NOVA IMS Information Management School

MGI

Mestrado em Gestão de Informação
Master Program in Information Management

A utilização de dados públicos abertos na construção de um Data Warehouse: A construção de um repositório estatísticas educacionais públicas brasileiras.

Luiz Marcelo Ferreira Carvano

Trabalho de Projeto apresentada(o) como requisito parcial para obtenção do grau de Mestre em Gestão de Informação

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

2018

A utilização de dados públicos abertos na construção de um Data Warehouse: A construção de um repositório estatísticas educacionais

Luiz Marcelo Ferreira Carvano

MGI

NOVA Information Management School
Instituto Superior de Estatística e Gestão de Informação
Universidade Nova de Lisboa

A utilização de dados públicos abertos na construção de um Data Warehouse: A construção de um repositório estatísticas educacionais públicas brasileiras.

por

Luiz Marcelo Ferreira Carvano

Trabalho de Projeto apresentada(o) como requisito parcial para a obtenção do grau de Mestre em Gestão de Informação.

Orientador: Roberto André Pereira Henriques

Fevereiro 2018

DEDICATÓRIA

Dedico esse trabalho aos meus pais (Jorge e Delta) que sempre me apoiaram em todas as minhas decisões. Sem o amor e carinho deles eu nunca conseguiria chegar até aqui.

Às minhas tias (Dilva e Marina) que sempre rezaram pelo meu sucesso.

Aos meus irmãos (Lourenço e Márcio) que foram uma inspiração constante ao longo da vida.

E por último, mas não menos importante, para minha esposa Patricia, que sempre esteve ao meu lado durante essa jornada a aconselhar e sugerir os melhores caminhos.

Para todos o meu Amor…

AGRADECIMENTOS

Gostaria de agradecer à Universidade Nova de Lisboa e ao curso Gestão de Informação, em especial aos seus docentes e funcionários, por terem possibilitado uma experiência de aprendizado única e inesquecível.

Ao professor Roberto André Pereira Henriques por ter tido a gentileza, dedicação e paciência na orientação desse trabalho.

À equipe do Instituto Brasileiro de Análises Sociais e Econômicas – IBASE, em especial à Rita Corrêa Brandão e a Cândido Gryzbowski, que me deram a oportunidade para desenvolver o presente projeto.

Ao professor Paulo de Martino Jannuzzi, pelos conselhos e o incentivo na finalização desse trabalho.

Por último, a um incontável número de amigos e amigas que estiveram presentes ao longo da minha vida profissional e acadêmica.

RESUMO

Na última década, diferentes países têm desenvolvido iniciativas relacionadas à divulgação de dados governamentais de forma aberta. Apesar da existência e disponibilização das bases de dados, a tarefa de utilização e extração de conhecimento dessas bases ainda apresenta alguns desafios, relacionados a à integração e à compatibilização das informações. Isso ocorre devido à baixa estruturação e a grande heterogeneidade das fontes, que faz com que as abordagens tradicionais de extração transformação e carga (ETL) tornem-se menos eficientes.

Esse trabalho busca analisar uma abordagem de construção de um repositório de dados abertos baseada na estrutura dos arquivos unidimensionais (flat files), que possibilite a construção dos modelos dimensionais de forma mais eficiente.

PALAVRAS-CHAVE

Dados Públicos Abertos, Ciências Sociais, Data Warehouse Design, ETL

ABSTRACT

In the last decade, different countries have developed initiatives related to the dissemination of open data. Despite the existence and availability of databases, the task of using this data and knowledge extraction still presents some challenges related to the integration and compatibility of information. This occurs due to both poor-structure and a great heterogeneity of sources, which make traditional extraction, transformation, and loading (ETL) approach less efficient.

This manuscript analyzes an approach for the construction of open data repository based on a flat files structure that enables a more efficient dimensional model building.

KEYWORDS

Open Public Data, Social Science, Data Warehouse Design, ETL

ÍNDICE

ÍNDICE DE FIGURAS

ÍNDICE DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

CGU	Controladoria Geral da União
CIGA	Comitê interministerial para Governo Aberto
Comperj	Complexo Petroquímico do Estado do Rio de Janeiro
DAC	Assistência ao Desenvolvimento
DATASUS	Departamento Nacional de Informação e Informática em Saúde
DDI	Data Documentation Initiative
DSS	Decision Support Systems
DW	Data Warehousing
EBT	Escala Brasil Transparente
EFA	Educação para Todos

e-PING	Padrões de Interoperabilidade de Governo Eletrônico
ePSI	PSI Scoreboard
ETL	Extract Transform Load
GEM	Relatório Global de Monitoramento da Educação
GT	Grupo de Trabalho
IBASE	Instituto Brasileiro de Análises Sociais e Econômicas
IBGE	Instituto Brasileiro de Geografia e Estatística
ICPSR	Inter-university Consortium for Political and Social Research
ICT	Information and Communications Technology
IES	Instituições de Educação Superior
INCID	Indicadores da Cidadania
INDA	Infraestrutura Nacional de Dados Abertos
INEP	Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira
KPI	Key Performance Indicator
LAI	Lei de Acesso à Informação
MTE	Ministério do Trabalho e Emprego
OD	Open Data
ODB	Open Data Barometer
ODE	Capgemini Consulting's Open Data Economy
ODI	Open Knowledge Foundation's Open Data Index
ODRA	World Bank's Open Data Readiness Assessment,
OECD	Organisation for Economic Co-operation and Development
OGD	Open Government Data
OGP	Open Government Partnership
OLAP	Online Analytical Processing
OLTP	Online Transaction Processing
ONG	Organização Não Governamental
PDA	Plano de Dados Abertos
PISA	Programa Internacional de Avaliação de Estudantes
PL	Projeto de Lei
PNAD	Pesquisa Nacional por Amostra de Domicílios
SDG	Objetivos de Desenvolvimento Sustentável
Seec	Serviço de Estatística da Educação e Cultura
SIAFI	Sistema Integrado de Administração Financeira do Governo Federal
SOD	Statistical Open Data
UIS	Instituto de Estatística da Unesco
UNESCO	Organização das Nações Unidas para a Educação, a Ciência e a Cultura
WB	Banco Mundial

1. INTRODUÇÃO

No Brasil, desde de meados dos anos de 1990, os grandes produtores de estatísticas públicas passaram a disponibilizar paulatinamente suas bases de dados de forma aberta - inicialmente em mídias digitais e posteriormente na internet (Gracioso, 2003; Senra, 2009). Associado a esse fato, o avanço da microinformática possibilitou que o processamento dessas bases de dados, anteriormente restrito aos ambientes de grande porte (mainframes), passasse a ser uma realidade para um número maior de instituições de diferentes áreas de atuação (Silva, 2003).

Apesar da existência e disponibilização das bases de dados, a tarefa de utilização e extração de conhecimento dessas bases apresenta desafios técnicos e metodológicos para a sua utilização efetiva. Mesmo com essas dificuldades, a relevância das informações que esses dados podem propiciar faz com que a utilização dos mesmos represente um poderoso subsídio para a elaboração de pesquisas acadêmicas e o aprimoramento das políticas públicas (Silva, 2003).

Nesse sentido, a utilização de técnicas de Data Warehousing pode otimizar o uso dessas bases de dados, principalmente por possibilitar que as informações estejam disponíveis de forma consistente e imediata para um sempre heterogêneo conjunto de usuários (Inmon, 2005).

O presente trabalho é uma tentativa de auxiliar o Instituto Brasileiro de Análises Sociais e Econômicas (Ibase) na construção de uma solução de Data Warehousing dentro do projeto do *Sistema de Indicadores de Cidadania*[1] para a Região Metropolitana do Rio de Janeiro, como parte dos trabalhos da Câmara Metropolitana de Integração Governamental para definição de um Plano Estratégico Urbano Integrado para toda a região.

O Ibase é uma organização não governamental (ONG) que surge em 1981 a partir dos esforços do sociólogo Hebert de Souza, em conjunto com Carlos Afonso e Marcos Arruda, quando estes retornam ao Brasil, após o exílio imposto a eles pela a ditatura militar que governou o país entre os anos de 1964 a 1985.

Com uma história ligada ao processo de redemocratização do Brasil, em especial às lutas que "permeiam a emergência da cidadania e a constituição da diversificada sociedade civil brasileira"[2] o Ibase tem atuado nesses últimos 36 anos para promover o debate entre os diferentes atores da sociedade civil e os entes públicos, por meio da utilização de diferentes ferramentas que vão das análises de dados socioeconômicos até o incentivo para a construção de fóruns coletivos de atuação.

Nesse sentido, foi desenvolvido em 2011 o *Sistema de Indicadores de Cidadania*. Este é baseado no conceito de "Cidadania Efetiva" que pode ser traduzido como sendo mais que a simples soma de direitos individuais e segmentados. Para o Ibase, a cidadania é efetivamente construída pelas pessoas e suas ações ao se organizarem coletivamente.

[1] O Sistema de Indicadores de Cidadania foi originalmente desenvolvido para monitorar estado da cidadania em 14 municípios da Área de Influência do COMPERJ. http://incid.org.br/
[2] http://ibase.br/pt/sobre-o-ibase

Para tentar mensurar a Efetividade da Cidadania, o *Sistema de Indicadores de Cidadania* organiza os Indicadores em quatro dimensões interdependentes e em três grandes conjuntos de direitos fundamentais. A figura abaixo mostra o conjunto de dimensões e de direitos de cidadania em sua interação e interdependência.

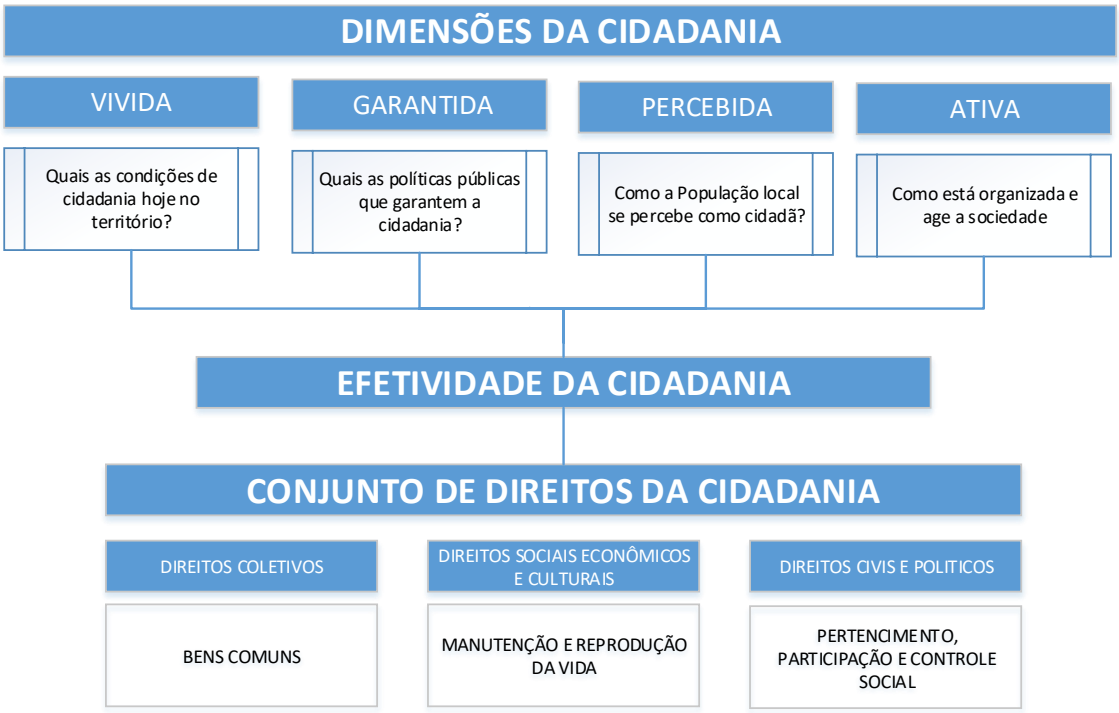

Figura 1 – Diagrama de dimensões e de direitos de cidadania[3]

Por se tratar de uma taxinomia que abrange todas as dimensões da vida humana, vamos trabalhar, para fins de exemplificação, somente os indicadores da dimensão da cidadania garantida que diz respeito aos direitos sociais, econômicos e culturais relacionados com o acesso à educação básica e superior.

Segundo o IBASE, a "Cidadania garantida (pelo Estado) diz respeito às condições criadas para o usufruto de direitos (acesso e disponibilidade). Corresponde à disponibilidade e às condições de acesso aos direitos de cidadania que devem ser garantidos pelo Estado como um todo, pelas políticas públicas, pelo judiciário e pelas instituições nas diferentes situações dadas".

[3] http://incid.org.br/sistema-de-indicadores/

1.1. OBJETIVOS GERAIS

O objetivo principal do presente projeto é construir uma solução de Data Warehousing que possibilite ao IBASE ter autonomia no processo de produção de indicadores sociais a partir de dados públicos abertos.

Essa solução deverá ser pensada para atender às características de uma instituição do terceiro setor, adequando-se ao perfil dos utilizadores internos e externos e as características da infraestrutura tecnológica já existente.

Inicialmente o projeto será um piloto que abrange apenas a área dos 92 municípios do Estado do Rio de Janeiro, em especial os 21 que compõem a sua Região Metropolitana[4].

Além disso, nesse primeiro momento, serão utilizadas informações do sistema estatístico nacional relativas à temática educacional.

Deve-se ressaltar que o presente projeto não está focado, nesse momento, no consumo das informações, pois já existe uma ferramenta produzida pela instituição para tal propósito.

1.2. OBJETIVOS ESPECÍFICOS

Abaixo, as etapas para realização eficaz desse projeto são:

- Identificar junto aos utilizadores internos a Instituição as necessidades e características desejáveis para a presente solução;
- Identificar o perfil dos utilizadores, externos a Instituição, em relação as limitações e necessidades;
- Identificar as infraestruturas tecnológicas (Hardware e Softwares) atualmente disponíveis na Instituição;
- Identificar as características dos indicadores utilizados pela Instituição;
- Identificar as características das fontes de dados públicas utilizadas pela Instituição;
- Desenvolver uma solução de ETL que possibilite aos utilizadores da Instituição a criação dos indicadores sociais.
- Construir os modelos de dados (Sistemas Transacionais e Data Warehouse), dependendo da definição do modelo que será adotado para atender as necessidades da Instituição;

[4] O conceito de Região Metropolitana surge no início da década de 1970 como parte de uma política nacional de desenvolvimento urbano e industrial, que visava a criação de unidades de planejamento regional com a finalidade compartilhar soluções para problemas comuns às grandes concentrações populacionais (Moura & Barion, 2006).

O presente trabalho está organizado da seguinte forma: O primeiro capítulo apresenta a introdução do problema e os objetivos esperados. O segundo capítulo apresenta a discussão sobre o movimento de dados públicos abertos e sua situação no Brasil. O terceiro capitulo introduz o conceito de Data Warehousing e as especificidades desse no contexto das ciências sociais. O quarto capítulo trata da construção do modelo de dados e das soluções de ETL utilizadas. O quinto e último capítulo apresenta as conclusões alcançadas e as limitações do presente trabalho.

2. DADOS PÚBLICOS E ABERTOS

2.1. DADOS GOVERNAMENTAIS ABERTOS

Segundo diferentes autores (Alexopoulos, Zuiderwijk, Charapabidis, Loukis, & Janssen, 2014; Gonzalez-Zapata & Heeks, 2014; Ubaldi, 2013) os governos têm a primazia na produção, utilização e divulgação de dados sobre os cidadãos, as organizações e a prestação de serviços públicos.

Essa formidável fonte de informações representa um valioso insumo para a criação de novos negócios e oportunidades, para a promoção da transparência e da accountability, bem como para o aumento da participação ativa dos cidadãos nos processos tomada de decisão (Attard, Orlandi, Scerri, & Auer, 2015; Susha, Zuiderwijk, Janssen, Ke, & Nlund, 2015).

Para que isso ocorra, esses dados devem ser disponibilizados de forma aberta, isto é, publicados num formato não proprietário e independentes de plataforma, legível por máquina e disponibilizados ao público, sem restrições a reutilização das informações. Foi dessa forma que, em dezembro de 2007, os trinta participantes do *Open Government Working Group* reunidos em Sebastopol (Califórnia-US), definiram os princípios necessários a divulgação de dados governamentais de forma aberta[5]. Os *Oito Princípios dos Dados Governamentais Abertos*, como ficaram conhecidos, se apresentam como um conjunto de diretrizes que tem servido, em alguma medida, como base para a maior parte das iniciativas de dados governamentais realizadas (Attard et al., 2015; Ubaldi, 2013). Desde então, dados assim divulgados podem assumir diferentes terminologias, tais como: “Open Government Data”, “Public Government Data”, “Public Open Data”, “Statistical Open Data” ou simplesmente “Open Data”.

Segundo Gonzalez e Heeks, (2014), com base no artigo de Howard[6], os Dados Governamentais Abertos (OGD, da sigla em inglês) seriam o resultado da intercessão de três conceitos: “Dados Governamentais”, “Dados Abertos” e “Governo Aberto”.

O conceito de Dados Governamentais se baseia, como foi dito acima, na elevada capacidade que os governos têm na coleta e armazenamento de informações de diferentes áreas de interesse, tais como demografia, trabalho e emprego, educação e formação, entre outras.

A possibilidade da existência de Dados Abertos advém das inovações no processamento e na difusão relacionados com a área de tecnologia da informação e da comunicação (ICT), que possibilitaram uma maior utilização dos dados, de forma muito mais rápida e eficiente que até então.

Por último, o conceito de Governo Aberto tem como origem a convicção de que a tomada de decisões e ações governamentais devem ser mais transparentes e participativas para os cidadãos (Gonzalez-Zapata & Heeks, 2014; Meijer, Curtin, & Hillebrandt, 2012) . Nesse sentido, o memorando presidencial do recém-eleito Barack Obama, que tratava da questão da Transparência e do Governo

[5] Para maiores detalhes, ver o sítio https://opengovdata.org/

[6] “Data for the Public Good”. Sebastopol, CA: O’Reilly Media - http://radar.oreilly.com/2012/02/data-public-good.html

Aberto[7], representa um marco na difusão do conceito (Dawes, Vidiasova, & Parkhimovich, 2016; Meijer et al., 2012; Susha et al., 2015)

Conjuntamente a esses três conceitos, Gonzalez e Heeks, (2014) identificam, a partir de uma análise da literatura existente, quatro diferentes perspetivas como elemento de união do conceito de OGD: a Burocrática, a Tecnológica, a Política e a Econômica.

Enquanto as três primeiras perspetivas, apontadas pelos autores, estariam ligadas respetivamente aos conceitos de "Dados Governamentais", "Dados Abertos" e "Governo Aberto", a quarta emergiria como uma consequência do resultado da intercessão das três perspetivas supracitadas ("Dados Governamentais Abertos"), já que esse teria o potencial para agregar valor econômico, a partir das inovações produzidas com base na utilização dos dados (Berro, Megdiche, & Teste, 2015).

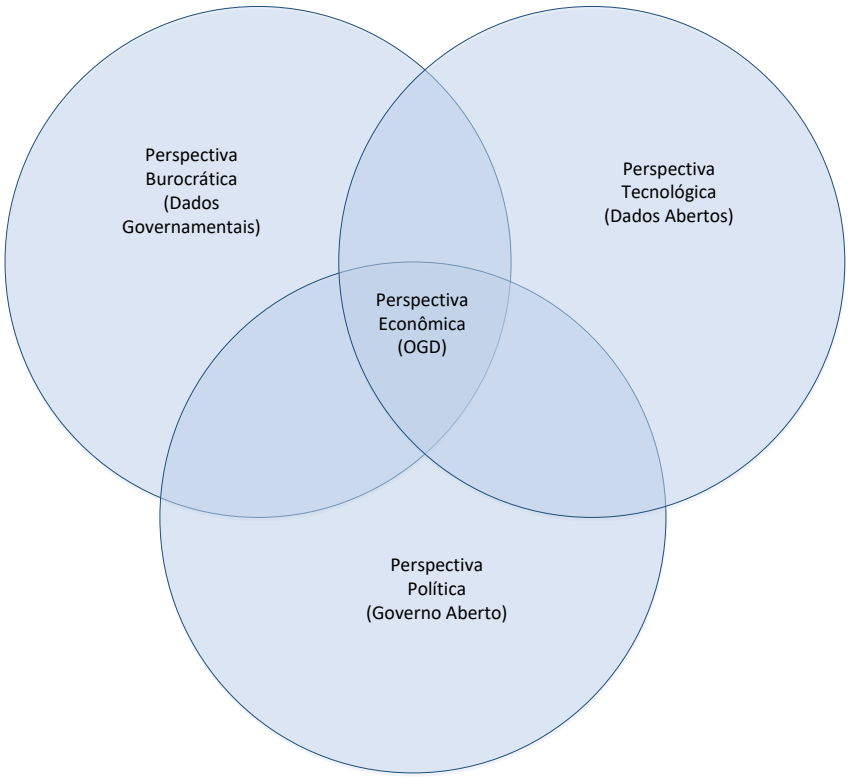

Figura 2 - Diagrama com o relacionamento entre conceitos e perspectivas. Baseada no Diagrama de Gonzalez-Zapata & Heeks.

