamentos. Através dos diagramas conseguimos mapear a estrutura das entidades e características de relacionamentos que estarão presentes nas tabelas do dicionário de dados.

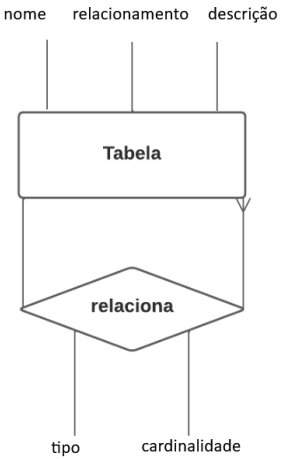

Figura 3: Mapeamento ER das da estrutura das tabelas do projeto. Na camada superior de uma tabela, ela recebe os atributos nome, relacionamento e descrição. E esta mesma entidade realiza um relacionamento que recebe os atributos tipo e cardinalidade.

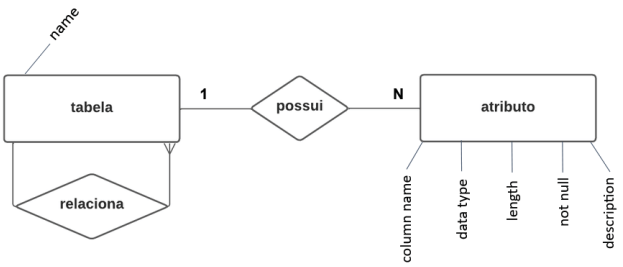

Figura 4: No mapeamento ER das entidades do projeto e seus atributos, uma tabela recebe um nome e possui um relacionamento com uma entidade atributo, em uma cardinalidade 1:N, ou seja, ela possui muitos atributos e muitos atributos pertencem a uma única tabela. Nesta etapa, conseguimos construir a estrutura que recebera os campos pertencentes a cada tabela do dicionário de dados.

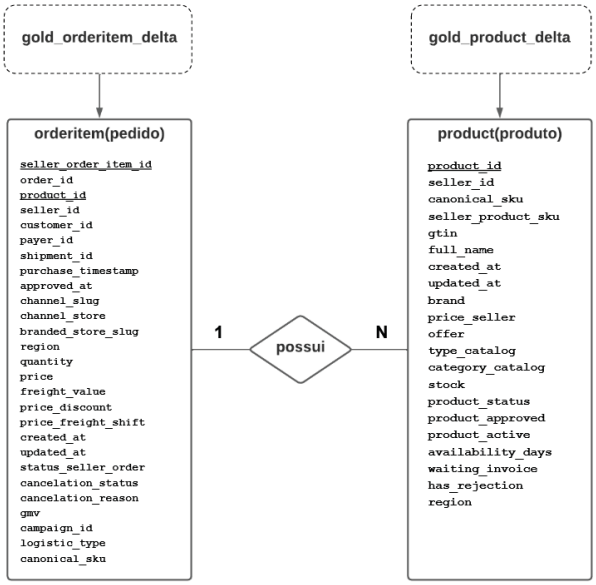

Figura 5: Mapeamento ER que descreve as entidades do projeto, seus relacionamentos e atributos. Existem dois esquemas (*gold_orderitem_delta* e *gold_produtct_delta*), ambos possuem duas entidades (tabelas): A entidade *orderitem*, possui os dados do pedido e a entidade *product* os dados de cadastro do produto. Cada entidade conta com seus atributos, sendo que os campos do tipo chave estão sublinhados para devida referência. O relacionamento entre as entidades é do tipo um para muitos (1:N), sendo que um pedido pode conter vários produtos e vários produtos podem estar vinculados a um único pedido.

3.2 Dicionário de dados (DD)

Dentre os exemplos de metamodelos que utilizam dos recursos dos metadados, nós podemos citar o dicionário de dados. Um dicionário de dados é um componente responsável pelo armazenamento dos metadados sobre a estrutura do banco de dados **[5],** considerando as informações sobre o conteúdo, atributos, estruturas e relacionamentos.