Ao analisar conjuntamente as quatro perspetivas sugeridas por Gonzalez e Heeks e as três principais razões para a existência das iniciativas de dados governamentais abertos – que consistem em transparência, promover inovação que agregue valor econômico e a participação ativa dos cidadãos, apontadas por Attard et al. (2015) – é possível identificar uma interconectividade entre estes, já que as razões para a adoção de uma política de dados aberto ao redor do mundo, tais como o combate à corrupção, permaneceram constantes ao longo da última década (Alexopoulos et al., 2014; Attard et al., 2015; Bertot, Jaeger, & Grimes, 2010; Dos Santos Brito, Costa, Garcia, &

[7] https://www.whitehouse.gov/the_press_office/TransparencyandOpenGovernment

De Lemos Meira, 2014; dos Santos Brito, da Silva Costa, Garcia, & de Lemos Meira, 2014; McHugh, 2013; Meijer et al., 2012; Zuiderwijk & Janssen, 2014).

Mesmo que não haja um consenso entre as definições dos conceitos e seu relacionamento (Weinstein & Goldstein, 2012), os autores sugerem que a "parceria" entre os conceitos pode proporcionar o fortalecimento deles enquanto movimento, já que traria ganhos para todos.

É nesse cenário que podemos identificar, a partir de 2009, que a temática OGD tem ganhado espaço nas agendas de diversos países (Susha et al., 2015) de forma gradual, mas constante.

Segundo os autores (Shen, Riaz, Palle, Jin, & Peña-Mora, 2015), a partir da análise da edição de 2014 do *Open Data Barometer* (ODB)[8], foi possível identificar 86 países analisados para a construção do ranking. Na versão da publicação de 2015, esse número havia subido para 92 países[9].

Diversos países têm se destacado na difusão de suas bases de dados em formato aberto. Podemos apontar o caso do Estados Unidos (data.gov) e do Reino Unido (data.gov.uk), que têm desenvolvido uma política de disponibilização de suas bases em portais especificamente criados com esse objetivo (Susha et al., 2015). Nesses portais é possível encontrar uma grande variedade de bases de dados (aproximadamente 193.976 no data.gov e 40.736 no data.gov.uk)[10] nas mais diferentes temáticas, tais como saúde, meio ambiente, gastos governamentais e econômicos.

Independentemente da quantidade de bases de dados disponibilizadas, atualmente é possível aceder dados governamentais de uma grande quantidade de países de forma facilitada. Assim, é possível obter informações sobre países como Canadá (open.canada.ca/data), Nova Zelândia (data.govt.nz), Portugal (dados.gov.pt), Brasil (dados.gov.br), Grécia (data.gov.gr) ou até mesmo Nepal (data.opennepal.net)[11], dentre muitos outros.

Não só o número de países analisados tem aumentado ao longo do tempo, visto que o número de *benchmarks* voltados para a temática dos Dados Abertos, em especial os dados governamentais, também tem aumentado ao longo desse período.

Segundo *Shen et al* (2015) e *Susha et al*.(2015), foi possível listar várias iniciativas, tais como: *PSI Scoreboard* (ePSI); *World Bank's Open Data Readiness Assessment* (ODRA); *Open Knowledge Foundation's Open Data Index* (ODI) e *Capgemini Consulting's Open Data Economy* (ODE), além do Open Data Barometer (ODB), o que reforça o interesse que o movimento do OGD tem despertado[12].

Com diferentes metodologias e objetivos, esses *benchmarks* buscam avaliar critérios que passam pelo nível de observâncias dos *Oito Princípios*, Impacto social, econômico e político das iniciativas até o grau de implementação ou maturidade das políticas de dados abertos, por exemplo. A pesar das diferenças, os *benchmarks* têm em comum a capacidade de avaliar o progresso da adoção de dados abertos em vários países, ao identificar os pontos fracos e fortes dessas iniciativas com a finalidade de ajudar no desenvolvimento do processos de aperfeiçoamento (Susha et al., 2015).

[8] http://opendatabarometer.org/2ndEdition/
[9] http://opendatabarometer.org/3rdEdition/report/#executive_summary
[10] Os portais foram visitados em 24 de janeiro de 2017.
[11] Idem 5.
[12] Além dessas iniciativas citadas acima, ainda é possível destacar o *benchmark* da *Organisation for Economic Co-operation and Development* (OECD).

Apesar dos avanços obtidos no período recentes, a divulgação dos dados governamentais de forma aberta ainda enfrenta vários desafios. O fato de os dados estarem disponíveis para download não garante o aumento da transparência ou controle social que esses deveriam proporcionar (Heise & Naumann, 2012; McHugh, 2013). Problemas relacionados a questões tecnológicas, como a utilização de diferentes formatos na divulgação das bases (Attard et al., 2015; Dos Santos Brito et al., 2014), a qualidade dos metadados ou metainformações disponibilizados (Attard et al., 2015; B et al., 2015; Dawes et al., 2016; McHugh, 2013) ou até mudanças dos paradigmas político administrativos referentes à questão da transparência e dados abertos[13] representam desafios que devem ser considerados quando se trata da utilização dos OGD na produção de sistemas de informação.

2.2. DADOS GOVERNAMENTAIS ABERTOS NO BRASIL

O Estado brasileiro tem trilhado um longo caminho na divulgação das informações públicas produzidas pelos governos e suas agências.

Desde a promulgação da Constituição Federal de 1988, várias medidas têm sido tomadas para garantir ao cidadão o acesso a um variado rol de informações públicas (Gruman, 2012; OGP, 2011; Polo, 2015). O artigo 5º, inciso XXXIII, da Constituição Federal, que afirma que "todos têm direito a receber dos órgãos públicos informações de seu interesse particular, ou de interesse coletivo ou geral, que serão prestadas no prazo da lei, sob pena de responsabilidade, ressalvadas aquelas cujo sigilo seja imprescindível à segurança da sociedade e do Estado", abre caminho legal para as iniciativas realizadas nos anos seguintes.

Na década de 1990, ainda em relação à legislação, podemos apontar dois marcos importantes para o acesso à informação pública (Polo, 2015). O primeiro é a adoção da Lei nº 8.159/1991, a chamada "Lei dos Arquivos", que define as regras básicas para a gestão documental, a organização dos serviços arquivísticos nacionais e o acesso aos documentos nestes guardados. O segundo marco seria a Lei nº 9.507/1997, conhecida como "Lei do Habeas Data", que regulariza o inciso LXXII do art. 5º, da Constituição Federal, ao assegurar o acesso às informações sob a forma de "registros ou bancos de dados de entidades governamentais ou de caráter público".

No campo tecnológico, o avanço da microinformática possibilitou que diferentes órgãos e agências da administração pública, ligados ao chamado Sistema Estatístico Nacional, passassem a disponibilizar paulatinamente suas bases de dados em meios digitais (CD-ROM e posteriormente na Internet) para um grupo maior de usuários especialistas (L. de S. G. Gracioso, 2004; Senra, 2009; Silva, 2003). Dentre as iniciativas que merecem destaque, podemos apontar as realizadas pelo Instituto Brasileiro de Geografia e Estatística (IBGE), pelo Ministério do Trabalho e Emprego (MTE) e Departamento Nacional de Informação e Informática em Saúde (DATASUS)[14]. Essas iniciativas

[13] Durante a redação do presente trabalho, a recém-eleita administração do Presidente Donald Trump ordenou à Agência de Proteção Ambiental (Environmental Protection Agency – EPA) que excluísse de sua webpage informações e dados referentes às mudanças climáticas. Fonte: http://www.reuters.com/article/us-usa-trump-epa-climatechange-idUSKBN15906G.

[14] A partir de 2008 o DATASUS passou a se chamar Departamento Nacional de Informática em Saúde.

possibilitaram o acesso a dados sociodemográficos, de mercado de trabalho e de mortalidade, restritos anteriormente aos ambientes de grande porte (mainframes).

A década de 2000 representou a consolidação de um conjunto de ferramentas, desenvolvidas no final da década anterior, relacionadas ao fortalecimento do governo eletrônico (E-gov) dentro da administração federal (Jardim, 2004). Tendo sido beneficiadas pelo sucesso no enfrentamento do "bug do milênio", as áreas de ICT do governo federal aproveitaram-se da restruturação de seu parque tecnológico (Diniz, Barbosa, Junqueira, & Prado, 2009) para aperfeiçoar a utilização dessas ferramentas dentro da administração pública. Dentre as várias iniciativas, podemos destacar o Sistema Integrado de Administração Financeira do Governo Federal (SIAFI) destinado ao registro, controle e acompanhamento da execução orçamentária, financeira e patrimonial do Governo Federal[15].

Em 2003 é criada a Controladoria Geral da União (CGU) com o objetivo de incrementar as ações de transparência e controle voltadas para o combate à corrupção (Gruman, 2012; Polo, 2015). Em 2004 a CGU lança o Portal da Transparência[16] (Polo, 2015) com o objetivo de centralizar a apresentação dos dados de uma série de sistemas governamentais brasileiros. A partir de 2009, com a publicação da Lei Complementar 131 que determinava a disponibilização das informações orçamentárias e financeiras em tempo real, há um aumento na abrangência das informações disponibilizadas no Portal (Paiva, Revoredo, & Baião, 2016), ampliando com isso a sua visibilidade.

Em setembro de 2011, o Brasil adere ao *Open Government Partnership* (OGP), como um dos países cofundadores (Dos Santos Brito et al., 2014; OGP, 2011). O OGP é uma iniciativa de oito nações (Brasil, Indonésia, México, Noruega, Filipinas, África do Sul, Reino Unido e Estados Unidos)[17] que tem como objetivo principal assegurar o combate a corrupção, a promoção da transparência e a participação cívica por meio de uma gestão pública aberta, eficaz e com responsabilidade (Freitas & Dacorso, 2014). Nessa mesma época, é criado o Comitê Interministerial para Governo Aberto (CIGA), cujo objetivo seria centralizar os debates relacionados à implementação e ao monitoramento do 1º *Plano de Ação Nacional para Governo Aberto* (Guimarães, 2014; OGP, 2011).

Em novembro de 2011, é promulgada a Lei de Acesso à Informação (LAI, Lei nº 12.527, de 18 de novembro de 2011) que define em seu artigo 3º "a publicidade como preceito geral e o sigilo como exceção e a divulgação de informações de interesse público, independentemente de solicitação" (A. K. Pereira, Pires, & Pinto, 2014). Na prática, a LAI finalmente regula o acesso à informação que foi previsto no inciso XXXIII do artigo 5º, no inciso II do § 3º do artigo 37 e no § 2º do artigo 216 da Constituição Federal de 1988 (Cintrão & Bizelli, 2013; Gruman, 2012).

Em abril de 2012, ocorre em Brasília a primeira conferência anual da Parceria para Governo Aberto, então com 38 países participantes (Guimarães, 2014). Nesse encontro, ocorre a criação de um Grupo de Trabalho (GT) da Sociedade Civil. Apesar da existência do GT, a sua participação no CIGA não tem caráter oficial, sendo por isso um motivo de atrito entre a sociedade civil e o governo (Steibel, 2014). Ainda em abril de 2012, a Instrução Normativa SLTI/MP nº 4, institui a Infraestrutura Nacional de Dados Abertos (INDA) que se constitui em "um conjunto de padrões,

[15] http://www.tesouro.fazenda.gov.br/siafi
[16] http://www.portaldatransparencia.gov.br/
[17] Em fev. de 2017, 75 países participavam da parceria - http://www.opengovpartnership.org/about.

tecnologias, procedimentos e mecanismos de controle necessários para atender às condições de disseminação e compartilhamento de dados e informações públicas no modelo de Dados Abertos"[18], conforme os Padrões de Interoperabilidade de Governo Eletrônico (e-PING)[19]. Em maio de 2012 a LAI entra efetivamente em vigor, conforme definido em seu artigo 47º. No mês seguinte, vai ao ar a primeira versão do portal oficial de dados abertos brasileiro - Dados.gov.br (Matheus & Ribeiro, 2014).

Em abril de 2013, a CGU publica o "Guia de Implantação de Portais de Transparência" com um conjunto de parâmetros ou sugestões para que os diferentes níveis federativos desenvolvam os respetivos portais de transparência (Resende & Nassif, 2015). No mês seguinte, maio de 2013, a CGU divulga ao fim da vigência do *1º Plano de Ação Nacional para Governo Aberto* uma avaliação apontando que, dos trinta compromissos firmados, vinte e seis haviam sido concluídos e dois não haviam sido iniciados conforme o planejado[20]. Dentre esses dois, o compromisso "Realização de pesquisa para identificação das demandas da sociedade sobre acesso a informação, visando ao aperfeiçoamento da política de transparência ativa" foi substituído pela criação do Sistema Eletrônico do Serviço de Informação ao Cidadão (e-SIC). Esse sistema possibilita aos cidadãos solicitar acesso às informações ainda não disponibilizadas em outros portais de forma simplificada[21]. Ainda em maio, é lançado o *2º Plano de Ação Nacional para Governo Aberto,* com inicialmente 45 compromissos assumidos por 17 órgãos do governo. Em julho de 2013, após uma revisão proposta pelo Grupo Executivo do CIGA, mais sete compromissos são incluídos no 2º Plano de Ação, sendo quatro desses oriundos de sugestões da sociedade civil.

Dentre as ações prevista dentro do 2º Plano, em 2014, podemos destacar a criação do Portal Brasileiro de Participação Social (www.participa.br) cujo objetivo é reunir "informações sobre oportunidades de participação social no governo federal e estimular a formação de comunidades de usuários" [17]. Outra ação que merece destaque em 2014 foi a realização do Programa Brasil Transparente, que tem por objetivo disseminar e apoiar a implantação da LAI e das iniciativas de governo aberto para os demais entes federativos.

No campo legislativo, em agosto de 2014, o Projeto de Lei (PL) 7804/2014[22] foi apresentado na Câmara dos Deputados, o que levaria à instituição da Lei de Dados Abertos, com o objetivo de normatizar os procedimentos para a disponibilização de dados abertos e interfaces de aplicação Web pelos entes federativos.

Em 2015, com a finalidade de avaliar a adoção das iniciativas de governo aberto pelos Estados e Municípios, a CGU lança um ranking, denominado de Escala Brasil Transparente (EBT), que consiste em um conjunto de doze quesitos de avaliação que tem o objetivo de medir o grau de transparência e observância à Lei de Acesso à Informação[23].

---

[18] http://wiki.gtinda.ibge.gov.br/

[19] https://www.governoeletronico.gov.br/eixos-de-atuacao/governo/gestao/interoperabilidade/eping-padroes-de-interoperabilidade-de-governo-eletronico

[20] CGU - 2º Plano de Ação Brasileiro - http://www.governoaberto.cgu.gov.br/no-brasil/planos-de-acao-1/2o-plano-de-acao-brasileiro.

[21] De janeiro de 2013 até janeiro de 2017 foram feitas 206.211 solicitações de acesso às informações ao e-SIC. Deve-se ressaltar que dessas, 35.562 solicitações foram negadas e outras 19.335 tiveram acesso parcialmente concedidos - https://esic.cgu.gov.br/sistema/site/index.html

[22] http://www.camara.gov.br/proposicoesWeb/fichadetramitacao?idProposicao=620193 (25 de setembro de 2017)

[23]http://www.cgu.gov.br/assuntos/transparencia-publica/escala-brasil-transparente/metodologia)

Em dezembro de 2015, ao final do prazo de vigência do *2º Plano de Ação Nacional para Governo Aberto,* 69% das ações previstas haviam sido implementadas e o restante se encontrava em execução pelos órgãos responsáveis.

Em 2016, o Governo Brasileiro inicia a construção do 3º *Plano de Ação Nacional para Governo Aberto*, para o biênio 2016 – 2018. O plano vigente é composto por dezasseis compromissos elaborados com a participação de representantes da sociedade civil e do governo, que nessa edição incluiu representantes dos poderes Executivo, Legislativo e Judiciário.

Não fazendo parte do 3º plano de ação, mas sendo inevitavelmente relacionado a este, é assinado o decreto nº 8.777 de maio de 2016[24] que institui finalmente a política de dados abertos do poder Executivo Federal.

Apesar dos avanços observados no período, as políticas de dados abertos no Brasil ainda enfrentam vários desafios. Problemas relacionados com questões tecnológicas e burocráticas fazem com que o país fique aquém de suas possibilidades no campo dos dados governamentais abertos.

Em relação às questões tecnológicas, mesmo com a criação da INDA, a inexistência de um padrão para a divulgação dos arquivos de dados dificulta sua plena utilização. Conforme apontado por Shen et al (2015), essa falta de aderência, dos dados disponibilizados, aos princípios dos Dados Abertos também acarreta em uma perda de posição do país nos rankings internacionais.

Além dessa diversidade de formatos de divulgação, questões como a ausência de uma documentação detalhada, diferenças na granularidade dos dados e frequência de atualização (Paiva et al., 2016) tornam a tarefa de utilização mais difícil.

2.3. DADOS EDUCACIONAIS

Para fins de exemplificação das soluções de DW do presente trabalho, serão utilizadas algumas bases de dados públicas relativas à temática da educação no Brasil.

O monitoramento e a análise dos dados educacionais representam, independente da matriz ideológica ou motivação, uma valiosa fonte de informação para o desenvolvimento das políticas públicas. Nesse sentido, podemos datar a utilização desses dados de forma mais intensiva após a segunda metade do século XX (Capistrano, Cirotto, Nascimento, & Silva, 2016).

Segundo Pronko (2015), na década de 1960 a Organização das Nações Unidas para a Educação, a Ciência e a Cultura (UNESCO) assume a educação como sendo o seu compromisso principal. Não obstante a educação já representar na época um direito básico universal[25], essa mudança de paradigma é justificada pelo fortalecimento da teoria do crescimento econômico oriunda do final da década de 1950, especialmente no que tange à questão do capital humano e nos investimentos para a formação dos indivíduos – nesse caso específico, a massa de trabalhadores (Post, 2015;

[24] http://www.planalto.gov.br/ccivil_03/_Ato2015-2018/2016/Decreto/D8777.htm
[25] http://unesdoc.unesco.org/images/0013/001394/139423por.pdf

Pronko, 2015; Viana & Lima, 2010). Essa visão economicista do processo educacional demandou das nações e dos organismos internacionais a produção de dados que permitisse o monitoramento e a avaliação dos investimentos realizados.

Nesse cenário destaca-se, além da própria UNESCO, a atuação de outros organismos internacionais na difusão do binômio "educação & desenvolvimento", tais como a Organização para a Cooperação e o Desenvolvimento Econômico (OECD), o Comitê de Assistência ao Desenvolvimento (DAC) e o Banco Mundial (WB). Esse viés econômico dos processos educacionais perdurou até meados da década de 1970. Pronko (2015) ilustra que nesse período o setor educacional do Banco Mundial era exclusivamente formado por profissionais da área de economia.

É na década de 1970 que ocorre a consolidação da OECD como um foro central na produção de estatísticas na área educacional ao redor do mundo. O Programa Internacional de Avaliação de Estudantes (PISA) é uma iniciativa desenvolvida pela organização que tem por objetivo a coleta de informações e a produção de indicadores que possibilitem a comparação entre países.

No final da década de 1990, é criado o Instituto de Estatística da Unesco (UIS) com um de seus objetivos principais apoiar o desenvolvimento de sistemas estatísticos nacionais voltados para as áreas de educação, ciência e cultura (Capistrano et al., 2016).

Além do Programa Internacional de Avaliação de Estudantes, várias outras iniciativas têm sido desenvolvidas a partir das bases de dados educacionais. Desde o início da década de 2000, o *Relatório Global de Monitoramento da Educação* (GEM) é produzido pela UNESCO com o objetivo de monitorar os progressos das metas educacionais – inicialmente do programa "Educação para Todos" (EFA) e posteriormente do "Objetivos de Desenvolvimento Sustentável" (SDG) – em diferentes países do mundo, subsidiando a sociedade civil em suas demandas por mudanças e forçando os governos para o debate (Post, 2015).

Não só os governos e a sociedade civil têm mostrado interesse nos dados educacionais. Os setores privados veem cada vez mais no segmento educacional um campo para a realização de investimentos e de lucros (Verger, Lubienski, & Steiner-Khamsi, 2016) e têm utilizado os dados educacionais para esse fim.

Sejam coletados por meio de pesquisas ou por meio de registros administrativos, os dados produzidos no processo de educar as futuras gerações provêm informações referentes a uma variada gama de métricas sobre alunos (absenteísmos, proficiência, desempenho, etc.), instituições (estrutura física e administrativa, corpo docente, etc) ou docente (nível de formação, área de atuação, etc.), que possibilitam a criação de indicadores e análises pelos mais variados critérios.

No âmbito dos países, a responsabilidade da produção e consolidação das informações educacionais públicas pode ser atribuída a diferentes atores. Normalmente estas atividades estão vinculados ao ministério ou departamento de educação do país ou a algum órgão ou instituto de alguma forma relacionado a esse (Capistrano et al., 2016). Nesses casos, as informações produzidas propõem-se a investigar a infraestrutura e organização (turmas, escolas ou universidades), as pessoas envolvidas no processo (discentes, docentes ou funcionários técnicos e administrativos) e os resultados obtidos (taxas de proficiência ou desempenho). Estes têm como principal objetivo monitorar e orientar a execução das políticas públicas na área educacional.

Também é possível encontrar informações educacionais em pesquisas, censitárias ou amostrais, produzidas diretamente pelos órgãos oficiais de estatística. As informações produzidas dessa forma estão relacionadas ao levantamento da condição de alfabetização ou situação educacional de uma determinada população.

Ambas as formas de produção de dados educacionais representam importantes fontes de informação sobre a situação da educação em um país e de certa forma são complementares, já que as informações produzidas no primeiro caso representam uma imagem sobre a estrutura produzida pelas políticas educacionais da atualidade, enquanto as estatísticas educacionais da população representam a consolidação dos efeitos das políticas educacionais do passado.

Ao longo da história do Brasil podemos obter dados educacionais em ambas as esferas de produção de informação. Gil (2009) data do final do século XIX a produção das primeiras estatísticas educacionais brasileiras, restrita nesse momento aos totais de escolas e alunos dos níveis básicos da educação.

Atualmente, o órgão responsável pela produção de informações sobre a estrutura do ensino brasileiro é o Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP). Criado em julho de 1939, com a denominação de "Instituto Nacional de Estudos Pedagógicos"[26], tinha desde o início como um dos seus objetivos principais a promoção de inquéritos e pesquisas relativos à organização do ensino no Brasil (Capistrano et al., 2016; Saviani, 2012). Ao longo de suas quase oito décadas de existência, o INEP passou por várias mudanças organizacionais que acompanharam o desenvolvimento da educação no país e no mundo.

Em meados da década de 1990, o INEP passa por um processo de redefinição de sua missão institucional que focou principalmente na questão da produção e disseminação de estatísticas destinadas ao melhoramento da avaliação educacional (Castro, 2000). Não por coincidência, remonta a essa época a reformulação ou criação de importantes pesquisas e avaliações voltadas à análise da situação educacional do Brasil (Censo da Educação Superior – 1995; Censo Escolar – 1996; Sistema Nacional de Avaliação da Educação Básica[27] – 1995; Exame Nacional do Ensino Médio – 1998).

Além dos dados produzidos pelo INEP, podemos encontrar informações sobre as condições de alfabetização e a situação educacional brasileira em alguns inquéritos promovidos pelo Instituto Brasileiro de Geografia e Estatística (IBGE). Criado em 1934, sob a denominação de Instituto Nacional de Estatística (Senra, 2009), o IBGE tem como uma de suas missões institucionais a "Produção e análise de informações estatísticas" e a "Coordenação e consolidação das informações estatísticas"[28] e nesse intuito realiza sistematicamente uma variedade de inquéritos sociais, demográficos e econômicos. Entre as pesquisas realizadas que investigam a situação educacional

[26] O INEP passou por outra mudança de denominação, em 1972, quando passou a ser chamado de Instituto Nacional de Estudos e Pesquisas Educacionais. A mudança para a atual denominação ocorreu em 2003, como uma homenagem ao intelectual, educador e escritor Anísio Spínola Teixeira (1900-1971), expoente da educação no Brasil.

[27] A primeira edição do Sistema Nacional de Avaliação da Educação Básica é do ano de 1990, mas somente em 1995, em sua terceira edição, é que este assume mais próximas ao perfil atual da avaliação, que estaria focado no monitoramento da situação da educação no Brasil (Bonamino & Franco, 2013).

[28] http://www.ibge.com.br/home/disseminacao/eventos/missao/default.shtm

da população brasileira, podemos citar os censos demográficos, realizados decenalmente, e as pesquisas por amostra de domicílios, de periodicidade anual ou trimestral.

Apesar de ambos os institutos divulgarem seus dados para consulta pública desde a década de 1990, as políticas de dados abertos destes datam de julho de 2016 e representam uma resposta oficial dessas instituições ao Decreto nº 8.777/2016. Tanto o Plano de Dados Abertos (PDA) do IBGE[29] como a portaria que institui o PDA do INEP[30] apresentam os alinhamentos que ambas instituições propõem para adequar a produção de informações aos modelos da LAI e do OGP.

2.4. Principais Bases de Dados da Educação no Brasil

Como já foi dito na secção anterior, tanto o INEP, quanto o IBGE produzem regularmente informações sobre o sistema educacional ou as condições educacionais do Brasil. Na presente secção, serão analisadas as bases relacionadas com a educação básica e superior que serão utilizadas como parte do levantamento de requisitos do presente trabalho, a saber:

- Censo Escolar da Educação Básica;
- Censo da Educação Superior;
- Censo Demográfico.

2.4.1. Censo Escolar da Educação Básica

A principal fonte de dados sobre a educação básica (educação infantil, ensino fundamental e médio) produzida pelo INEP é o Censo Escolar da Educação Básica, ou simplesmente Censo Escolar.

Antes da criação do INEP, várias iniciativas foram desenvolvidas pelo Estado com o objetivo de coletar e divulgar informações educacionais no Brasil. É possível datar uma dessas primeiras iniciativas em 1939, quando é realizada a primeira divulgação de um conjunto de estatísticas educacionais, produzida com dados coletados no início daquela década. Após essa primeira edição, as informações passam a ser sistematicamente publicados sob a denominação de "Sinopse Estatística" (Souza & Oliveira, 2012; Capistrano et al., 2016), que pode ser considerado como um precursor do atual Censo Escolar.

Ao longo dos anos, o Censo Escolar tem passado por várias alterações com a finalidade de se adequar às mudanças estruturais do sistema educacional brasileiro. Castro (2000) aponta que, em meados da década de 1990, ocorreram mudanças estrututurais no INEP (que passa a ser uma autarquia federal), no mesmo contexto em que houve a promulgação da lei Lei no 9.394, de 20 de dezembro de 1996, que estabelece as diretrizes e bases da educação brasileira, a qual indica no seu artigo 9º, inciso V, que é de obrigação do Estado "coletar, analisar e desseminar informações sobre a Educação".

[29] http://www.ibge.gov.br/home/disseminacao/eventos/missao/Plano_de_Dados_Abertos_IBGE_2016_2017_20160831.pdf
[30] http://download.inep.gov.br/institucional/legislacao/2016/portaria_n370.pdf

Nos moldes atuais, o Censo Escolar é uma pesquisa anual de carater declaratório, regulamentado pelo decreto nº 6.425/2006[31], que determina às instituições educacionais públicas e privadas o fornecimento, de forma mandatória, das informações solicitadas pelo INEP, por ocasião da realização do censo da educação ou para fins de elaboração de indicadores educacionais.

O processo de coleta das informações é realizado de forma descentralizada e em regime de colaboração entre os entes federados (estadual e municipal). A responsabilidade pelo fornecimento das informações é dos gestores dos estabelecimentos de ensino, cabendo ao poder público local o controle da veracidade das informações e do processo censitário (Capistrano et al., 2016). Ao INEP compete, além da organização conceitual do Censo Escolar, a consolidação final das informações e o assessoramento dos gestores locais sobre o processo de coleta das informações.

As informações coletadas são agregadas nas entidades Escolas, Turmas, Docentes e Alunos/Matrículas, sendo que é coletada uma variada gama de atributos para cada uma dessas dimensões. Desde 2007 as informações são coletadas por meio de um sistema informatizado que possibilita a identificação unívoca das entidades, facilitando assim o processo de coleta e controle (Capistrano et al., 2016).

Além de ser utilizado para o acompanhamento e planejamento do sistema de educação básica brasileira, o Censo Escolar é utilizado para balizar a distribuição de recursos federais, destinados à educação, aos demais entes.

2.4.2. Censo da Educação Superior

Da mesma forma que o Censo Escolar, o Censo da Educação Superior é realizado anualmente pelo INEP. Segundo o sítio do Instituto, este representa o "instrumento de pesquisa mais completo do Brasil sobre as Instituições de Educação Superior (IES)"[32].

Apesar de alguns autores datarem o processo de avaliação da educação superior no Brasil a partir da década de 1970 ou 1980 (Lacerda, Ferri, & Duarte, 2016; Polidori, Marinho-Araujo, & Barreyro, 2006), é possível apontar a criação do Serviço de Estatística da Educação e Cultura (Seec), em 1956, como uma das primeiras iniciativas para a coleta, apuração e divulgação de informações relativas às IES (BRASIL, 2004).

A partir de 1997, com as mudanças ocorridas no INEP, são editadas sucessivas portarias ministeriais com o intuito de regulamentar e aperfeiçoar a coleta de informações acerca das IES.

Atualmente, o preenchimento das informações é obrigatório a todas as IES do Brasil (universidades, centros universitários, faculdades integradas, faculdades, escolas ou institutos superiores e centros de educação tecnológica), públicas ou privadas, que possuam um ou mais cursos regularmente funcionando no início do ano-base do cadastragem. Os dados são coletados por meio de um formulário online, disponibilizados às instituições (BRASIL, 2004).

[31] http://www.planalto.gov.br/ccivil_03/_ato2007-2010/2008/Decreto/D6425.htm (25 de setembro de 2017)
[32] http://inep.gov.br/web/guest/censo-da-educacao-superior (25 de setembro de 2017)

As informações coletadas abrangem aspetos da organização física e administrativa das instituições tais como dados financeiros, infraestrutura, pessoal docente, pessoal técnico-administrativo, graduação presencial e à distância, entre outras.

Umas das principais utilizações das informações do Censo da Educação Superior é subsidiar o planejamento e a avaliação das ações e programas governamentais.

Além disso, os dados do Censo da Educação Superior são utilizados como insumo para o cálculo de alguns indicadores de qualidade do ensino superior, tais como o Índice Geral de Cursos (IGC) e o Conceito Preliminar de Curso (CPC).

2.4.3. Censo Demográfico

O primeiro recenseamento oficial brasileiro data de 1872, ainda sob o governo imperial, e representou à época um grande avanço na produção de estatísticas públicas no país, que até então só havia passado por experiências restritas a área menores, tais como províncias ou cidades. Tendo tido características inovadoras para a época, tais como a apuração centralizada e a investigação de características de homens livres e de escravos, teve seus trabalhos finalizados somente em 1876, tendo sido coordenando pela então Diretoria Geral de Estatística (Senra, 2009), que pode ser considerada a precursora do atual IBGE.

Sendo realizado decenalmente, nos anos de final zero[33], o Censo Demográfico brasileiro tem sofrido periódicas transformações a fim de se adequar às mudanças ocorridas na população ao longo das décadas. Nesse sentido, Oliveira & Simões (2005) exemplificam as mudanças ocorridas no censo de 1960, que passou por profundas alterações, de caráter técnico e metodológico, com a finalidade de adequar a coleta das informações ao crescimento da população no período.

Entre as mudanças ocorridas à época, vale destacar a reorganização dos questionários da pesquisa em dois questionários complementares. O primeiro, conhecido como questionário básico, deveria ser respondido pelo universo dos domicílios e investigava uma quantidade reduzida de questões ligadas às pessoas e às condições dos domicílios. Já o segundo, de constituição mais abrangente em relação às temáticas investigadas, deveria ser aplicado somente a uma fração de amostra da população total. Essa forma de aplicação persiste até os dias de hoje.

Em relação à temática educacional, é possível identificar uma evolução nos quesitos investigados ao longo das décadas (Rigotti, 2004). Essa evolução no processo de coleta de informações ocorre por causa da necessidade de a pesquisa acompanhar as alterações existentes no processo de pensar e desenvolver o ensino no país, e reflete as características de uma sociedade ainda em processo de transformação. Um exemplo disso seria a permanência da pergunta sobre alfabetização (Literacy) nas últimas seis edições do censo brasileiro, indicando uma incapacidade do Estado brasileiro em

[33] A exceção a isso foi o recenciamento de 1990, que por motivos da crise pela qual passava o Instituto e o país à época, foi somente realizado em 1991(Oliveira & Simões, 2005).

prover educação básica para toda sua população e por isso mantendo a questão relevante à investigação[34].

Atualmente, o Censo Demográfico representa a mais completa pesquisa domiciliar, de caráter sociodemográfico, que divulga informações para níveis geográficos inferiores ao municipal. Não obstante a divulgação dos seus resultados apresentar limitações causadas por seu longo período, representa uma valiosa fonte de informações para a elaboração do planejamento governamental, em todas as suas esferas de poder.

[34] É possível por meio do sítio do *Integrated Public Use Microdata Series* (Ipums-International) https://international.ipums.org/international-action/variables/LIT#availability_section (25 de setembro de 2017) - verificar a existência da variável em diferentes pesquisas ao redor do mundo.

3. SISTEMAS DE INFORMAÇÃO NA ÁREA DE CIÊNCIAS SOCIAIS

3.1. DATA WAREHOUSING

A busca por soluções que possibilitassem melhor armazenar, processar e compreender a crescente massa de dados que nos rodeia tem sido a razão principal da existência dos computadores desde a sua invenção.

Segundo Inmon, os chamados Sistemas de Apoio à Decisão (DSS, na sigla em inglês) remontam ao início dos anos de 1960 (Inmon, 2005). Nessa época, o processamento dos dados era realizado por meio de programações de "cartões perfurados" em linguagem COBOL e os dados eram armazenados em "arquivos mestres" baseados em fitas magnéticas. O processamento dos dados era uma tarefa restrita aos governos nacionais e suas instituições, que dispunham de recursos financeiros e tecnológicos e tempo para tal empreitada.

No decorrer dos anos o avanço tecnológico dos hardwares e softwares fez com que o processamento de grandes massas de dados passasse por um processo de disseminação em diferentes segmentos da sociedade e da economia, tais como universidades, instituições financeiras, bancárias e industriais. Nesse sentido, o desenvolvimento das tecnologias de banco de dados e o surgimento dos sistemas transacionais (OLTP), em meados da década de 1970, fizeram com que a produção de dados e o consumo de informações passassem por uma forte transformação (Inmon, 2005), que abriu o caminho para as técnicas de Data Warehousing[35], desenvolvidas e aperfeiçoadas nas décadas seguintes.

Desenvolvidas inicialmente para o tratamento e análise de informações na esfera corporativa e empresarial, as técnicas de Data Warehousing representam uma poderosa ferramenta para obtenção de vantagens competitivas para as empresas que as utilizam (Ponniah, 2001).

Há uma literatura consolidada que detalha a construção de soluções de Data Warehousing em ambientes corporativos e empresariais (Golfarelli & Rizzi, 1999; Group Kimball, 2013; Inmon, 2005; Kimball, 1998; Kimball & Caserta, 2015; Kimball & Ross, 2013). Nesses ambientes as informações trafegam entre os sistemas transacionais e analíticos de forma relativamente fechada (Paterson, 2003), já que a maioria das fases envolvidas na produção, no tratamento e no consumo das informações residem dentro de uma mesma instituição (Strand, 2005) e são norteadas pelos objetivos de negócio dessa. Mesmo que haja a utilização de dados produzidos fora dessas instituições, tais como informações mercadológicas ou macroeconômicas, essas destinam-se a complementar o painel analítico e conjuntural em que a instituição se encontra e representam uma pequena fração do esforço necessário para a construção da solução de suporte à decisão.

Mesmo não sendo um processo simples ou mesmo unívoco, já que pode ser realizado de diferentes maneiras, a construção de uma solução de Data Warehousing tem sido repetidamente mapeada. Podemos apontar diferentes versões desse mapeamento, desde a proposta apresentada por Golfarelli & Rizzi, de 1999 (Figura 3), que se baseiam em uma estrutura tradicional de modelagem de banco de dados relacionais, até abordagens mais atuais que levam em consideração técnicas

[35] Por *Data Warehousing* se compreende como sendo o conjunto de procedimentos voltados para a transformação de dados operacionais em informações utilizadas para nos processos de tomada de decisões.

que buscam a extrair automaticamente (sempre que possível) a estrutura dos dados de entrada e propor soluções para os processos de extração e armazenamento (Berro, Megdiche, & Teste, 2015)

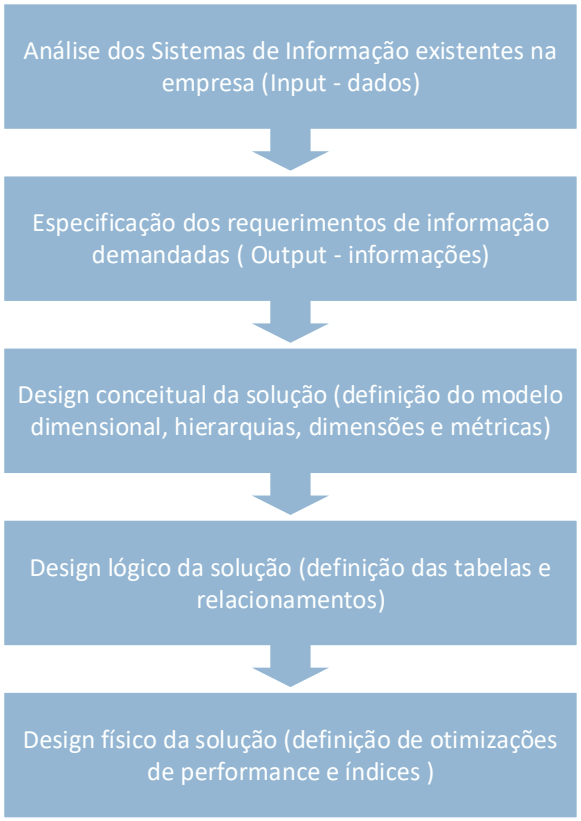

Figura 3 - Fluxo de desenvolvimento de um Data Warehouse. Baseado em Golfarelli & Rizzi, 1999.

3.2. DATA WAREHOUSING NAS CIÊNCIAS SOCIAIS

Mesmo existindo uma vasta literatura que trata do desenvolvimento de sistemas de Data Warehouse, a construção de uma solução de DW baseada em estatísticas públicas e com foco nas Ciências Sociais apresenta características e desafios próprios.

Diferentes autores relatam que a construção de sistemas de informação baseados em estatísticas públicas costuma apresentar peculiaridades relacionadas a integração e compatibilização dos dados (Paterson, 2003), à forma de realização do levantamento de requisitos e necessidades dos usuários (Schaefer, Tanrikulu, & Breiter, 2011) ou à quantidade de métricas ou dimensões utilizadas (Berndt, Hevner, & Studnicki, 2003; Paterson, 2003).

Paterson (2003) detalha essas diferenças e as agrupa em cinco grandes categorias: contexto institucional, grupo de possíveis usuários, fontes de dados, processos de integração dos dados e as estratégias de análise possíveis.

O contexto institucional define a visibilidade, os objetivos e as diretrizes que vão nortear a construção da solução de DW. No caso das soluções empresariais, esses três componentes normalmente são bem definidos e, mesmo podendo variar entre empresas, são focados para o

interior da própria instituição com o objetivo de aferir vantagens competitivas (lucro) e guiados pelas estratégias de negócio elaboradas internamente.

Na construção de uma solução voltada para o público das Ciências Sociais esses componentes não são necessariamente bem claros nem exclusivos. Nesses casos, as soluções normalmente apresentam um escopo aberto, pois, mesmo quando são desenvolvidos dentro de uma única instituição, são elaborados para terem visibilidade fora destas. Também são diferentes seus objetivos e diretrizes, visto que são focados em atender demandas sociais e favorecer a elaboração de políticas públicas (que podem variar dependendo do local e da época) bem como são guiados por normas constitucionais ou tratados internacionais – que indicam muito mais quais são os parâmetros desejados ou ideais, do que os caminhos para serem seguidos.

A análise do grupo de utilizadores dos dois sistemas (os corporativos e os voltados para a área social) revela grandes diferenças entre eles. Enquanto os utilizadores dos sistemas corporativos são compostos pelos níveis gerenciais ou técnicos destas instituições, as soluções voltadas para a área social possuem um corpo de utilizadores heterogêneos, que pode variar de um ministro de estado ao cidadão comum.

Essa heterogeneidade dos utilizadores fica latente em diferentes fases do processo de construção dos DW na área social. Conforme apontou Schaefer *et al* (2011), o levantamento de requisitos do sistema nesses casos costuma ser diferenciado, uma vez que as informações precisam ser obtidas, por exemplo, entre diferentes pesquisadores individuais que podem estar espalhados em distintas instituições a trabalhar propósitos particulares. Além disso, como há uma grande diversidade de metodologias, conceitos e objetivos empregados por estes, torna-se difícil algumas vezes atingir alguns consensos sobre o melhor design a ser utilizado. Isso fica visível quando comparamos o número de métricas necessárias para atender as demandas de informação de cada grupo de usuários. Enquanto os usuários corporativos trabalham com, no máximo, duas dezenas de métricas devido à multiplicidade de objetivos e público da solução, a quantidade de métricas pode passar facilmente desse número. Com efeito, Berndt et al. (2003), tratam de um sistema de DW com mais de 250 “indicadores” voltados para fins de planejamento de saúde de comunidade locais da Flórida (EUA).

Talvez sejam no âmbito das fontes e integração de dados que as diferenças entre as duas estruturas sejam mais evidentes. Quando tratamos características relativas à aquisição dos dados, os sistemas corporativos se caracterizam por apresentar uma frequência alta e regular, normalmente diária ou semanal. Os dados são fortemente padronizados e obedecem às especificações definidas pelas estratégias de negócio da empresa.

Ao passo que nos sistemas voltados para ciências sociais, a periodicidade de aquisição dos dados é baixa, sendo muito comum apresentar periodicidade anual (dados demográficos) ou trimestral (dados econômicos). A padronização dos dados existe, mas apresenta maior grau de variabilidade ao longo do tempo, já que os fenômenos por estes investigados costumam variar conforme os ciclos de mudanças sociais.

Outra grande diferença reside na forma como os dados são produzidos. No caso dos dados corporativos, esses são oriundos dos sistemas transacionais atrelados aos processos produtivos e aos negócios das empresas. Enquanto os dados que alimentam os DW sociais, provêm

normalmente de pesquisas censitárias ou inquéritos socioeconômicos. Mesmo quando são oriundos de registros administrativos (informações sobre atendimentos hospitalares, por exemplo) costuma haver uma latência entre a sua produção/aquisição e a divulgação final.