A importância de manter um dicionário de dados atualizado é crucial na gestão dos metadados de uma organização, por isso podemos considerá-lo como um repositório de metadados. É um processo que é considerado uma boa prática se a criação for realizada na fase de implementação(modelagem), mas também pode ser realizado após as tabelas estarem em produção. E normalmente as áreas que atuam na administração do banco de dados são as responsáveis pela implementação e manutenção dos dicionários de dados.

Sobre as principais características, podemos mencionar alguns dos recursos que fazem parte de um dicionário de dados:

- Informações dos nomes das tabelas e colunas do banco de dados;
- Informações das chaves primárias, relacionamento

de chave estrangeira e se existem valores nulos;
- Atributos e suas definições (Exemplo: numérico, inteiro, texto etc.)
- Informação das descrições das colunas com o intuito de detalhar os campos e as regras de negócios que foram aplicadas.

3.2.1 Dicionário de dados de tabela

Tem a função de apresentar a estrutura com as entidades, relacionamentos e descrições que fazem parte dos metadados de um banco de dados. Vamos apresentar alguns exemplos para explicar como é a estrutura do dicionário de tabelas.

nome	relacionamento	cardinalidade	tipo	descricao
carro	montadora	1:N	pertence	Cadastro dos carros da loja
carro	marca	1:N	possui	Cadastro dos carros da loja
montadora	carro	N:1	pertence	Cadastro das montadoras de carros
marca	carro	N:1	possui	Cadastro das marcas de carros

Figura 6: Exemplo de um dicionário de dados em estrutura de tabela. Através da utilização mapeamento ER ilustrado na **figura 3**, este exemplo considera 3 tabelas: carro; marca e montadora. A segunda coluna demonstra os relacionamentos entres as tabelas. A terceira coluna contempla os tipos de cardinalidade e na quarta coluna temos os tipos de relacionamentos. Na última coluna, apresentamos as descrições de cada tabela.

As características do dicionário de dados de tabela:
- **Tabela:** Nome das tabelas ou entidades do banco de dados;
- **Relacionamento:** Descreve o relacionamento entre as tabelas;
- **Cardinalidade:** Define o tipo de cardinalidade utilizada entre as tabelas;
- **Tipo:** Descreve o relacionamento utilizado entre as tabelas. Normalmente é utilizado um verbo como identificação.
- **Descrição:** Descreve, de forma textual e por extenso, a finalidade de cada tabela.

3.2.2 Dicionário de dados dos atributos de tabela

Tem a finalidade de apresentar os metadados da tabela, bem como identificar os atributos, tipos de dados, tamanho, restrições e descrições dos campos:

name	column name	data_type	length	not null	description
carro	id_carro	integer	4 bytes	pk, not null	Numero de identificacao do carro. Gerado automaticamente
carro	nome_carro	string	50 bytes	not null	Informacao do nome oficial do carro
carro	cor	string	50 bytes		Informacao da cor atual(pintura) do carro
carro	ano	integer	8 bytes		Data de fabricacao do carro, conforme registro oficial
carro	id_marca	integer	4 bytes	fk	Numero de identificacao da marca. PK da tabela marca
carro	id_montadora	integer	4 bytes	fk	Numero de identificacao da montadora. PK da tabela montadora

Figura 7: Exemplo de um dicionário de dados dos atributos de tabela ou entidade. Através da utilização mapeamento ER ilustrado na **figura 4**, identificamos os nomes das colunas, tipos de dados, comprimento ou tamanho do campo, restrições e descrições dos atributos da tabela *carro*.