No tocante à integração dos dados, uma das diferenças entre as soluções reside na forma pela qual as fontes de dadossão acedidas. Como já foi dito anteriormente, os dados nas soluções corporativas são normalmente extraídos dos bancos de dados relacionais que dão suporte aos sistemas transacionais destas. No caso dos dados do DW social, pelo fato de os dados serem produzidos – na sua maioria – por organizações públicas ligadas às diferentes esferas de governo, é quase certo que o acesso aos "sistemas transacionais" dessas instituições não seja possível, limitando o acesso aos arquivos de estrutura unidimensionais (*flat files*) disponibilizados[36]. Dependendo dos requisitos de informações levantados com os usuários ou pesquisadores, a quantidade de fontes de dados pode crescer ao ponto de tornar o processo de carga dos dados bastante complexo, já que cada uma das fontes utilizadas pode apresentar layout, padronização e codificação diferentes, não só entre instituições, mas entre fontes/pesquisas de uma mesma instituição. Normalmente, a ligação entre as diferentes fontes ocorre somente por médio das dimensões geográfica (território) e temporal.

Por último, o consumo e análise das informações produzidas também apresenta diferenças entre ambos os tipos de soluções. Enquanto nas soluções corporativas a utilização de ferramentas de OLAP (Online Analytical Processing) ou de dashboards com a representação visual dos indicadores chave de desempenho (KPIs) são corriqueiras, no ambiente das Ciências Sociais essas soluções nem sempre são possíveis ou desejáveis. Como já foi dito acima, a heterogeneidade dos utilizadores, assim como a multiplicidade de objetivos desses sistemas, faz com que a utilização de ferramentas ou técnicas mais complexas de visualização de dados intimide ou até afaste possíveis utilizadores não especialistas. Nesses casos, pode ser comum o desenvolvimento de mais de uma ferramenta a fim de atender esses diferentes públicos. Abaixo, a Tabela 1 apresenta um resumo dessas diferenças.

	DATA WAREHOUSE EMPRESARIAL	DATA WAREHOUSE NAS CIÊNCIAS SOCIAIS
CONTEXTO		
Escopo Institucional	Interno a empresa	Uma ou mais instituições, com visibilidade aberta muitas vezes para a sociedade
Objetivo	Obter vantagens competitivas	Atender demandas sociais e a elaboração de políticas públicas
Diretrizes	Estratégia corporativa	Guiados por normas constitucionais ou tratados internacionais
UTILIZADORES		
Base de utilizadores	Homogêneo - Níveis técnicos e gerenciais das empresas	Heterogêneo - Gestores governamentais, pesquisadores e o público em geral

[36] Mais recentemente, algumas instituições brasileiras produtoras de dados passaram a disponibilizar seus dados em formatos XML e JSON.

DADOS		
Frequência de aquisição	Alta e regular - aquisição diária ou semanal	Baixa e em alguns casos irregular - Aquisição Decenal, anual, trimestral ou mensal
Padronização	Alto nível de padronização	Padronização existe, mas pode variar ao longo do tempo e entre as diferentes fontes e instituições
Forma de aquisição	Especificada pelos níveis gerenciais e os objetivos das empresas	Especificado por diferentes grupos de pesquisadores ou técnicos governamentais
INTEGRAÇÃO DOS DADOS		
Fonte(s)	Limitada - normalmente aos sistemas transacionais internos	Múltipla - produzidas por diferentes órgãos governamentais
Grau de ligação	Alto, já que os dados são construídos com essa finalidade	Baixo, normalmente limitados aos níveis geográfico e temporal
Organização	Organizados em bases de dados relacionais	Organizado segundo temáticas e disponibilizados em arquivos de estrutura unidimensionais
ANÁLISES		
Ferramentas de análise	Ferramentas OLAP e Dashboard	Diferentes formas de visualização, dependendo do público utilizador.

Tabela 1 - Resumo das diferenças entre as estruturas. Baseado em Paterson (2003)

3.3. INDICADORES SOCIAIS COMO MÉTRICAS DE UM DW

Não só pela grande quantidade é que se destaca a construção das métricas em um DW voltado para a área de humanidades.

Conhecidos como Indicadores Sociais, tais métricas possuem trajetória tão longa em tempo quanto o surgimento dos primeiros grandes computadores comerciais. Datando de meados da década de 1960, o chamado "Movimento de Indicadores Sociais" surge como uma tentativa de aperfeiçoar os sistemas estatísticos para acompanhamento das transformações sociais em um cenário dividido entre o crescimento econômico e a persistência de elevados níveis de pobreza, observados ao redor do mundo (Jannuzzi, 2001).

Ainda segundo Jannuzzi, um indicador social é uma tradução "em cifras tangíveis e operacionais" de diferentes dimensões que povoam a realidade de uma determinada sociedade. Essas cifras tangíveis nada são mais que medidas, na sua maioria quantitativas, que tentam cristalizar fenômenos sociais de modo a torná-los operacionalizáveis, seja para fins acadêmicos, seja para a formulação de políticas públicas.

A natureza quantitativa dos indicadores sociais é uma característica desejável – na medida em que possibilita a comparação – ao longo do tempo ou entre diferentes regiões geográficas de diferentes dimensões (gênero, raça, idade, classe social, etc.) da vida social de uma determinada população. Com isso, medidas como taxa de analfabetismo, taxa de proficiência ou custo médio por aluno no

ensino público podem ser calculadas e servir de parâmetros para a correção ou aperfeiçoamentos das políticas educacionais de um país ou região.

Por consistirem predominantemente em percentagens e rácios (razões), os indicadores sociais apresentam uma característica que influencia o processo de planejamento e desenvolvimento de soluções de DW para ciências humanas. Tais medidas constituem os chamados *fatos não-aditivos* que são aqueles que, em uma estrutura dimensional, não podem ser somados em nenhuma de suas dimensões (tais como tempo ou território, por exemplo), já que o resultado de tal soma não possui qualquer significado válido. Esses casos, a literatura (Group Kimball, 2013) recomenda que sejam armazenados separadamente os componentes totalmente aditivos do indicador (no caso de uma percentagem, os valores do numerador e do denominador) para que, no momento da exibição dos resultados, sejam realizados os cálculos necessários para a criação do indicador. Apesar de relativamente simples, tal solução esconde algumas armadilhas. Como veremos no capítulo 4.4, é necessário que haja um cuidadoso mapeamento dos indicadores para evitar o armazenamento duplicado de informações, tais como totais populacionais. Além disso, deve ser levado em consideração que, dependendo da ferramenta utilizada, a execução dos cálculos em tempo execução da demanda, pode afetar o desempenho desta.

3.4. A DIMENSÃO TEMPO NAS CIÊNCIAS SOCIAIS

A dimensão temporal ou somente tempo é fundamental na construção de qualquer sistema de DW, pois permite aos utilizadores acompanhar a evolução ou mudanças nos fenômenos e eventos investigados. Nos sistemas de DW empresariais a dimensão tempo apresenta um alto nível de detalhamento, normalmente ao nível das datas do calendário, com granularidade de um único dia (Kimball & Caserta, 2015), mas podendo chegar ao detalhamento de horas e minutos - em sistemas que analisem as vendas ao longo de um dia, por exemplo.

No caso dos sistemas desenvolvidos para as Ciências Sociais, quase nenhuma informação é produzida com um grau de detalhamento no que concerne à hora e minuto da ocorrência do evento[37]. Mesmo quando possuem uma periodicidade de produção diária, na maioria das vezes as informações são divulgadas de forma a serem agregadas em unidades de análise mensais ou trimestrais (como no caso dos dados sobre mercado de trabalho) ou anuais (nos casos de dados demográficos ou educacionais). Na maioria dos casos, para as análises normalmente realizadas com essas informações, a periodicidade de divulgação é mais que suficiente para satisfazer as necessidades investigativas (institutos de pesquisas ou acadêmicas) ou práticas (técnicos governamentais e formuladores de políticas públicas).

Isso não significa que os sistemas de DW para as ciências sociais não possam trabalhar com níveis de detalhamento temporal mais próximos aos encontrados em seus pares corporativos. É possível pensar na utilização de técnicas de DW e de mineração de dados para medir fenômenos, tais como a evasão escolar de alunos, por meio da combinação de diferentes fontes de dados, em que uma

[37] Os dados de mortalidade são uma exceção a essa regra, já que possuem a indicação do dia e hora em que o óbito ocorreu.

das granularidades poderia ser o absenteísmo diário dos alunos, extraído a partir dos sistemas transacionais das instituições de ensino (Rigo, Cambruzzi, Barbosa, & Cazella, 2014).

3.5. DESAFIOS PARA A CONSTRUÇÃO DE UM DW COM OS DADOS PÚBLICOS BRASILEIROS

No caso dos dados públicos brasileiros, é possível identificar vários pontos indicados pelos autores. No que concerne à questão das fontes de informação, encontramos na história recente do país uma grande diversidade de instituições produtoras de informações públicas, sendo extensa a relação de bases de dados produzidas e disponibilizadas por estas instituições.

Em outubro de 2017, estavam disponibilizados no catálogo do Portal Brasileiro de Dados Abertos[38] aproximadamente 3.290 conjuntos de dados, produzidos por 92 diferentes organizações, dentre ministérios e agências governamentais. Além dessas fontes, ainda existem os dados produzidos e divulgados pelo Instituto Brasileiro de Geografia e Estatística (IBGE), o Instituto Nacional de Estudos e Pesquisas Educacionais Anísio Teixeira (INEP) e pelo Departamento de Informática do Sistema Único de Saúde (DATASUS), que apesar de não estarem listados no portal, são regularmente disponibilizados na forma de microdados.

Essa grande variedade de fontes e instituições exige que o desenvolvimento de qualquer solução tenha que considerar diferentes padrões de divulgação dos dados e metadados na sua elaboração.

Além da variedade de bases existentes, o desenvolvimento dessas soluções de Data Warehousing, deve lidar com alterações na estrutura dos arquivos, que ocorrem ao longo do tempo dentro de um mesmo conjunto de dados. Essas alterações devem-se, em parte, à complexidade dos fenômenos sociais estudados (Silva, 2003) que acarreta, conforme a área temática, uma constante atualização das informações investigadas.

Ao analisar somente duas das bases de dados (*Censo Escolar da Educação Básica* e o *Censo da Educação Superior*) produzidas pelo Instituto Nacional de Estudos e Pesquisas Educacionais (INEP), é possível identificar um aumento no número de variáveis investigadas de aproximadamente 49,8% (Censo Escolar) e de 31,2% (Censo da Educação Superior), entre os anos de 2010 e 2015[39].

Outras bases de dados públicas apresentam as mesmas características. A Pesquisa Nacional por Amostra de Domicílios (PNAD), divulgada anualmente pelo IBGE desde a década de 1970 e descontinuada em 2015, comumente apresentava alterações em seu layout devido a inclusões, ao seu corpo básico tradicional, de suplementos temáticos especiais.

Essas constantes atualizações, além da diversidade de formas de estruturação das bases de dados, obrigam que os processos de extração, transformação e carga (ETL, do inglês *Extract Transform Load*), normais no desenvolvimento de um DW, tenham que ser constantemente revistos e atualizados, de modo a aumentar o custo dessa etapa já tradicionalmente dispendiosa (Kimball & Caserta, 2015).

[38] http://dados.gov.br/ (consultado em 10 de outubro de 2017).
[39] Levantamento realizado pelo autor a partir dos metadados disponibilizados pelo Instituto Nacional de Estudos e Pesquisas Educacionais (INEP).

O entendimento dessas particularidades pode representar um dos fatores de sucesso ou fracasso no desenvolvimento de uma solução baseada em estatísticas públicas e por esse motivo devem ser melhor compreendidas e documentadas.

4. IMPLEMENTAÇÃO DA SOLUÇÃO DE DATA WAREHOUSING

4.1. ABORDAGENS PARA A CONSTRUÇÃO DE UM DATA WAREHOUSE

Segundo vários autores (List et al., 2002; Sá et al., 2012), existem três abordagens comumente utilizadas no processo de desenvolvimento de um sistema de Data Warehouse: “Goal-driven”, “Data-driven” ou “User-driven”.

Baseada nesses autores, a tabela abaixo (Tabela 2) descreve as principais características dessas três abordagens.

Abordagem	Foco	Ator	Vantagens	Desvantagens
Goal-driven	Key Performance Indicator (KPI)	Níveis gerenciais médios e altos	É focado diretamente nas métricas de desempenho da empresa.	Depende do envolvimento dos níveis gerenciais.
User-driven	Necessidades do negócio	Usuários da camada de negócio	É baseado na participação direta dos usuários no processo de construção do modelo	Pode confundir os objetivos dos usuários com os objetivos do negócio. Além disso, por não levar em consideração outros aspectos a priori (análise dos dados disponíveis).
Data-driven	Base de dados	Sistema transacional (Modelo Entidade Relacionamento)	Possibilita que o modelo dimensional seja criado de forma rápida.	Pode criar estruturas de dados disfuncionais, já que pode haver uma desconexão entre os dados armazenados e as necessidades dos usuários.

Tabela 2 - Principais características das abordagens utilizadas para o desenvolvimento de um DW.

Podendo ser utilizadas separadamente ou de forma combinada (Guo et al., 2006; Sá et al., 2012), essas abordagens levam em consideração que os elementos necessários para a elaboração de um DW (objetivos, usuários e dados) residem sob uma mesma instituição.

No caso dos repositórios de dados abertos dificilmente é possível encontrar as condições necessárias para a utilização dessas abordagens de forma pura, seja porque há diferentes objetivos para a construção do repositório, conforme o grupo de usuários envolvido (Schaefer et al., 2011), seja porque os dados encontram-se dispersos entre diferentes instituições, com diferentes graus de estruturação (B et al., 2015; Berro et al., 2015; Coletta et al., 2012; Heise & Naumann, 2012).

Essas características (poder assumir diferentes finalidades e ter as fontes de dados com baixa estruturação) fazem com que as soluções de DW baseadas em dados abertos tenham que se valer de novas abordagens para a criação dos repositórios, em especial no que se refere às tarefas de integração dos dados (ETL).

No presente trabalho, pretendemos desenvolver parte da abordagem proposta por Berro et al. (2015), em que o processo de desenvolvimento do repositório é guiado de forma automática, "sempre que possível" pela estrutura encontrada nas fontes dos dados.

Essa abordagem, descrita pelos autores como "Content-driven", aparenta trazer vantagens para o desenvolvimento de soluções de repositório de dados abertos, já que se baseia nas estruturas existentes nos arquivos unidimensionais (CSV, TXT, XLS, etc.) e não nos sistemas transacionais – indisponíveis para a maioria dos usuários, no caso dos dados abertos.

A partir da análise das estruturas existentes nos arquivos unidimensionais, espera-se identificar padrões que auxiliem no processo de desenvolvimento do modelo multidimensional.

4.2. LEVANTAMENTO DOS REQUISITOS

Atualmente, o Instituto Brasileiro de Análise Sociais e Econômicas - IBASE possui uma solução para a coleta e visualização de informações dentro de um sistema denominado Indicadores da Cidadania (Incid). O Incid[40] foi desenvolvido em 2011 com a finalidade de analisar a efetividade da cidadania observada no território para monitoramento do estado da cidadania em 14 municípios da chamada Área de Influência do Complexo Petroquímico do Estado do Rio de Janeiro (Comperj).

As ferramentas atualmente em funcionamento são:

O Sistema de Indicadores de Cidadania - É composto por indicadores de caráter quantitativo que buscam interpretar as informações sob a perspetiva de Cidadania Efetiva, isto é, "o direito de todos e todas a terem direitos iguais, fundada no reconhecimento em nós e nos outros da titularidade de direitos e de responsabilidades comuns por eles".

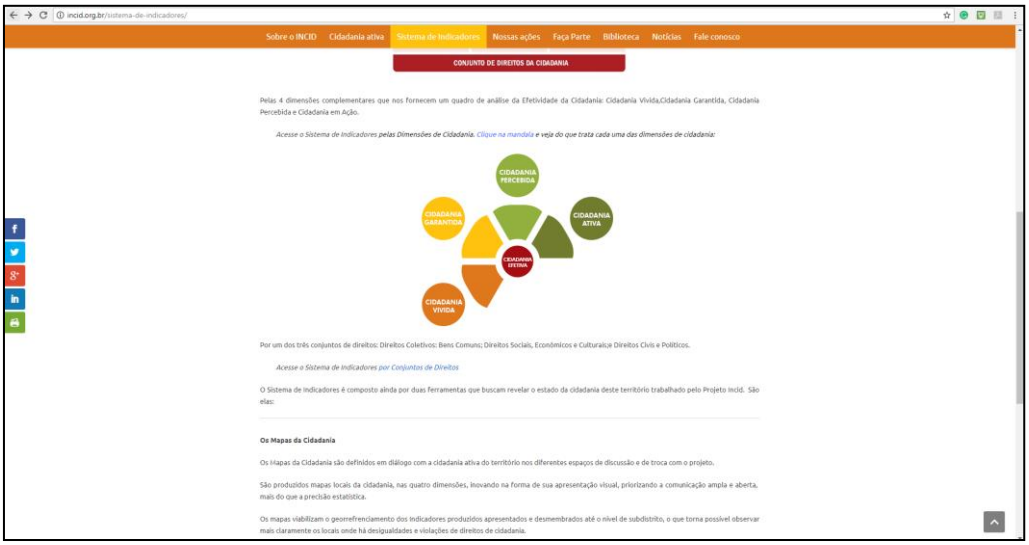

Figura 4 - Site do Incid. Detalhamento das "dimensões da cidadania".

[40] http://incid.org.br/

Mapas da Cidadania – Enquanto o Sistema de Indicadores de Cidadania representa a parte conceitual do projeto, os Mapas da Cidadania representam a forma de visualização e consumo das informações armazenadas no sistema. A ideia por traz dos Mapas da Cidadania é possibilitar um acesso fácil às informações sob a forma de mapas temáticos (cartogramas) de leitura rápida e intuitiva.

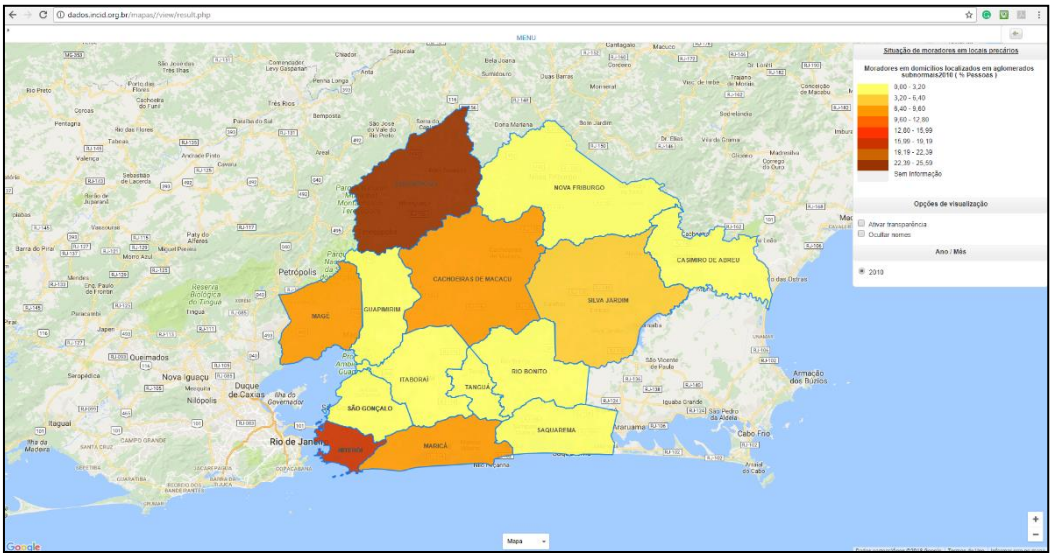

Figura 5 - Exemplo de cartograma dos Mapas da Cidadania.

Além de apresentar as informações na forma de mapas, a ferramenta também possibilita a geração de gráficos e tabelas exportáveis.

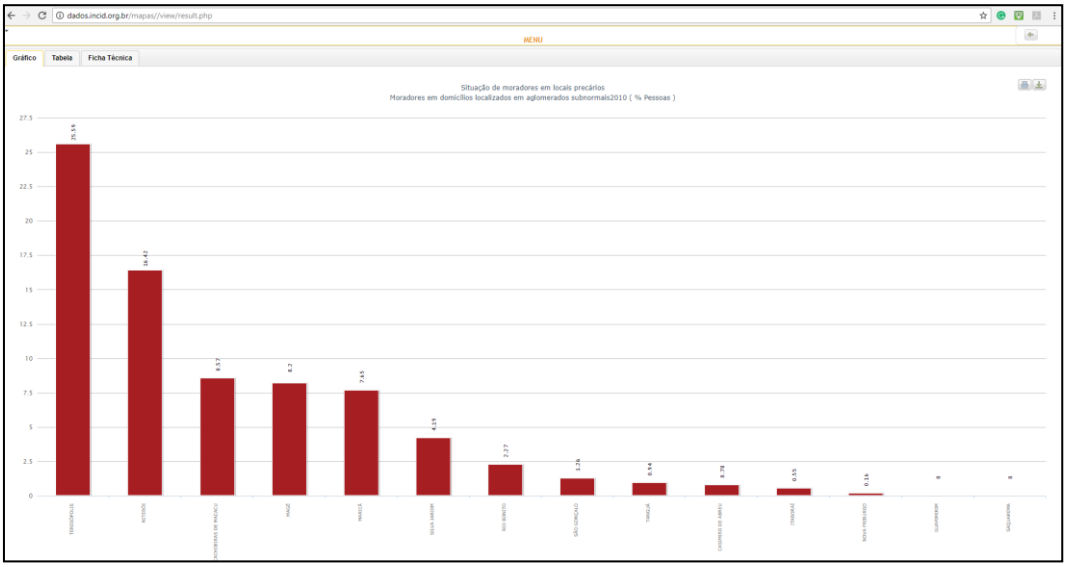

Figura 6 - Exemplo de tabela gerada pela ferramenta Mapas da Cidadania.

Na medida em que precisa ser uma ferramenta de fácil utilização para atingir a um público não especialista, composto por lideranças sociais, ativistas de direitos humanos ou causas ambientas e até mesmo o cidadão comum sem nenhum vínculo institucional, a ferramenta desenvolvida não incorporou funcionalidades típicas das aplicações de BI/OLAP, tais como *Drill Up, Drill Down ou Drill & Dice*, por exemplo, limitando-se a uma simples espacialização das informações.

Além de ter que atender a um público não especialista, o desenvolvimento da ferramenta teve que se adequar à infraestrutura de TI disponível na instituição. Como não dispõe de servidores próprios, o IBASE faz uso de hospedagens contratadas que oferecem suporte para aplicações desenvolvidas em linguagem PHP com JavaScript. Os dados das aplicações são armazenados em bancos de dados MySQL e MariaDB.

A partir desse cenário inicial, foi realizado um levantamento de requisitos junto à equipe de pesquisadores da instituição para se saber quais as características básicas da solução desejada. Com base nesse levantamento, foram analisadas as fontes de dados indicadas para a primeira etapa do projeto para que suas características fossem levantadas.

Abaixo, as três principais demandas apresentadas pela equipe de pesquisadores:

- Possibilidade de os utilizadores dedicarem seus esforços somente na construção dos indicadores e métricas, deixando os detalhes de TI o mais transparente possível.
- Deve ser uma solução de baixo custo e que possa ser hospedada dentro das limitações tecnológicas da estrutura existente na organização.
- Possibilite processar e consultar indicadores produzidos com diferentes bases de dados (com diferentes layouts e granularidades).

Além das demandas supracitadas, foram levantadas as seguintes características das bases que deverão ser utilizadas e dos indicadores produzidos:

- Pelo fato de mais de 90% dos indicadores sociais utilizados serem taxa, percentuais ou rácios (razões), a solução terá que lidar quase exclusivamente com métricas não-aditivas.
- A grande variedade de bases e layouts torna o processo de ETL uma atividade mais custosa que o normal.
- Como essas diferentes bases possuem granularidades diferentes, tanto na dimensão temporal como na geográfica, faz com que o armazenamento dos fatos não possa ser realizado de maneira direta em uma única estrutura.

A questão da diversidade de granularidades existentes nas bases de dados apresenta um desafio no momento da definição da estrutura do DW, visto que, segundo Kimball (Kimball & Ross, 2013), deve-se evitar a todo o custo misturar factos de granularidade diferente em uma única estrutura.

PESQUISA	BASE	GRÃO	
		GEO	TEMPO
CENSO ESCOLAR	ESCOLA	DISTRITO	ANO
	TURMA	DISTRITO	ANO
	MATRICULA	DISTRITO	ANO
	DOCENTE	DISTRITO	ANO
CENSO DEMOGRÁFICO - AMOSTRA	PESSOA	MUNIC	ANO
	DOMICILIO	MUNIC	ANO
CENSO IES	IES	MUNIC	ANO

	CURSO	MUNIC	ANO
	DOCENTES	MUNIC	ANO
	ALUNOS	MUNIC	ANO
SIM	OBTOS	MUNIC	ANO
SINASC	NASCIMENTO	MUNIC	ANO
RAIS	TRABALHADOR	MUNIC	ANO
MUNIC	MUNIC	MUNIC	ANO
CAGED	TRABALHADOR	MUNIC	MÊS
PNAD CONTINUA	PESSOA	RM	TRIMESTRE
CENSO DEMOGRÁFICO - UNIVERSO	SETOR	SETOR	ANO
PNAD	PESSOA	RM	ANO
	DOMICILIO	RM	ANO

Tabela 3 - Granularidade das principais fontes de dados utilizada na construção do DW.

Como é possível ver na tabela 3, as bases trabalhadas possuem diferenças de granularidade, tanto na dimensão temporal quanto na geográfica. Em relação à dimensão temporal, aproximadamente 90% das bases que se deseja trabalhar apresentam periodicidade anual. Em relação à granularidade geográfica, predominam os níveis municipais (MUNIC – 58%) e distrital (DISTRITO – 21%).

Além das dimensões temporais e geográfica, que serão obrigatórias na construção de qualquer indicador/métrica, os pesquisadores da instituição levantaram a necessidade – para essa fase do projeto – de inclusão de três novas dimensões importantes na análise das condições de vida e para o entendimento dos fenômenos sociais que se pretende estudar.

As dimensões em questão são:

Dimensão Sexo/Gênero: busca investigar as diferenças observadas entre homens e mulheres em fenômenos como a remuneração do trabalho, acesso a serviços ou direitos, grau de letramento/escolaridade, entre outros.

Dimensão Etária: busca avaliar a forma como os fenômenos sociais se distribuem ao longo das etapas de vida das pessoas. A dimensão etária possibilita que um mesmo indicador seja visualizado em diferentes momentos ou recortes da vida, fazendo com que o rol de análise seja expandido.

Dimensão Étnico-racial: busca avaliar as diferenças observadas entre os indivíduos segundo a sua cor autodeclarada[41] nas pesquisas ou registros administrativos. Não sendo comum em muitos países, principalmente no continente europeu, as análises produzidas a partir dos recortes dessa dimensão possibilitam avaliar as condições dos diferentes grupos étnico-raciais em relação a diferentes fenômenos sociais. No caso brasileiro, essa dimensão tem um papel muito importante na explicação de fenômenos, tais como a taxa de homicídios de jovens, tornando-se fundamental em um DW multitemático.

[41] A autodeclaração de cor é a forma como essa variável é coletada nas diferentes pesquisa e registros administrativos do Brasil. A ideia central é dessa metodologia é respeitar a resposta da pessoa entrevista/atendida (Petruccelli & Saboia, 2013). Deve-se ressaltar que o Estado brasileiro não possui uma tipologia oficial de classificação étnico-racial de sua população, apesar de que a classificação utilizada pelo IBGE, ser adotada por diversas outras instituições.

Segundo o levantamento realizado, pode ocorrer que a unidade de investigação em análise não possua essas características/dimensões acima listadas. Isso ocorre, por exemplo, quando a unidade em questão se trata de uma unidade administrativa ou funcional (uma administração municipal ou uma escola, por exemplo). Nesses casos, para se evitar a criação de um novo repositório será criada nessas dimensões uma categoria "Não se aplica", para tratar desses casos.

4.3. ANÁLISE DAS FONTES DE DADOS

Para fins da presente dissertação, só serão analisadas três fontes de dados voltadas para a investigação do sistema educacional brasileiro, como dito na secção 2.4. Será feita nessa secção uma rápida análise da estrutura física dos arquivos de dados e metadados com a finalidade de auxiliar no processo de desenvolvimento da solução.

As bases são: Censo Escolar da Educação Básica, Censo da Educação Superior e Censo Demográfico. No caso das duas primeiras bases, serão analisados os últimos cinco anos divulgados (2012 – 2016). Já no caso do Censo Demográfico, será trabalhada a última edição divulgada em 2010.

4.3.1. Censo Escolar da Educação Básica

O censo escolar é composto por quatro conjuntos de arquivos representando as diferentes entidades investigadas pela pesquisa (Escolas, Turmas, Docentes e Alunos/Matriculas). Além disso, para os dados de Docentes e Alunos/Matrículas, as bases são divididas em cinco arquivos, segundo a macrorregião a qual as entidades pertencem. As bases apresentam ao longo do tempo grandes alterações em relação ao layout dos arquivos e dos metadados da pesquisa. Até o ano de 2012 os dados eram divulgados em arquivos texto (*.TXT), em que as variáveis/campos eram definidas com base na sua posição inicial dentro do arquivo e no seu tamanho. A partir de 2013 os arquivos passam a ser disponibilizados em formato delimitado por vírgula (*.CSV)[42]. No caso dos metadados dos arquivos, as informações são fornecidas em arquivos do Microsoft Excel.

Analisando a composição interna dos arquivos é possível notar uma grande variação no número de variáveis disponibilizadas ao longo do tempo (Tabela 4). Além disso, a denominação das variáveis muda ao longo dos anos fazendo com que os utilizadores tenham que reeditar suas programações para se adequar a essas mudanças.

Essa variabilidade do número de variáveis investigadas e de sua composição reforça o que afirmou Silva (2003) em relação às constantes alterações e aperfeiçoamentos por quais passam as bases de dados, na tentativa de melhor refletir os fenômenos investigados.

Escolas	
Ano	**Variáveis**
2012	139
2013	140

Docentes	
Ano	**Variáveis**
2012	128
2013	128

[42] Atualmente, já é possível baixar os todos os anos da pesquisa em formato CSV.

2014	141
2015	166
2016	166

2014	126
2015	135
2016	132

Turmas	
Ano	Variáveis
2012	80
2013	80
2014	79
2015	88
2016	88

Matrículas	
Ano	Variáveis
2012	73
2013	85
2014	85
2015	93
2016	92

Tabela 4 - Número de variáveis segundo arquivo do Censo Escolar e ano.

4.3.2. Censo da Educação Superior.

Da mesma forma que o Censo Escolar, o Censo da Educação Superior é composto por cinco conjuntos, segundo a entidade investigada (Alunos, Cursos, Docentes, IES e Local de Oferta), cada um deles com somente um arquivo. Também originalmente distribuídos em arquivos no formato texto, passam a partir de 2013 a serem disponibilizados em formato CSV. Os metadados são disponibilizados também em formato de planilha Microsoft Excel.

Em relação à composição interna dos arquivos é possível ver que todos os arquivos sofrem alterações ao longo do tempo (Tabela 5), mesmo que em alguns casos, em número menor que o Censo Escolar.

Alunos	
Ano	Variáveis
2012	102
2013	110
2014	117
2015	119
2016	119

Cursos	
Ano	Variáveis
2012	74
2013	84
2014	95
2015	95
2016	96

Docentes	
Ano	Variáveis
2012	49
2013	49
2014	50
2015	50
2016	50

IES	
Ano	Variáveis
2012	41
2013	42
2014	41
2015	50
2016	50

Polos	
Ano	Variáveis
2012	14
2013	16
2014	16
2015	16
2016	16

Tabela 5 - Número de variáveis segundo arquivo do Censo IES e ano.

4.3.3. Censo Demográfico.

No presente trabalho iremos trabalhar somente com os dados da amostra do Censo Demográfico, pois possuem um maior número de variáveis relacionadas a temática educacional, além de possibilitarem o cruzamento das informações no âmbito dos indivíduos investigados[43]. As bases do Censo Demográficos são disponibilizadas em arquivos texto (*.TXT), em que cada variável é delimitada pela sua posição inicial e pelo o seu tamanho. Em 2010, último ano disponível, os dados eram organizados em quatro conjuntos de arquivos representando as diferentes entidades/temáticas investigadas pela pesquisa (Pessoas, Domicílio, Emigração e Mortalidade). Além disso, devido ao tamanho dos arquivos, os quatro arquivos são subdivididos segundo a unidade da federação (Estado) a qual pertence. Os metadados dos arquivos são disponibilizados em um arquivo do Microsoft Word (*.DOC), o que não facilita a sua extração e leitura por via de um script de programação.

Da mesma forma como ocorre com as outras bases trabalhadas, o Censo Demográfico apresenta alterações nos layouts dos arquivos ao longo dos anos, seja em relação ao número de variáveis disponibilizadas, seja em relação à nomenclatura utilizada a cada edição. Para fins de comparação, pois não trabalharemos com a edição de 2000 do Censo Demográficos, podemos ver as alterações sofridas na estrutura da pesquisa ao longo do tempo (Tabela 6).

Censo 2000			
Domicílios	**Pessoas**	**Emigração**	**Mortalidade**
57	109	-	-

Censo 2010			
Domicílios	**Pessoas**	**Emigração**	**Mortalidade**
48	138	14	14

Tabela 6 - Número de variáveis segundo arquivo do Censo Demográfico e ano.

Se por um lado há uma redução no número de variáveis relacionadas ao domicílio, com a exclusão de algumas questões relativas à posse determinados bens de consumo (Videocassete – VHS, por exemplo), por outro, há a um maior detalhamento das informações relacionadas aos indivíduos. Além das alterações na estrutura dos arquivos, o que é normal nesse tipo de inquérito – pois deve acompanhar as alterações observadas na sociedade – a alteração nas nomenclaturas das variáveis representa um desafio, já que obriga os utilizadores a compatibilizar manualmente as variáveis de dois ou mais anos da pesquisa.

[43] Os dados do universo são fornecidos somente em versões agregadas por setor de coleta censitária. Isso ocorre para respeitar a privacidade dos entrevistados, que em caso contrário poderiam ser identificados em algumas situações.

4.4. DESENVOLVIMENTO DA SOLUÇÃO.

Após analisar as necessidades levantadas pelo grupo de pesquisadores da instituição e avaliar as características das bases de dados que deverão ser utilizadas nessa primeira fase do projeto, optou-se por uma estratégia que possibilitasse aos utilizadores se dedicarem à construção dos indicadores e métricas, deixando a parte de processamento a mais automatizada possível. Para atingir esse objetivo, desenvolveu-se uma arquitetura ilustrada na Figura 6.

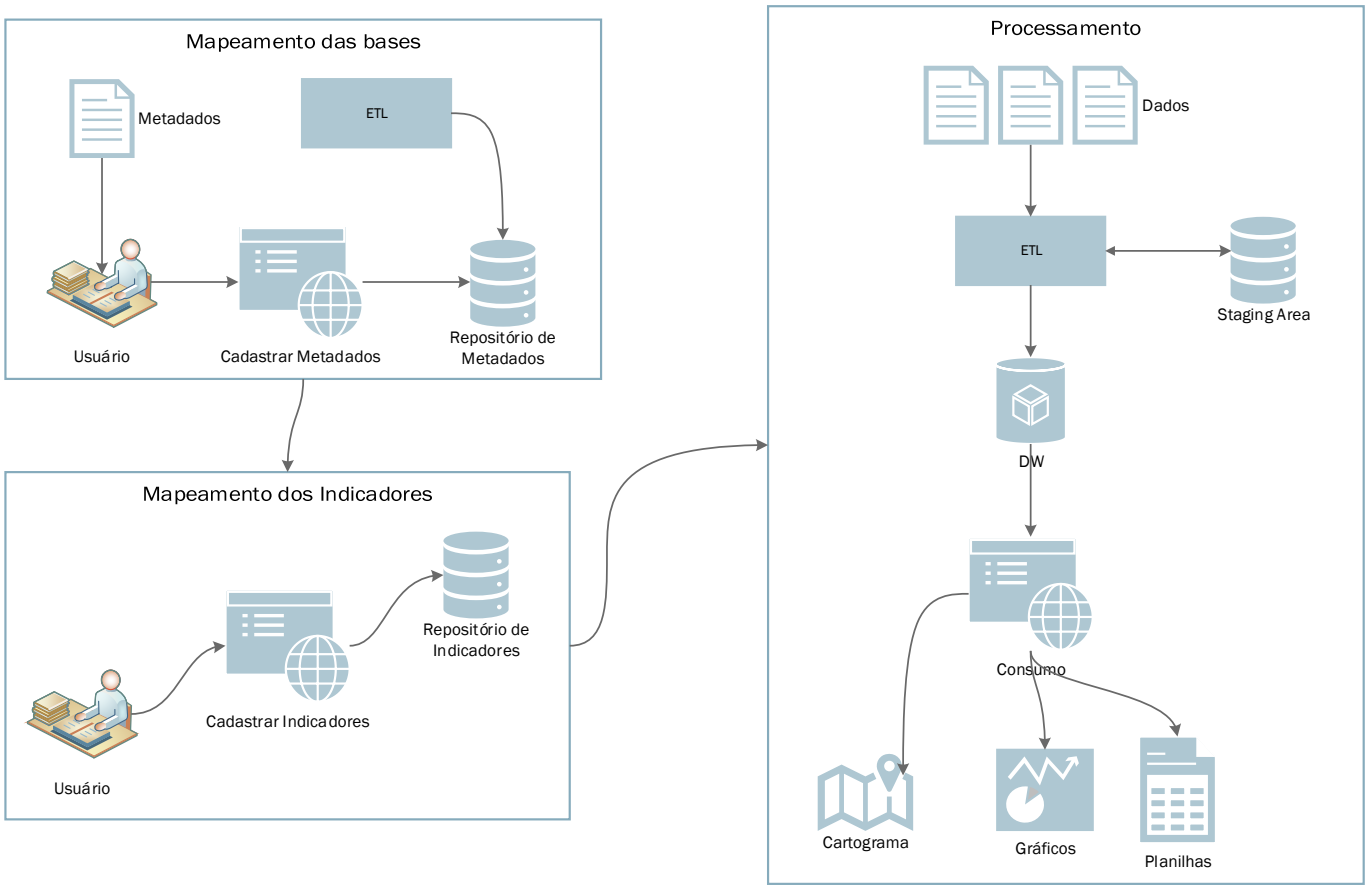

Figura 6 – Fluxo de construção de carga do DW

Como pode-se ver na Figura 6, o fluxo de carga da solução é composto por três macro etapas: Mapeamento das bases de Dados, Mapeamento dos Indicadores e Processamento dos Indicadores.

Para adequar a solução às características existentes na instituição, no que diz respeito à infraestrutura de tecnologia da informação, foi necessário segmentar a solução em mais de um tipo de tecnologia/linguagem de desenvolvimento.

Para as etapas de mapeamento, foram desenvolvidas interfaces gráficas baseadas em linguagem PHP[44] e banco de dados MySQL. A ideia dessas interfaces é possibilitar aos utilizadores um acesso

[44] A ferramenta ScriptCase foi utilizada para otimizar o desenvolvimento. Para maiores detalhes sobre esta ferramenta, consultar o sítio http://www.scriptcase.com.br/

rápido e facilitado às diferentes estruturas de mapeamento, de forma a otimizar ao máximo essas etapas.

Em relação ao processamento, não foi possível utilizar uma ferramenta de BI/ETL comercial, tal como o pacote SQL Server Integration Services (SSIS) – da Microsoft. No lugar desse, optou-se por desenvolver uma solução de ETL baseada na linguagem de programação Python. Por ser uma linguagem versátil de fácil aprendizado e utilização, Python, suporta múltiplos paradigmas de programação, o que torna ideal para diferentes perfis de desenvolvedores. Além disso, a linguagem suporta estruturas de dados complexas o que a torna adequada para esse tipo de desenvolvimento. Foram também utilizadas bibliotecas externas à biblioteca padrão da linguagem, tais como:

> **Pandas:** É uma biblioteca licenciada com código aberto que oferece estruturas de dados de alto desempenho e fácil utilização[45].
>
> **Pymysql:** É uma biblioteca que estabelece a conexão entre o Python e os banco MySQL. Ela foi utilizada para fazer a leitura dos repositórios de metadados e indicadores[46]
>
> **Sqlalchemy:** Também é uma biblioteca que estabelece a conexão entre o Python e diferentes bancos de dados. Ela foi utilizada para fazer a carga dos dados nas tabelas stages e fatos por apresentar um melhor desempenho quando combinada a biblioteca Pandas[47]

Para efeitos de comparação de desempenho da solução baseada em Python e Pandas com a disponível no pacote SSIS, foi realizado um teste simples de carga, ocasião em que a mesma instrução de carga foi executada em ambas as soluções. A massa de dados foi de 4.734.613 registros, referentes a um único arquivo do Censo Escolar de 2016. Enquanto o script Python/Pandas executou a carga em 11 minutos e 20 segundos, o SSIS realizou a carga em mais de 15 minutos. Mesmo parecendo muito para uma quantidade tão pequena de dados, deve-se ressaltar que, para os utilizadores, mais importa a transparência do processamento que a sua rapidez, na medida em que os utilizadores não são especializados em TI.

Por último, deve-se ressaltar que a parte de consumo das informações incorporadas ao modelo dimensional será realizada por meio da ferramenta Mapas da Cidadania, já existente na instituição e por esse motivo não será tratado nesse projeto.

4.4.1. Mapeamento das bases de Dados

Como foi dito na seção 4.3, há uma grande variabilidade das estruturas dos dados que torna o processo de mapeamento das bases de dados fundamental para o desenvolvimento da aplicação. Sem este, as etapas seguintes da solução não têm como ser realizadas.

O problema que se coloca nessa etapa de mapeamento é como tratar de forma eficiente uma variedade tão grande de formas de organização dos metadados. A ausência de um padrão, tal como

[45] https://pandas.pydata.org
[46] https://pypi.python.org/pypi/PyMySQL
[47] https://www.sqlalchemy.org

proposto pelo *Data Documentation Initiative* (DDI)[48], faz com que o utilizador tenha que realizar essa cadastragem, pelo menos em parte, de forma manual.

Inspirado nos campos existentes no catálogo produzido pelo *Inter-university Consortium for Political and Social Research* (ICPSR)[49], utilizado na metodologia do DDI, foi desenvolvida uma interface gráfica que possibilita aos utilizadores a cadastragem inicial das pesquisas/bases de dados.