As características do dicionário de dados dos atributos de tabela:
- **Tabela ou entidade:** Nome da entidade que foi definida na M.E.R;
- **Nome da coluna ou atributo:** São características da entidade;
- **Tipos de dados ou domínio:** O tipo do valor que o atributo irá receber conforme um processo lógico, exemplo: atributo ANO recebe um tipo de dados no formato de data.
- **Comprimento ou tamanho:** Define a quantidade de caracteres que serão necessários para armazenar o seu conteúdo;
- **Restrição:** Informações das colunas que são chave primária, chave estrangeira e se podem ou não receber registros vazios;
- **Descrição:** Descreve, de forma textual e por extenso, a finalidade de cada campo da tabela.

3.2.3 Dicionário de dados em um banco de dados Oracle

O dicionário de dados no SGBD Oracle é composto de tabelas base (*base table*) que são acessíveis somente a partir do usuário *SYS*, criado durante a instalação do Oracle.

As informações armazenadas no dicionário de dados incluem os nomes dos usuários do servidor Oracle, os privilégios concedidos aos usuários, os nomes dos objetos do banco de dados, as *constraints* de tabelas e as informações de auditoria. Há quatro categorias de *views* de dicionário de dados. Cada categoria possui um prefixo distinto que reflete o uso pretendido.

O prefixo do nome tem o seguinte significado:
- **DBA:** Apenas podem ser consultadas por utilizadores que tenham recebido o privilégio DBA;
- **USER:** Referem-se a objetos armazenados no *schema* do usuário que iniciou sessão na base de dados;
- **ALL:** Referem-se a todos os objetos aos quais o utilizador que iniciou sessão tem privilégios de leitura;
- **V$_:**Armazena informações sobre o desempenho ou bloqueio do servidor do banco de dados; disponível para os administradores do banco. Todas as *views* V$ são sinônimos de *views* V_$. Elas fornecem informações muito relevantes, como a *view V$SESSION*, *V$DATAFILE*, *V$INSTANCE* e outras. A *V$FIXED_TABLE* contém informações sobre todas essas *views*.

3.2.4.1 Dicionário de dados dos atributos da tabela no Oracle

Durante a operação do banco de dados, o Oracle lê o dicionário de dados para verificar se existem objetos de

esquema e se os usuários têm acesso adequado a eles. O Oracle também atualiza o dicionário de dados continuamente para refletir as mudanças nas estruturas do banco de dados, auditoria, concessões e dados.

Os acessos de leitura do DD servem como referência para todos os usuários do banco de dados e às visualizações do dicionário de dados é realizado através de comandos SQL. Algumas visualizações são acessíveis a todos os usuários e outras são destinadas apenas a administradores de banco de dados.

O dicionário de dados está sempre disponível quando o banco de dados está aberto. Para isso o *tablespace SYSTEM* deverá estar sempre online.

3.3 Gerenciamento do dicionário de dados do projeto

Para este artigo, as atualizações do DD serão realizadas através de uma planilha online com extensão em um arquivo (csv). Esta escolha foi determinada pela possibilidade de manter o dicionário de dados atualizado sem a necessidade de ficar dependente de uma arquitetura de dados específica. Além disso, os níveis de acessos dos usuários podem ser configurados conforme a definição da área responsável pela governança ou engenharia de dados.

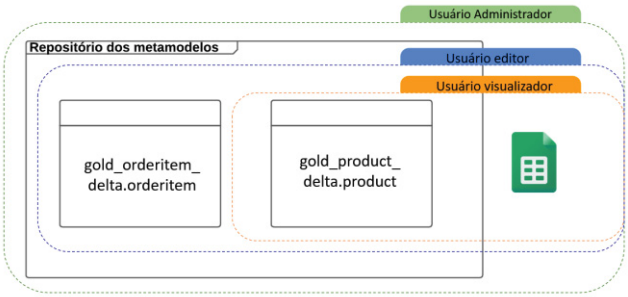

Figura 8: O repositório dos metamodelos contempla as planilhas online contendo os metadados das tabelas selecionadas para o projeto. Cada usuário, conforme seu papel dentro da gestão dos metadados, possui acesso específico do repositório e dos metadados. Estes acessos podem ser definidos como: **01) Administrador**: Perfil com acesso completo no repositório e todos os metadados. Tem o escopo de administrar e manter a estrutura dos metadados atualizada. Também é o responsável por controlar e liberar os acessos das camadas inferiores; **02) Editor:** Possui acesso a metadados definidos pelo administrador e está autorizado para realizar a edição; **03) Visualizador:** possui apenas acesso de leitura dos metadados e pode sugerir edições. Outra característica que contribui para manter o DD em formato de planilha online é a possibilidade da consulta do histórico de modificações e formas de realizar o *rollback* quando necessário.