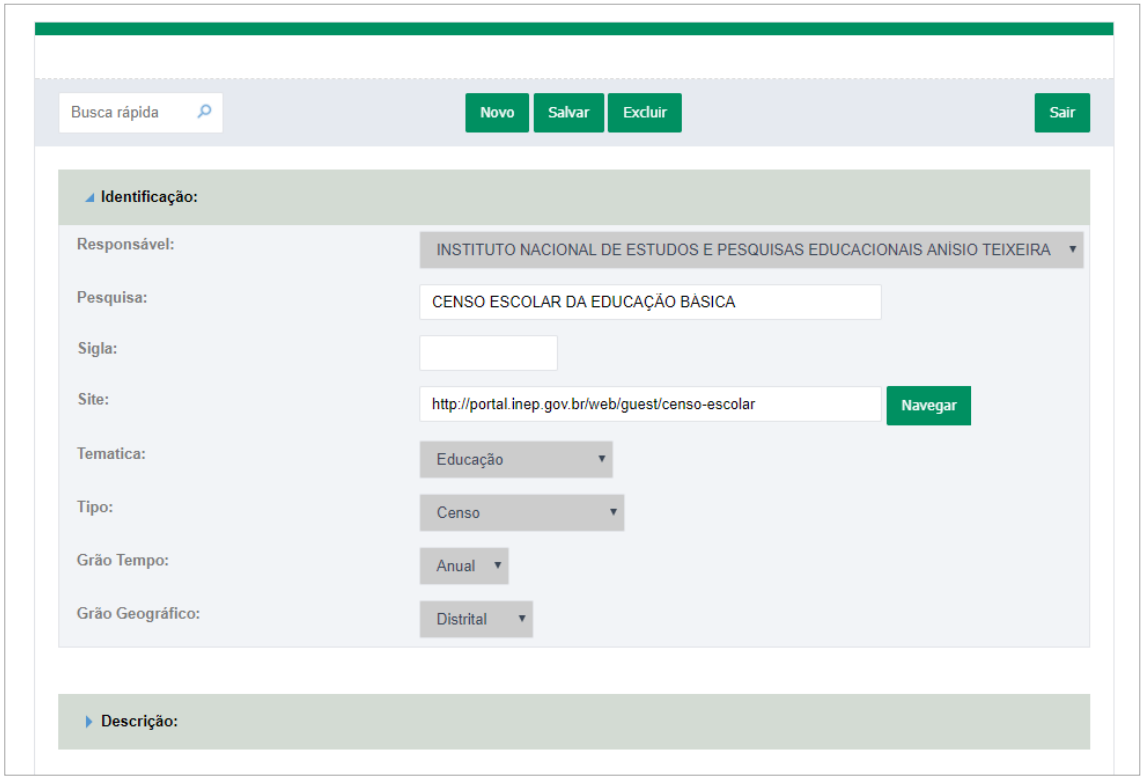

Figura 7 – Formulário de identificação da Base/Pesquisa

A função desse formulário inicial é identificar as informações básicas de uma pesquisa/base de dados, tais como o responsável pela sua produção ou divulgação, o seu nome, sitio onde pode ser encontrado, temática principal, tipo de recolha utilizada e, mais importante, a granularidade mínima dos níveis geográfico e temporal.

Após essa cadastragem inicial, os utilizadores passam a cadastragem das entidades investigadas. Esse tem por objetivo registrar informações que serão utilizadas no momento da criação e processamento dos indicadores propriamente ditos.

Além de registrar o nome da base de dados, são registrados o tipo de arquivo, a existência ou não de cabeçalho e o tipo de separador (se houver). A próxima seção do cadastramento registra o nome

[48] http://www.ddialliance.org/
[49] https://www.icpsr.umich.edu/icpsrweb/

e o caminho físico onde o arquivo (s) se encontra. Todas essas informações serão utilizadas para a leitura dos dados no momento da carga das tabelas stage.

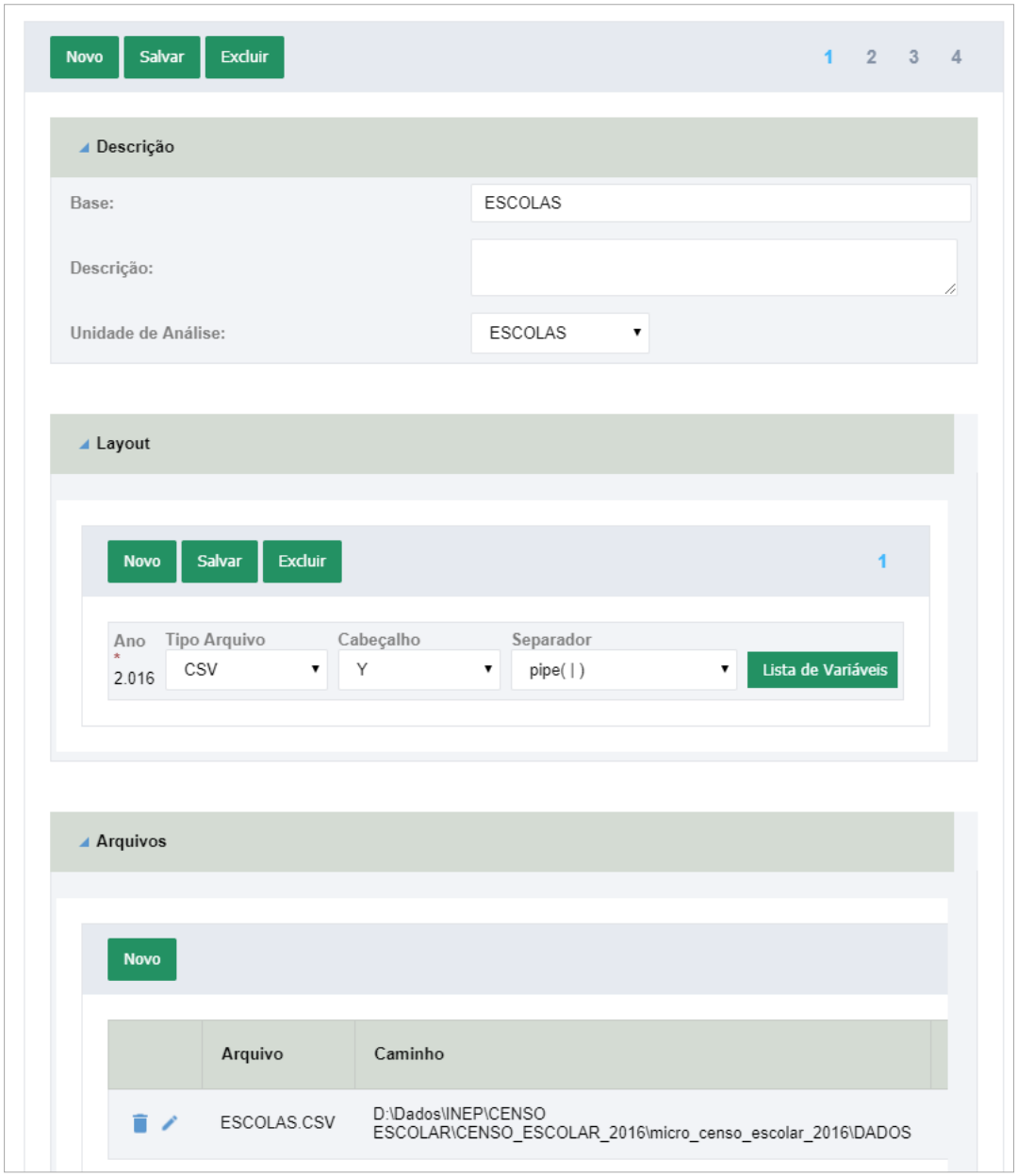

Figura 8 – Formulários para o cadastramento das características dos arquivos.

Assim que os arquivos tenham sido cadastrados, o utilizador deve fazer proceder a cadastragem das variáveis existentes neles. Nessa etapa, tendo em vista o grande número de variáveis que cada pesquisa pode ter (mais de 400 variáveis no caso do Censo Escolar, por exemplo), optou-se por desenvolver um script em Python para tender auxiliar o processo de carga dessas informações no banco de metadados.

Na Figura 9 temos uma descrição do fluxo de processamento do algoritmo. Desenvolvido em forma de função, o algoritmo deve ser receber os seguintes parâmetros: *nome da pesquisa e ano da pesquisa e caminho do arquivo de metadados*. Em relação ao arquivo de metadados, para facilitar a importação das informações, foi pensado um modelo no qual as informações devem ser apresentadas. Esse modelo apresenta o seguinte padrão:

Para bases em formato CSV:

Nome da Variável – identifica o nome da variável na base (Não aceita espaços).

Flag – identifica a função especial desempenhada pela variável no banco. Os códigos possíveis são:

Código	Função	Descrição
PK	Chave de Contagem	É a chave que identifica de forma unívoca um registro
FK	Chave Estrangeira	É a chave Estrangeira, que permite a ligação entre tabelas
GK	Chave Geográfica	É o identificador geográfico de menor grão
TK	Chave Temporal	É o identificador temporal de menor grão
DI	Dimensão Idade	É a variável que será utilizada para a compatibilização da dimensão idade
DC	Dimensão Raça-Cor	É a variável que será utilizada para a compatibilização da dimensão étnico-Racial
DS	Dimensão Gênero	É a variável que será utilizada para a compatibilização da dimensão Gênero/Sexo
W	Peso	É a variável que define a ponderação do registro (somente em casos de amostras)
M	Valor Monetário	É um valor monetário

Quadro 6 – Variáveis especiais

Descrição – é a descrição da variável

Tipo – identifica se a variável de tipo “string” ou “numérica”

Tamanho – é o número de caracteres que a variável ocupa no arquivo

Categorias – é o detalhamento do conteúdo da variável (livro de códigos)

Para as bases em formato TXT, os campos são os mesmos, acrescidos da informação da posição inicial e o tamanho da variável no arquivo. Essa informação é fundamental para fazer a leitura dos dados no caso dos arquivos posicionais.

Nos casos onde os metadados são disponibilizados em formatos tabulares (CSV ou XLS), essa transformação ocorre de maneira relativamente fácil, bastando aos utilizadores a padronização dos nomes e a exclusão de qualquer outra coluna que não necessite ser processada.

Para as bases de dados, em que os metadados são fornecidos em outros formatos, será necessária uma maior intervenção dos utilizadores para adequar as informações ao modelo preestabelecido. Não está descartada a hipótese de se desenvolver, no futuro, extratores para otimizar essa etapa do trabalho.

Na Figura 9 abaixo, há um detalhamento do fluxo utilizado para a importação das variáveis e do livro de códigos.

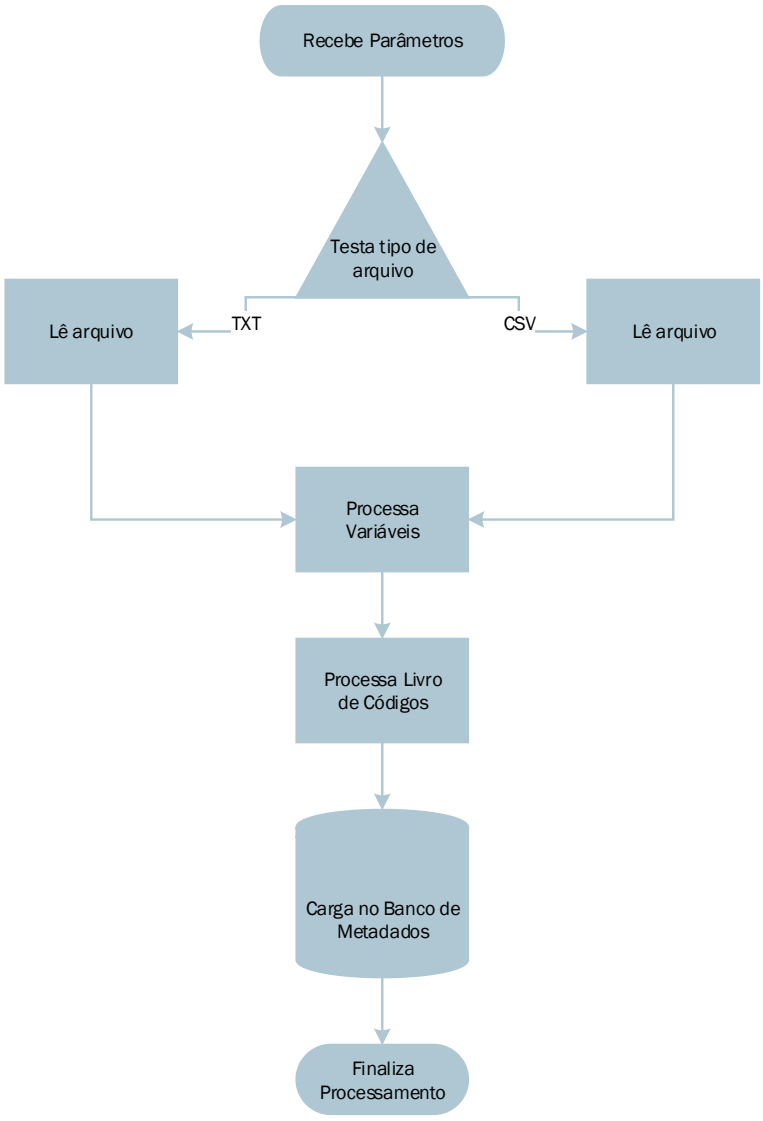

Figura 9 – Fluxo de Processamento do Livro de Códigos

O resultado da importação pode ser verificado na tela abaixo

	Valores	Nome	Papel	Tipo	Descrição	Inicial	Final	Tamanho	Missing
		NU_ANO_CENSO	Chave Temporal - TK	Numérico	Ano do Censo			4.0	
		CO_ENTIDADE	Chave de Contagem - PK	Numérico	Código da Escola			8.0	
		NO_ENTIDADE		String	Nome da Escola			100.0	
		CO_ORGAO_REGIONAL		String	Código do Órgão Regional de Ensino			5.0	
		TP_SITUACAO_FUNCIONAMENTO		Numérico	Situação de funcionamento			1.0	
		DT_ANO_LETIVO_INICIO		Data	Início do ano letivo			nan	
		DT_ANO_LETIVO_TERMINO		Data	Término (Previsão) do ano letivo			nan	
		CO_REGIAO		Numérico	Código da região geográfica			1.0	
		CO_MESORREGIAO		Numérico	Código da mesorregião			4.0	
		CO_MICRORREGIAO		Numérico	Código da microrregião			5.0	
		CO_UF		Numérico	Código da UF			2.0	

Figura 10 – Lista de Variáveis

Por último, o utilizador deve fazer o cadastro das ligações existentes entre diferentes arquivos (Chaves Primários e Chaves Estrangeiras), necessária para a realização de junções (joins) entre as bases.

	Base Pai	Chave Pai	Base Filho	Chave Filho
	ESCOLAS	CO_ENTIDADE	DOCENTES	CO_ENTIDADE
	ESCOLAS	CO_ENTIDADE	TURMAS	CO_ENTIDADE
	ESCOLAS	CO_ENTIDADE	MATRICULAS	CO_ENTIDADE

Figura 11 – Cadastramento das ligações

Abaixo, um esquema de como ficou o modelo de dados do mapeamento dos metadados.

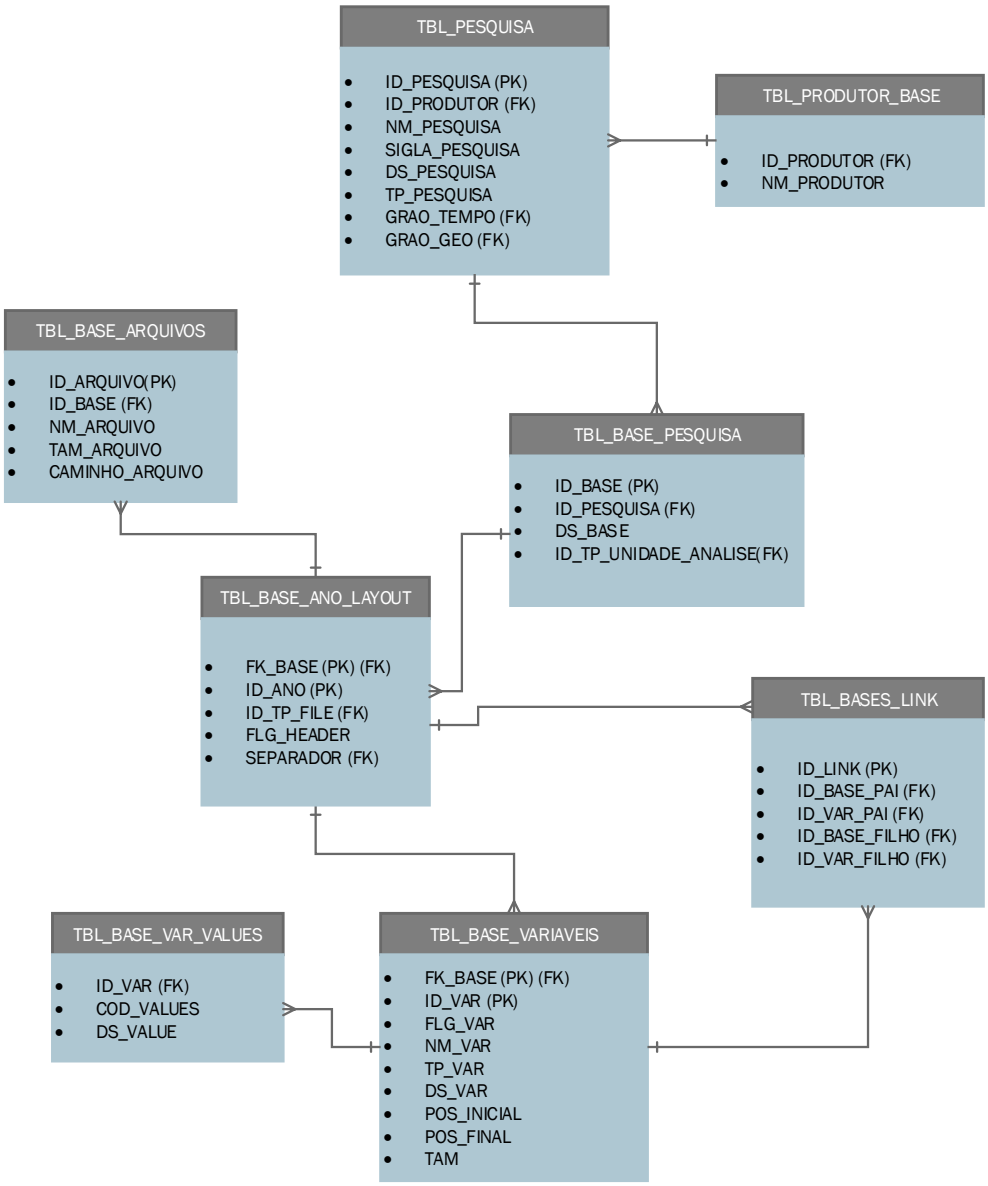

Figura 7 - Modelo de dados resumido, Banco de Metadados.

Apesar de ser um procedimento relativamente demorado e potencialmente tediosos, o mapeamento dos metadados só necessita ser realizado uma vez por edição da base de dados e é fundamental para o processamento dos indicadores. Após a realização do mencionado, os utilizadores estarão aptos para realizar a construção dos indicadores.

4.4.2. Mapeamento dos Indicadores

Ainda inspirado no catálogo produzido pelo ICPSR, foi desenvolvida uma interface gráfica para possibilitar aos utilizadores a documentação e construção dos indicadores que deverão constituir o banco dimensional.

A primeira parte do mapeamento é a definição do indicador que se deseja construir. Nessa parte deverão ser registradas as seguintes informações:

Nome do Indicador – Identifica o indicador

Descrição – Traz a descrição detalhada do indicador. Além disso, esse campo pode ser utilizado para incluir informações sobre a utilização dos indicadores e sua forma de leitura.

Tipo – Informa o tipo de medida a partir de que o indicador deverá ser construído. Nessa fase do projeto, são cinco os tipos possíveis: Percentual, Média, Valor Absoluto, Taxa e Razão (Rácio). No caso das Taxas, o usuário deverá ainda informar qual o fator de leitura do indicador (100.000, 10.000, 1000).

Unidade de Análise – Indica qual unidade de análise o indicador se refere. Pode ser "*Pessoal* " (Alunos, professores, pessoas, etc) ou "*Não pessoal*" (Escolas, municípios, etc).

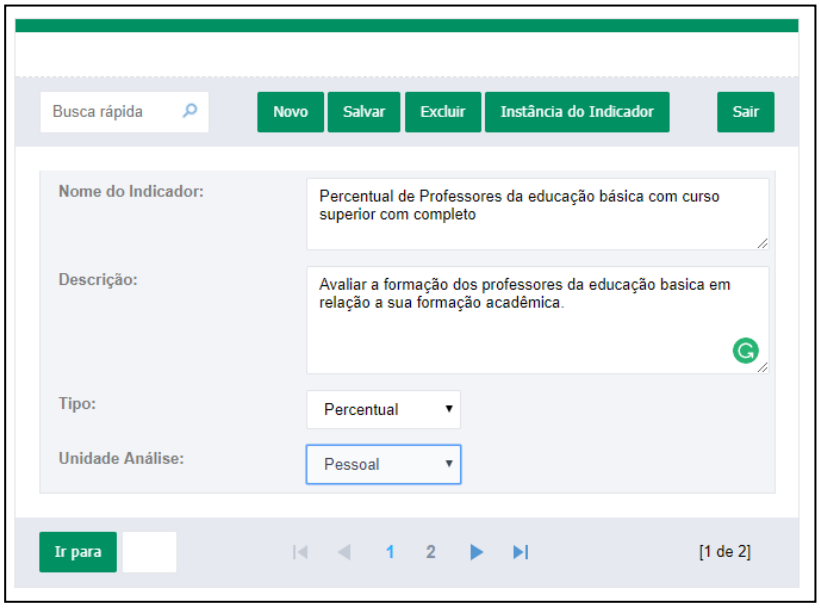

Figura 8 - Cadastro inicial dos Indicadores.

Além do caráter documental, essas informações deverão ser utilizadas para alimentar o modelo dimensional e definir parte do fluxo do processamento.

Feita a cadastragem a inicial, o utilizador deve passar a declaração das instâncias do indicador. Uma instância é compreendida como sendo um recorte temporal do indicador. Isso é necessário tendo em vista as alterações que os metadados podem apresentar ao longo do tempo. Por meio das instâncias é possível construir séries temporais com os indicadores, de forma a possibilitar análise longitudinais ao longo do tempo. É importante notar que a instância está relacionada a uma determinada edição da base de dados.

Após a declaração da instância que deverá ser trabalhada, o utilizador deve passar para o declaração das características do indicador propriamente dito.

As primeiras informações a serem cadastradas são as parcelas que compõem o indicador. Como foi dito na seção 4.2, por se tratar na sua maioria de métricas não-aditivas, é indicado que sejam armazenadas as parcelas que compõem o indicador e que os cálculos sejam realizados no momento da exibição. Podem ser três os tipos de parcelas: **Numerador** e **Denominador**, para os indicadores de tipo Percentual, Média, Taxa ou Razão (Rácio) ou **Parcela Única** para os indicadores do tipo valores absolutos. Além dessas informações, deverá ser cadastrado o nome da pesquisa que deverá ser utilizada para o processamento do indicador em questão e o tipo de totalização empregada.

Figura 9 - Cadastramento das parcelas

Após a definição das parcelas é necessário que o utilizador realize a definição de seus atributos dessas. Os atributos das parcelas são, em última análise, os filtros que se desejam atribuir aos dados dessas parcelas.

Para isso, em primeiro lugar, é necessário que sejam definidas que bases de dados serão utilizadas em cada uma das parcelas. Como a pesquisa/fonte que deve ser utilizada para a construção do indicador foi definida na criação das parcelas, o sistema só apresentará a bases relativas a essa fonte. Tendo escolhido a parcela e a base de dados desejada, o utilizador deverá começar a construir os filtros desejados.

Nessa etapa, será demandado dos utilizadores algum conhecimento de lógica de programação, principalmente no que diz respeito aos operadores de comparação (igual, diferente, maior que, maior ou igual que, menor que, menor ou igual que, contém, não contém) e lógicos (AND e OR).

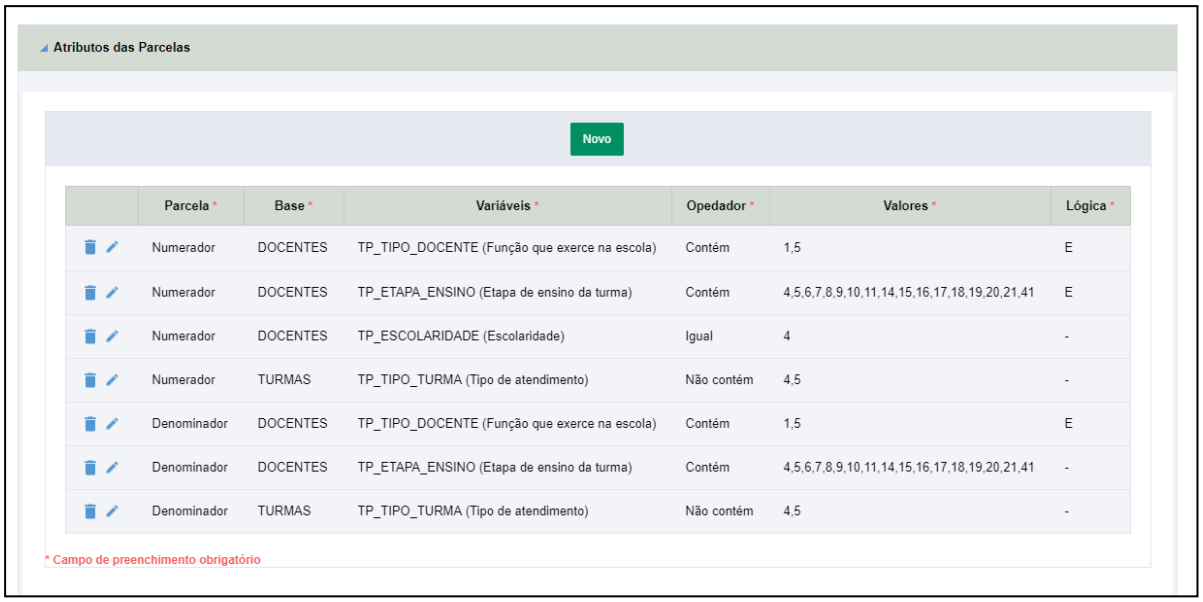

Figura 10 - Cadastramento dos atributos das parcelas

Dessa forma, é possível criar expressões lógicas envolvendo as variáveis selecionadas para cada parcela e os respetivos filtros, necessários para a construção dos indicadores. Apesar de relativamente complexa, essa etapa possibilita aos utilizadores a obtenção de poderosa ferramenta, já que permite a realização de filtros avançados nos dados e a construção de diferentes indicadores ou visões com os mesmos dados.

Nessa etapa são cadastradas as seguintes informações:

Parcela – identifica a parcela que se deseja utilizar,

Base – identifica a base de dados,

Variáveis – a partir da base de dados selecionada, são apresentadas as variáveis disponíveis,

Operador – são os operadores de comparação,

Valores – são os valores da variável que se deseja filtrar. No caso dos operadores “Contém” (IN) e “Não Contém” (NOT IN), que suportam valores múltiplos, esses devem ser separados por vírgulas,

Lógica – são os operadores lógicos. Esses devem ser utilizados quando se deseja incluir mais de uma variável em uma mesma consulta/filtro.

Tendo cadastrado os atributos das parcelas, a etapa seguinte requer a cadastragem das agregações desejadas para a construção dos indicadores. As agregações são entendidas como dimensões pelas quais se deseja poder visualizar os indicadores produzidos.

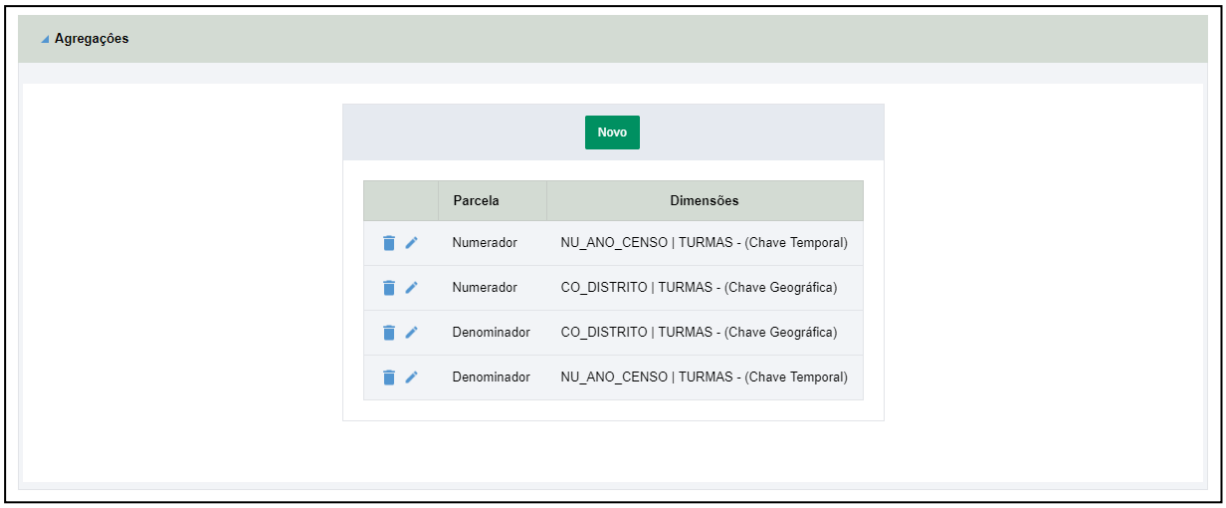

Figura 11 - Cadastramento das agregações

Como foi dito na seção 4.2, as dimensões temporal e geográfica são obrigatórias para todos os indicadores. As demais dimensões são de caráter opcional nos casos dos indicadores cuja unidade de análise são as pessoas.

Por último, é necessário que sejam cadastradas as unidades de totalização do indicador. As unidades de totalização são aquelas, indicadas na definição das parcelas, que vão definir como o indicador deverá ser agregado. Elas podem ser:

Contagem – representa a frequência absoluta de uma determinada variável (normalmente uma chave primária).

Ponderação – representa a soma da variável de ponderação (peso). Essa opção deve ser utilizada somente nos casos em que a base de dados representa uma amostra.

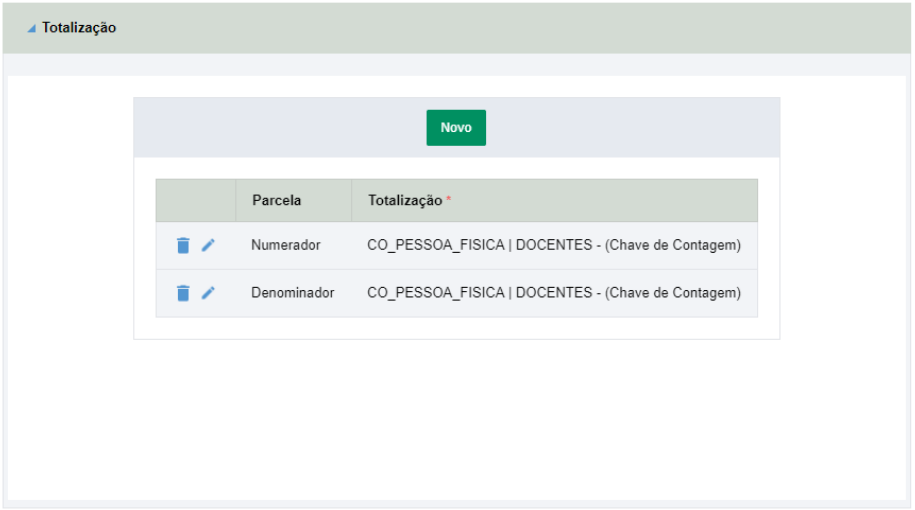

Figura 12 - Cadastramento da totalização

Na figura 13, podemos ver um modelo relacional simplificado da estrutura de armazenamento dos indicadores. Pela representação abaixo, podemos notar a existência do relacionamento desse modelo com os demais (metadados e dimensional). A interação entre modelo de indicadores e o

de metadados ocorre por meio do sistema de mapeamento dos indicadores. Já a interação com o modelo dimensional ocorre por meio do processamento do script Python, que será visto na próxima secção.

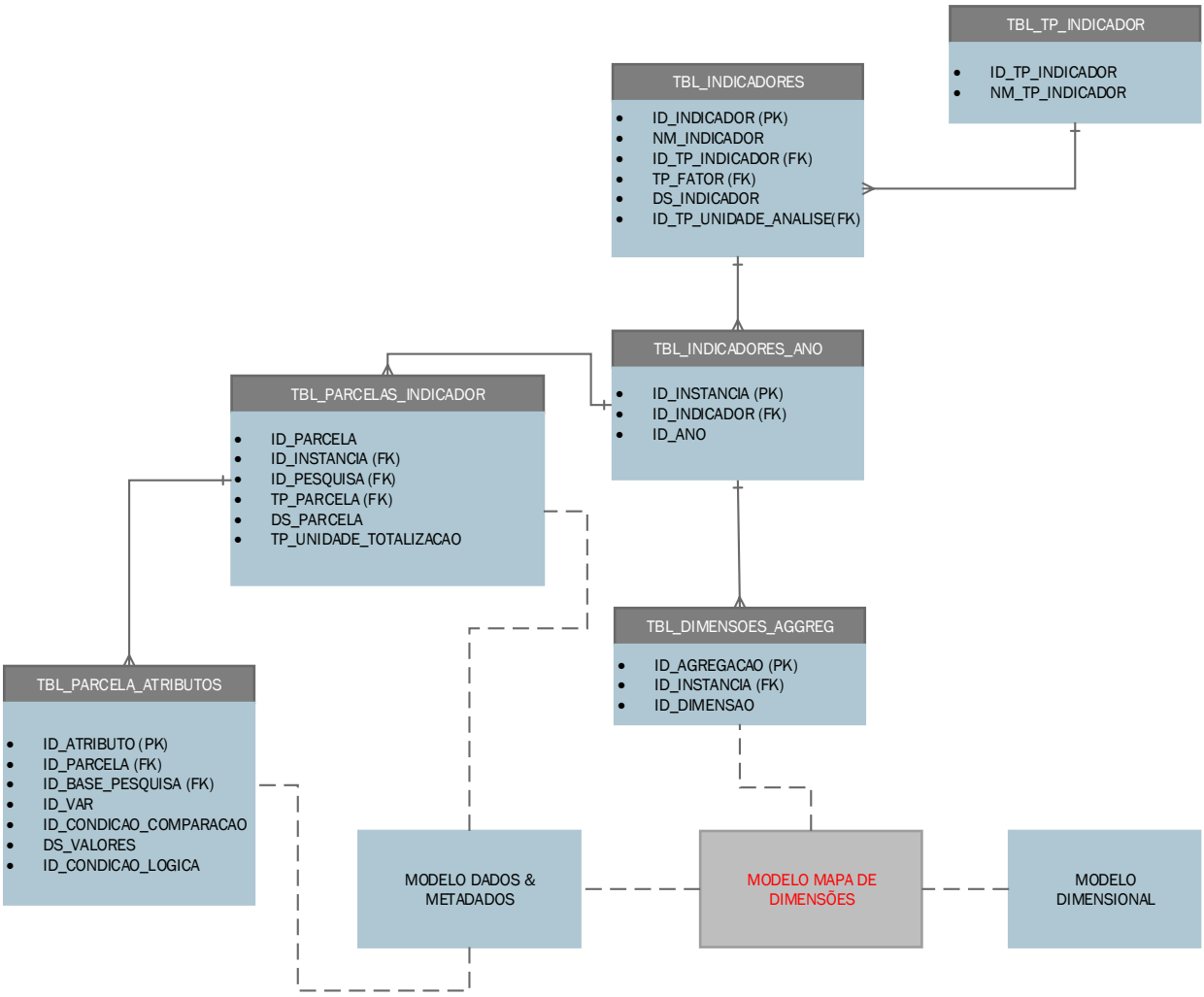

Figura 13 - Modelo de dados resumido, Banco de Indicadores

Por último, devemos ressaltar a importância da realização do mapeamento correto dos indicadores, seja em relação ao seu processamento, seja em relação a manutenção de uma documentação estruturada desses.

Além disso, vale reforçar a necessidade de que os utilizadores, mesmo desconhecendo procedimentos de programação de dados ou o funcionamento dos bancos de dados SQL, tenham uma maior afinidade com os procedimentos de construção dos indicadores e seus dados.

4.4.3. Processamento dos Indicadores.

Até o presente momento foram realizadas as etapas de mapeamento dos metadados e de cadastro dos indicadores. A partir de agora, essas informações serão utilizadas para o processamento dos dados, isso é o processo de ETL (Extract, Transform, Load), que permitirá que os dados brutos sejam transformados em informações.

Como foi dito anteriormente, a ideia dessa etapa era desenvolver uma solução não comercial, que possibilitasse a extração dos dados brutos de seus arquivos originais, a realização dos tratamentos e compatibilizações necessárias, bem como a carga final desses em um repositório. Para tanto, foram desenvolvidos scripts em linguagem Python que, baseados nas informações previamente cadastradas, fossem capazes de realizar tal conjunto de tarefas.

Para facilitar o processo de ETL, optou-se pela criação de um banco de dados que age como sendo uma "*Staging Area"*, que possibilitasse a criação de tabelas temporárias para a carga e tratamento dos dados brutos. Por se tratar de uma área temporária, há a garantia de que, durante o processamento de um determinado indicador, os dados presentes na "Staging" sejam sempre relativos a este, evitando assim a contaminação do ETL por "restos" de outros processamentos.

Após apagar as tabelas na "*Staging Area"*, o próximo passo é verificar a existência de indicadores na fila de processamento. Ao criar um novo indicador na interface gráfica, esse ganha um status de não processado (flg_proc = 0). Todo indicador com esse status será processado pelo algoritmo. Caso não haja indicadores nessa situação, o algoritmo interrompe o seu funcionamento, informando não haver informações para serem processadas.

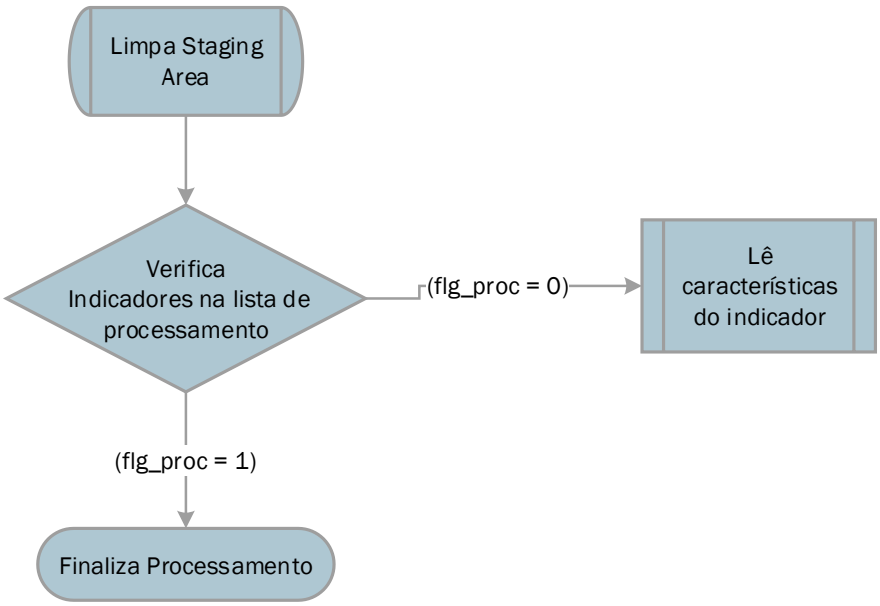

Figura 14 - Início do processo de ETL

Os indicadores na fila de processamento têm suas informações coletadas no banco de indicadores. As informações coletadas são: identificador do indicador, nome do indicador, tipo do indicador,

parcelas, base de dados utilizada, variáveis utilizadas e atributos (variáveis, operadores de comparação e lógica).

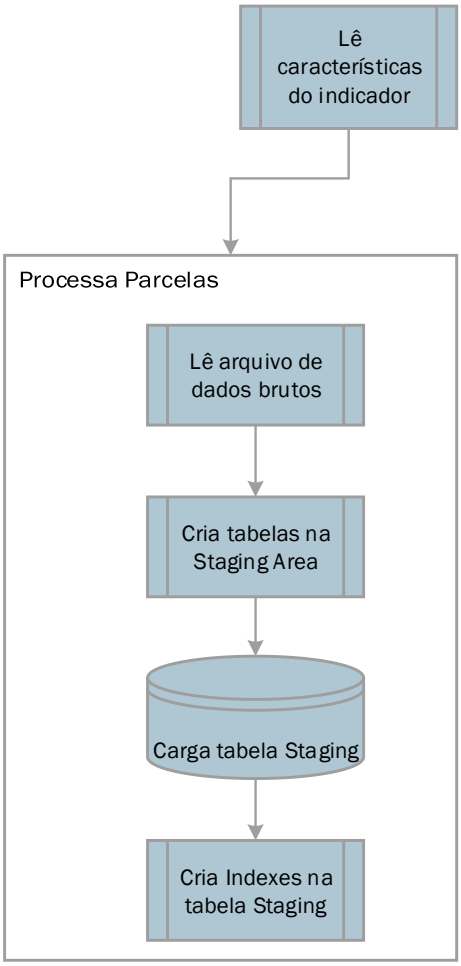

Figura 15 – Processamento das parcelas

Com essas informações o algoritmo identifica o as características das bases, tais como o tipo de arquivo (CSV ou TXT), existência ou não de cabeçalho e o tipo de delimitador de campos. Além de fazer a leitura dos dados, via Python/Pandas, o ETL cria o código DDL/SQL (Data Definition Language) para gerar as tabelas temporárias no banco Staging, conforme a estrutura dos dados existentes no arquivo original. Com os dados carregados em um dataframe Pandas e com as tabelas criadas, o algoritmo faz a carga dessas tabelas. Ao fim dessa etapa, o algoritmo cria os indexes dessas tabelas baseadas na função desempenhadas pelas as variáveis na base de dados (chave primária, chave estrangeira, etc). A importância da criação dos indexes será vista na etapa seguinte em que, dependendo da composição dos indicadores, será necessária a realização de junções (joins) de mais de uma base.

Tendo realizado essa etapa, o algoritmo de ETL passa para a transformação das variáveis e compatibilização dos valores originais das bases de dados com os das dimensões.

Isso ocorre por meio do mapeamento das dimensões, que é uma tarefa de fundamental importância para o bom funcionamento da solução. É por meio dela que as variáveis provenientes das diferentes bases de dados passam a se correlacionar com o modelo dimensional.

O mapeamento deverá ser realizado de dois modos, relacionados ao grau de complexidade das dimensões envolvidas.

O modo automático destina-se a fazer a transformação das dimensões com maior número de categorias e com maior grau de complexidade, que por isso mesmo trariam grande dificuldade para os utilizadores o seu tratamento manual. São exemplos desse tipo de variáveis as que envolvem a dimensões tempo e nível geográfico (em especial essa última). Essas dimensões são mínimas e obrigatórias para a criação de qualquer indicador.

Além dessas duas categorias, foi incluída a dimensão idade a esse grupo por apresentar uma grande variedade de formas de construção (faixas etárias), o que tornaria o trabalho do utilizador bem mais difícil.

Para que o mapeamento automático ocorra, é necessário que o utilizador indique, no momento do mapeamento das bases de dados, qual papel cada uma dessas variáveis assume no momento do processamento dos dados (seção 4.4.1).