3.3.1 Dicionário de dados das tabelas do projeto

Nesta etapa são criados os DD das entidades ou tabelas. Um dos principais objetivos é criar uma tabela que seja identificada pelos sistemas de gerenciamento de banco de dados ou por uma aplicação de catálogo de dados. Desta forma, o dicionário deve seguir o mesmo padrão proposto no **capítulo 3.2.1**, porém com os campos relativos as tabelas designadas para o projeto:

nome	relacionamento	cardinalidade	tipo	descricao
gold_orderitem_delta_orderitem	gold_product_delta_product	1:N	possui	Cadastro das informacoes diarias de pedidos
gold_product_delta_product	gold_orderitem_delta_orderitem	N:1	pertence	Cadastro dos produtos

Figura 9: Dicionário de dados em estrutura de tabela do projeto. Através da utilização mapeamento ER ilustrado na **figura 3**, este DD contempla 2 tabelas: *orderitem* e *product*. Ambas se relacionam entre si através do campo *product_id*. Na última coluna, apresentamos as descrições de cada tabela. Estas descrições são definidas, em conjunto, das áreas de negócio e técnica.

3.3.2 Dicionário de dados dos atributos das tabelas do projeto

Nesta etapa são criados os DD dos atributos das entidades ou tabelas. Através dessas informações os usuários conseguem identificar quais atributos pertencem a tabela, os tipos de dados, informações do tamanho do campo, se existe algum tipo de restrição e qual a finalidade(descrição) de cada atributo. Nesta fase a participação da área de negócios é crucial para o preenchimento das descrições dos atributos relacionados a métricas e indicadores. Quanto mais simples e objetiva for a descrição maior será o resultado de pesquisas por palavras-chave relacionadas aos dados:

name	column name	data_type	length	not null	description
gold_orderitem_delta.orderitem	seller_order_item_id	string	255 bytes	pk	Identificador do pedido do buyer relacionado ao seller_id
gold_orderitem_delta.orderitem	order_id	string	255 bytes		Identificador do pedido do buyer
gold_orderitem_delta.orderitem	product_id	string	255 bytes	fk	Identificador unico do produto
gold_orderitem_delta.orderitem	seller_id	string	255 bytes		Identificador único do seller dentro da empresa
gold_orderitem_delta.orderitem	customer_id	string	255 bytes		Identificador do buyer que realizou o pedido
gold_product_delta.product	product_id	string	255 bytes	pk	Identificador unico do produto na tabela product
gold_product_delta.product	seller_id	string	255 bytes		Identificador único do seller dentro da empresa
gold_product_delta.product	canonical_sku	string	255 bytes		Identificador único do produto no canal.
gold_product_delta.product	seller_product_sku	string	255 bytes		Identificador unico do produto que pertence a um determinado seller_id
gold_product_delta.product	gtin	string	255 bytes		Global Trade Item Number do produto. Similar ao codigo EAN

Figura 10: Dicionário de dados dos atributos de tabela do projeto. Através da utilização mapeamento ER ilustrado na **figura 4**. O preenchimento do campo *description* foi realizado através das informações passadas pelo time técnico de engenharia e o time de negócios, onde este último que definiu as descrições das principais métricas utilizadas.

Área(s) responsável(is) por esta etapa: Governança e/ou área de negócios.

4. Catálogo de dados

De forma simplificada uma aplicação de catálogo de dados é um inventário organizado de ativos de dados na

organização. Ela consome os metadados para ajudar as organizações a gerenciarem seus dados. Também auxilia os profissionais de dados a coletar, organizar, acessar e enriquecer metadados para oferecer suporte à descoberta e governança de dados.

A maioria dessas aplicações conecta-se a vários tipos de fontes de metadados, como por exemplo: planilhas, sistema de gerenciamento de banco de dados, dashboards, arquivos de texto etc.

Também podemos incluir outros recursos provenientes das informações das tabelas presentes no catálogo de dados, como: *data type*; informações da periodicidade e última atualização; quem são os *owners*; se possui dados sensíveis etc.

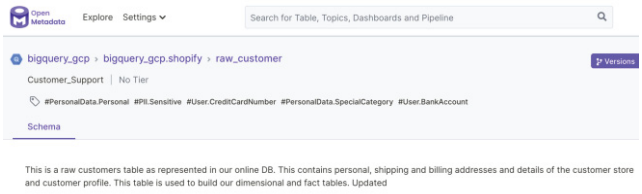

Figura 11: Exemplo da interface de uma aplicação de catálogo de dados conhecida como *open metadata*, em sua versão de junho de 2022.

4.1 Os tipos de aplicações de catálogo de dados

No momento do desenvolvimento deste artigo existem dois tipos de aplicações de catálogo de dados: de código aberto ou código proprietário.

Como exemplo de catálogos de código aberto, podemos citar: *Amundsen*, *Open metadata*, *Data hub projec*t etc. Para catálogos de código proprietário, temos: *Informatica*, *Google data catalog*, *Oracle data catalog*, *Atlan*,etc.

4.2 Como é a arquitetura de uma aplicação de catálogo de dados

Para contextualizar como uma aplicação de catálogo de dados funciona, utilizaremos o exemplo da arquitetura da aplicação de código aberto conhecida como *Amundsen*, que no momento deste trabalho estava atualizada na seguinte versão: **4.2.0, em junho de 2022.**

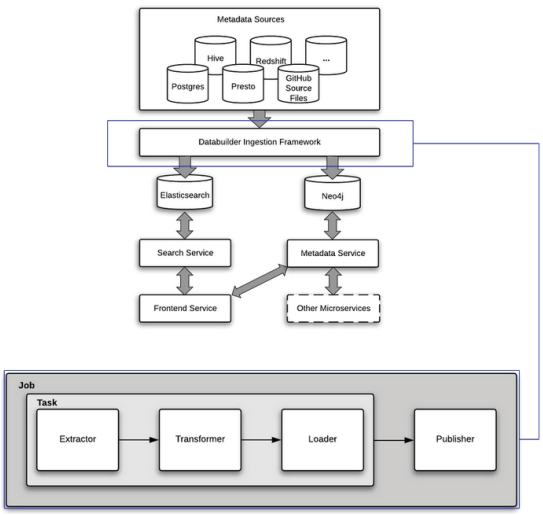

Figura 12: Ilustração da arquitetura da aplicação de catálogo de dados chamada *Amundsen*. De forma simplificada o fluxo inicia com a aplicação conectando as diversas fontes de metadados através de um framework de ingestão chamado de *databuilder*. O *databuilder* é um framework que realiza o processo de extração, transformação e carga dos dados consultados nas fontes de metadados até a fase de publicação. Após a publicação os dados transitam até estarem disponíveis em uma interface web para consulta do usuário final (*frontend service*).

[6]

4.3 Processo para seleção da aplicação de catálogo de dados do projeto

Para obter êxito nesta etapa, é importante, antes de escolher uma aplicação de catálogo de dados, .

Neste ponto devemos zelar em não selecionar requisitos subjetivos, como por exemplo: *"Necessário uma nota de performance superior a 5s em consultas em bancos relacionais"*. E se surgir a necessidade de um requisito muito específico o mesmo deve ser justificado para fazer sentido a ser elencado no levantamento.

Também existem requisitos que podem se tornar premissas para toda a sequência e levantamento da aplicação, como por exemplo a exigência que seja uma ferramenta de código aberto. Neste caso, apontar requisitos que não atendam a este perfil de aplicação se tornam desnecessários para a lista final.

Por fim, o ideal será selecionar uma aplicação que seja agnóstica a arquiteturas de dados, que seja escalável e suscetível a mudanças

4.3.1 Definindo os requisitos técnicos e de negócio

Nesta fase serão apresentados alguns exemplos de requisitos técnicos e de negócio selecionados a partir de um levantamento da companhia de e-commerce utilizada como referência para este projeto. Vale ressaltar que para a empresa selecionada para este artigo não era obrigatório que a aplicação fosse de código aberto.

Exemplos de requisitos técnicos:

- Conexão com fontes de metadados de bancos de dados em nuvem (Exemplo: Azure SQL, Amazon Athena, Google Big Query) ou locais (Exemplo: PostgreSQL);
- Conexão com metadados de ferramentas de visualização de dados (Exemplo: Power BI);
- Conexão com metadados em arquivos de planilhas;
- Conexão com metadados em arquivos de texto (csv);
- Gerenciamento dos níveis de acessos dos usuários aos metadados;
- Informações das descrições dos conjuntos de dados;
- Informações dos tamanhos dos conjuntos de dados;
- Informações das datas de atualizações dos conjuntos de dados;
- Informações sobre quais tabelas possuem dados sensíveis e quais campos registram essa informação

- Acesso via interface web;
- Customização da interface da aplicação;
- Integração nativa com o autenticador do servidor de usuário google;
- Multi-idioma ou possibilidade de tradução.

Exemplos de requisitos de negócios:

- Interface amigável e intuitiva;
- Página de pesquisa dos principais conjuntos de dados;
- Visualização das principais métricas e suas descrições;
- Visualização dos principais painéis publicados nas ferramentas de visualização de dados, suas métricas e descrições;
- Recurso de comentários sobre os metadados com o time técnico;
- Recurso de solicitação de edição de um campo ou descrição, com possibilidade de comentários com a sugestão;
- Recurso de avaliação (exemplo: por notas) de uma tabela, descrição ou visualização.

4.3.2 O recurso da matriz de aderência aos requisitos

Após o levantamento dos requisitos técnicos e de negócios precisamos avaliar quais aplicações de catálogo de dados possuem maior ou menor compatibilidade com a necessidade da organização.

Para essa etapa, podemos utilizar um recurso chamado matriz de aderência. Ela facilita o processo de decisão porque avalia qual aplicação é mais aderente aos requisitos informados. Como exemplo, vamos selecionar alguns dos requisitos técnicos com o intuito de demonstrar como a matriz é utilizada:

	Aderência > > >	100%	70%	50%	
	Requisitos	Aplicação 1	Aplicação 2	Aplicação 3	Descrição Detalhada Requisito
REQUISITOS TÉCNICOS					
1	Informações dos tamanhos dos conjuntos de dados	Sim	Sim	Sim	Descrição da quantidade de registros e tamanho do arquivo do conjunto de dados ou tabelas
2	Conexão com metadados de ferramentas de visualização de dados	Sim	Parcialmente	Não	Preferencialmente na ferramenta Microsoft Power BI
3	Informações sobre quais tabelas possuem dados sensíveis e quais campos registram essa informação	Sim	Sim	Parcialmente	Possibilidade de identificar quais tabelas possuem dados sensiveis, em quais campos e que área é responsavel por administrar esses dados
4	Integração nativa com o autenticador do servidor de usuário google	Sim	Não	Parcialmente	Autenticação nativa e segura através do usuário e senha google.
5	Conexão com metadados em arquivos de extensão .csv	Sim	Sim	Parcialmente	Nossos principais metadados serão gerenciados e atualizados em arquivos .csv

Figura 13: A matriz é construída através da descrição dos requisitos em suas linhas. Podemos listar tantos os técnicos quantos os de negócio na mesma matriz ou separar pelo tipo. Nas colunas é incluída a identificação das aplicações selecionadas. Na última coluna podemos incluir o campo responsável pelo detalhamento de cada requisito. No topo da matriz temos a informação percentual da aderência, que é referenciada em uma escala de 0% a 100%. Em cada requisito o usuário seleciona uma opção, esta que é definida considerando o nível de satisfação de atingimento em cada requisito e que será contabilizado através de uma nota: sim (1); parcialmente (0,5) e não (0). Após a seleção de todas as opções, as notas são somadas e divididas pelo total de requisitos, exemplo: 5 requisitos no total, todos selecionados como sim (1), então serão 5 requisitos x 1, que resulta em 5, que será divido pelo total de requisitos (5), dessa forma teremos 100% de aderência.

Área(s) responsável(is) por esta etapa: Governança e/ou engenharia de dados e área de negócios.

5. Processo de implementação dos metadados

A última etapa consiste na implementação das tabelas criadas para os dicionários de dados em uma aplicação de catálogo de dados.

Para esta etapa ser eficiente, precisamos garantir que as tabelas criadas sigam as estruturas apresentadas nas **figuras 9 e 10** e que também estejam no formato de extensão de arquivo de texto (csv). A partir desses requisitos, vamos demonstrar, de forma abstrata e simplificada, um exemplo do fluxo de consulta, leitura, carregamento dos metadados:

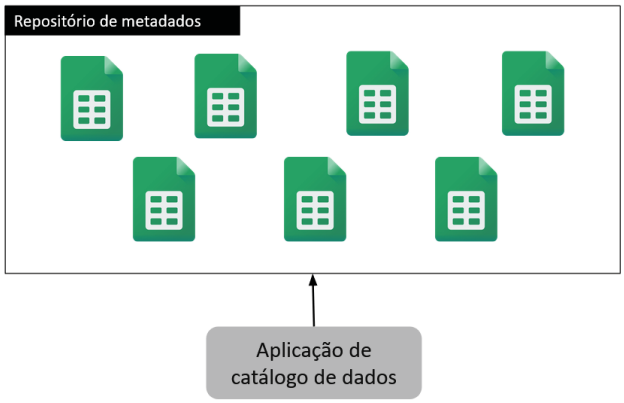

Figura 14: Inicialmente a aplicação se conecta no repositório dos metadados, onde se encontram os arquivos do tipo texto(csv), com as estruturas dos dicionários de dados de tabelas e de atributos de tabelas.

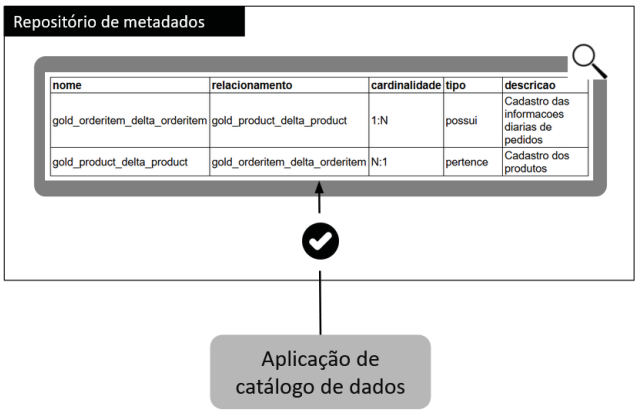

Figura 15: Após a conexão na fonte de metadados, a aplicação faz a leitura das tabelas de origem, seus relacionamentos, atributos de tabelas, descrições ,etc. Como premissa, as tabelas devem seguir o modelo de estrutura aplicada nas **figuras 9 e 10** deste artigo.

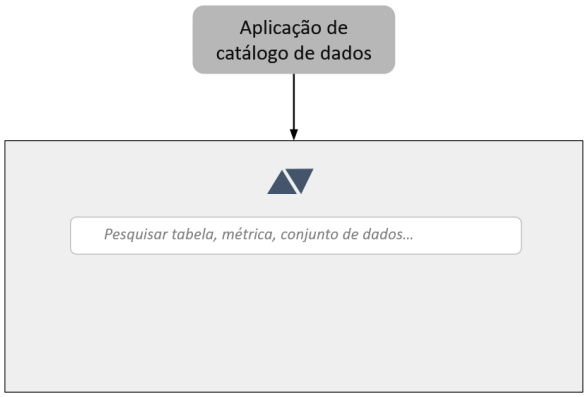

Figura 16: Por fim, a fonte contendo o metadado estará disponível para a aplicação executar seus recursos de tratamento e gestão do dado. A partir dessa etapa a gestão do metadado pode ser realizada pelo usuário através da própria aplicação de catálogo de dados.

Área(s) responsável(is) por esta etapa: Governança e/ou engenharia de dados.

6. Conclusão

A contextualização das etapas dedicadas neste projeto foram fundamentais para fornecer um documento com o objetivo de especificar todos os passos que podem ser aplicados em um processo para a gestão de metadados. Desta forma, os capítulos foram preparados com a finalidade de explicar cada fase do ciclo de vida dos dados, ou seja, iniciando na seleção das fontes de dados e seus metadados, até o levantamento dos principais requisitos técnicos e de negócios escolhidos para a seleção de uma aplicação capaz de interpretar e catalogar os dados em uma interface amigável para o usuário final.

Como ponto de destaque, identificamos que a utilização do modelo entidade relacionamento (MER) contribui para o entendimento e a construção das tabelas que farão parte do dicionário de dados. Além de ser agnóstico a um banco de dado específico.

Outro ponto relevante, foi o envolvimento das áreas técnicas e de negócio na fase de levantamento de requisitos dedicados à seleção da aplicação de catálogo de dados. Desta maneira, conseguimos alinhar todas as expectativas relacionadas a esta escolha.

A descrição mais técnica na etapa de implementação da aplicação de catálogo de dados não foi possível devido a uma mudança de arquitetura de dados da organização utilizada no projeto no momento do desenvolvimento deste artigo.

Agradecimentos

Agradeço ao professor Eduardo C. De Almeida por todo o suporte e orientação no projeto final e a todos os professores da especialização. Também agradeço a minha esposa Angélica, minha filha Gabi e a minha família por toda a paciência e apoio durante o período da especialização e do desenvolvimento do projeto final.

Referências

[1] Escola nacional de administração pública – ENAP, *Gerenciamento de metadados e da qualidade de dados*, (2019), <https://repositorio.enap.gov.br/bitstream/1/5008/4/M%C3%B3dulo%204%20-%20Gerenciamento%20de%20Metadados%20e%20da%20qualidade%20de%20Dados.pdf>

[2] A. Tannenbaum, *Using Metamodels, Repositories, XML,and Enterprise Portals to Generate Information on Demand,* Ed. Addison Wesley,(2002), ISBN-13 9780201719765

[3] A. Silberschatz, H. Korth, *Database System Concepts, 6th Edition*, (2011), ISBN 9780073523323

[4] Franck, M. Kewry, Pereira, Robson Fernandes & Filho, Jerônimo Vieira Dantas, *Ratio-Entity Diagram: a tool for conceptual data modeling in Software Engineering*, (2021), <https://rsdjournal.org/index.php/rsd/article/download/17776/15626/221575>

[5] Rogério G. Bittencourt, *Aspectos básicos de bancos de dados*, (2004), <https://www.marilia.unesp.br/Home/Instituicao/Docentes/EdbertoFerneda/BD%20-%20Aspectos%20Basicos.pdf>

[6] Tao Feng, *Open Sourcing Amundsen: A Data Discovery And Metadata Platform,* (2019), < https://eng.lyft.com/open-sourcing-amundsen-a-data-discovery-and-metadata-platform-2282bb436234 >