Já o modo manual é destinado às demais dimensões do sistema. Essas dimensões não são necessariamente obrigatórias a todos os indicadores ou sua criação é passível de ser feita pelo utilizador.

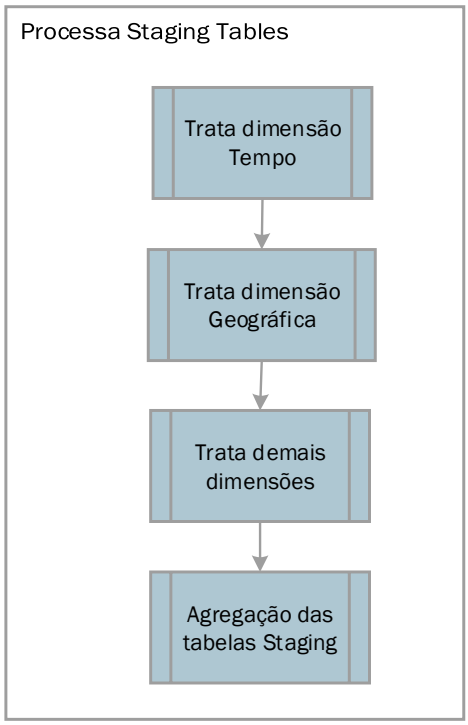

Figura 16 – Tratamento das dimensões e agregação

Com base nos valores compatibilizados entre os metadados e o modelo dimensional, o algoritmo faz a adequação dos valores de modo a garantir que estes possam ser carregados na tabela fatos ao final do processo.

O último passo dessa etapa é a realização das agregações e totalizações, que criarão as bases que serão posteriormente transferidas para o banco dimensional.

As informações mapeadas na etapa 4.4.2 (características de agregação e totalização) são utilizadas para a construção da instrução SQL que realizará a criação e carga das tabelas finais. É ainda nessa etapa que as junções (joins) existentes serão realizadas, a fim de garantir a correta construção dos indicadores.

Finalizada essa etapa sem erros, o algoritmo move o conteúdo das tabelas criadas para a tabela fatos, assim como as novas informações da dimensão Indicador. Por último, o algoritmo altera o status do indicador (flg_proc = 1) e passa para o próximo indicador da lista (caso haja).

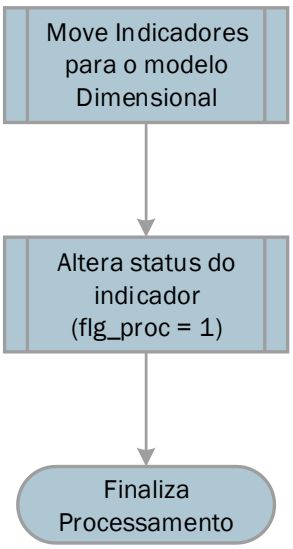

Figura 17 - Etapa final do processamento

Abaixo, na figura 18, o fluxo completo de processamento do ETL.

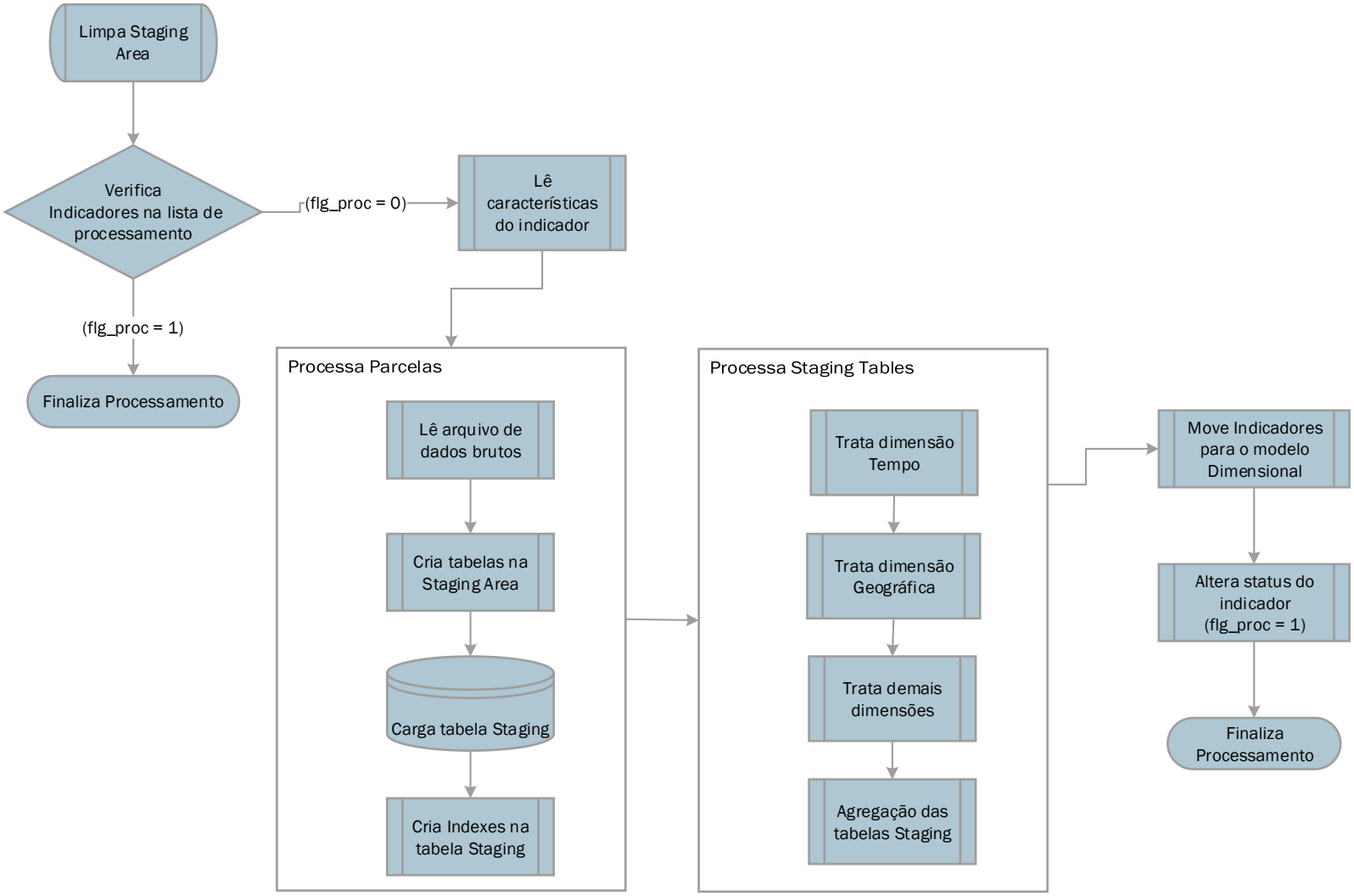

Figura 18 – Fluxo de processamento do ETL

4.4.4. Tratamento de Erros e Logging.

O tratamento e registros dos erros são etapas fundamentais de qualquer processamento de dados. Para tanto a linguagem Pyhton disponibiliza para isso uma biblioteca interna chamada “Logging”, que possibilita os registros das atividades de processamento de maneira rápida e fácil.

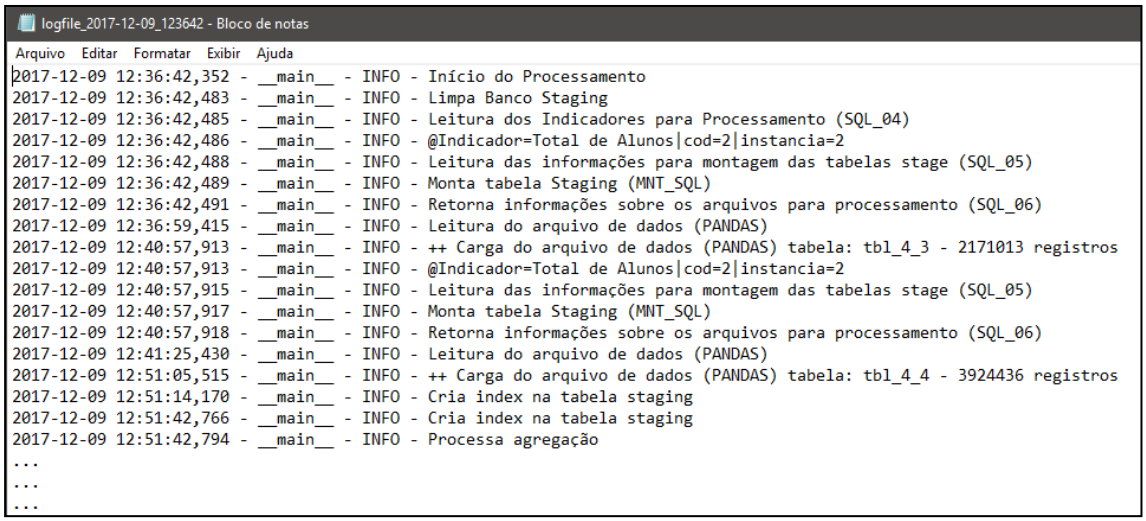

Figura 19 – Parte de um arquivo de logging

Além disso, a linguagem também disponibiliza uma estrutura que permite o tratamento das exceções.

```
try:
    df = pd.read_csv(caminho + '\\' + file , sep=separador ,
header=header_ , encoding='cp1252' , dtype=object , usecols=var)
    df = df.query(filtro.strip(' '))
    logger.info('Leitura do arquivo de dados (PANDAS)')
except:
    logger.error('Erro na leitura do arquivo de dados (PANDAS)')
```

Dessa forma, o utilizador pode acompanhar o processamento dos dados e ter mais facilidade de intervir, caso algum problema ocorra durante o processamento.

Em uma etapa futura do projeto, essas informações serão armazenadas em banco para um melhor controle do processo de criação dos indicadores.

4.5. MODELO DIMENSIONAL.

A construção do modelo dimensional do DW foi baseada no processo de levantamento de requisitos (secção 4.2) realizado com os usuários, na avaliação dos sistemas existentes na instituição e na análise das bases de dados que se deseja utilizar para a construção dos indicadores.

A partir desses levantamentos, foram definidas três diretivas:

- Devido às múltiplas granularidades (geográfica e temporal) observadas nas bases de dados, o modelo mais adequado deveria ser uma variante do "Snowflake Schema".
- Ainda relativo à múltipla granularidade, os factos seriam divididos em mais de uma tabela (Fact Constellation Schema).
- A descrição das métricas (indicadores/parcelas) seria transportada para uma dimensão. Dessa forma, o DW poderia ser utilizado pelos sistemas legados existentes.

Abaixo, descreveremos mais detalhadamente cada uma delas.

4.5.1. Snowflake Schema.

Segundo Kimball (Kimball & Caserta, 2015), enquanto o esquema em "Estrela" (Stat Schema) é caracterizado por ter as dimensões não normalizadas, no que ele chamou de "flat tables", o esquema em "Floco de Neve" (Snowflake Schema) se caracteriza por manter algum grau de normalização dentro da estrutura das dimensões. Essa técnica é especialmente útil quando se pretende destacar as estruturas hierárquicas existente nessas dimensões.

Apesar da sua utilização não ser recomendada por dificultar a compreensão e manutenção do modelo, o esquema em floco de neve possibilita que factos com diferentes granularidades sejam armazenados em um mesmo modelo dimensional. No presente caso a utilização do modelo em floco de neve permite que indicadores provenientes de diferentes bases de dados mantenham sua granularidade temporal ou geográfica mínima, sem a necessidade de se agregar tudo a uma unidade espaço-tempo comum.

Como foi visto na Tabela 3, apesar das bases de dados utilizadas na primeira fase do projeto apresentarem granularidade temporal no âmbito do ano, é desejável que o modelo já esteja preparado para suportar outras granularidades temporais (trimestres e mês). Da mesma forma, a menor granularidade espacial utilizada nessa fase será restrita aos níveis municipal e distrital, mas o modelo já deverá estar preparado para aceitar o nível estadual (pela agregação dos níveis municipal ou distrital) ou o nível de setor censitário (menor nível possível de desagregação).

Na Figura 20 apresentamos o detalhamento das dimensões tempo e geográfica.

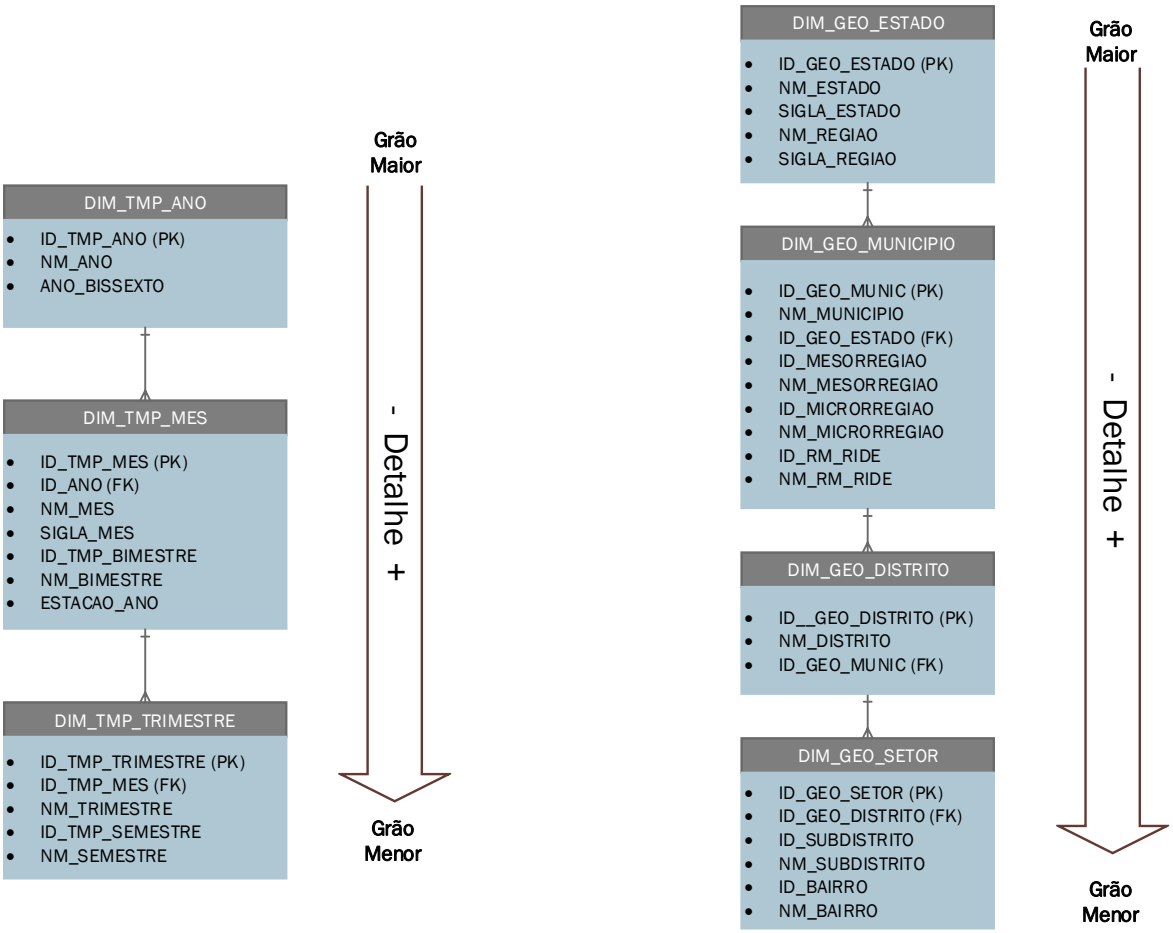

Figura 20 – Granularidade das dimensões tempo e geográfica

4.5.2. Fact Constellation Schema.

Como foi dito acima, as bases de dados utilizadas apresentam diferentes granularidades temporais e/ou espaciais e é de interesse da instituição manter essas informações desagregadas, sempre que possível. Se o esquema em floco de neve resolve essa questão em relação às dimensões, em pouco colabora para evitar que diferentes granularidades sejam armazenadas em uma mesma tabela factos, conforme é recomendado pela literatura (Group Kimball, 2013).

Para evitar que isso ocorra, temos que dispor de múltiplas tabelas de factos em um mesmo modelo dimensional. Denominado como Constelação de Factos (Fact Constellation), esta arquitetura permite que diferentes tabelas de facto sejam explicitamente atribuídas às dimensões que lhe são relevantes. Dessa forma, alguns factos estarão associados a um determinado nível de dimensão enquanto outros factos são atribuídos com um nível de dimensão mais detalhado. Apesar de ser uma solução flexível, ela apresenta o inconveniente de ser mais difícil de gerenciar e manter.

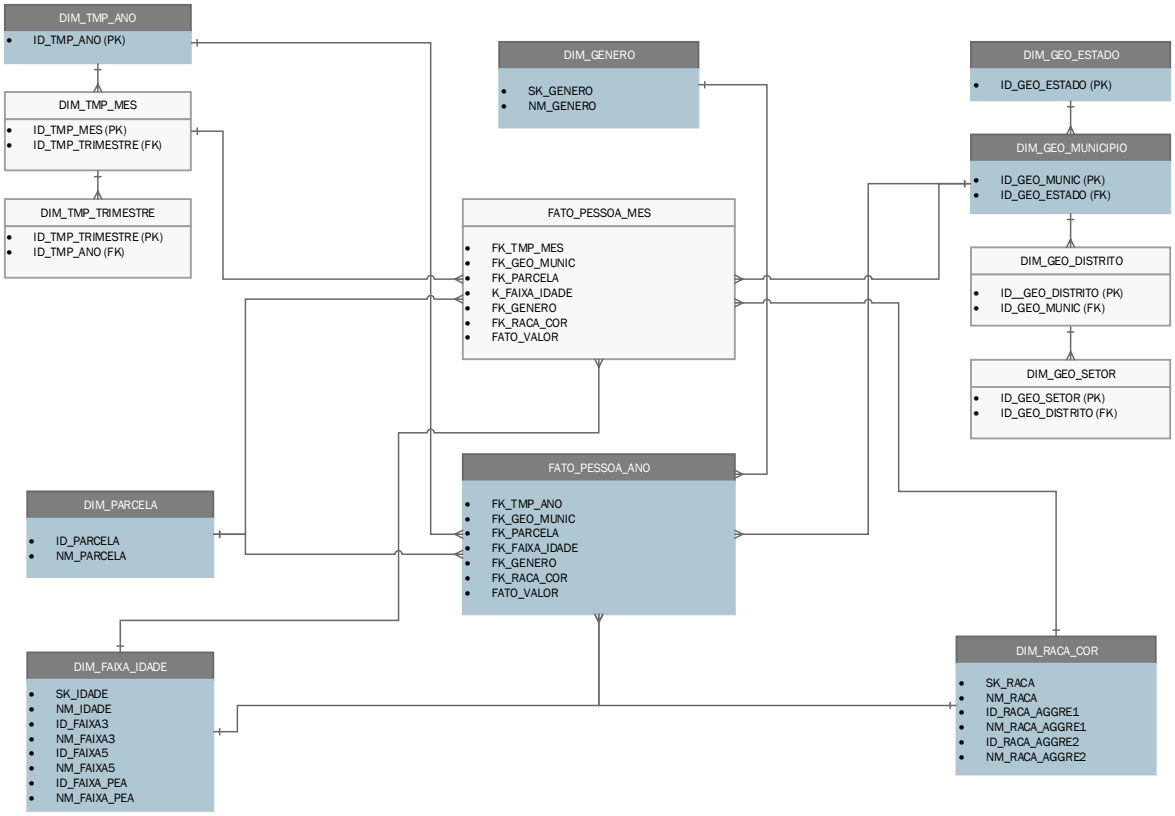

Figura 21 – Constelação de factos.

Como se pode ver na Figura 21, é possível construir um esquema de constelação de factos dividindo o esquema original em esquemas com mais tabelas, em que cada uma delas descreve factos em outro nível de hierarquias de dimensão tempo e geográfica.

4.5.3. Transposição das Métricas para o nível dimensional.

Conforme foi constatado durante o levantamento de requisitos, o IBASE já dispõe de um sistema de recuperação de informações estatísticas denominado “Mapas da Cidadania”. Para que a presente solução fosse compatível com este, foi necessário que a descrição dos indicadores fosse associada a uma dimensão de modo explícito. Assim, ao invés de um novo indicador representar uma nova coluna na tabela factos, essa passa a ser um conjunto de registros, referenciados por uma dimensão específica.

Dessa forma, é possível atribuir aos indicadores a taxonomia utilizada dentro da instituição de maneira fácil e rápida.

DIM_ACAO			
id_acao	**nm_acao**	**id_classe**	**nm_classe**
1	Cidadania Ativa	11	Cidadania Vivida
1	Cidadania Ativa	12	Cidadania Garantida
1	Cidadania Ativa	13	Cidadania Percebida
1	Cidadania Ativa	14	Cidadania e Ação
2	Conjunto de Direitos	21	Direitos Coletivos
2	Conjunto de Direitos	22	Diretos Sociais e Econômicos e Culturais
2	Conjunto de Direitos	23	Direitos Civis e Políticos

DIM_AREA		
id_classe	**id_area**	**nm_area**
21	2011	Direito à Cidade
21	2012	Direitos Ambientais
22	2021	Direito à Saúde
22	2024	Direito à Renda
22	2022	Direito à Educação
22	2023	Direito ao Trabalho
23	2032	Direito à Igualdade e Diversidade
23	2031	Direitos à Participação

DIM_INDICADOR		
id_classe	**id_indicador**	**nm_indicador**
22	20	Garantia de qualificação dos docentes

Tabela 7 - Taxonomia dos Indicadores.

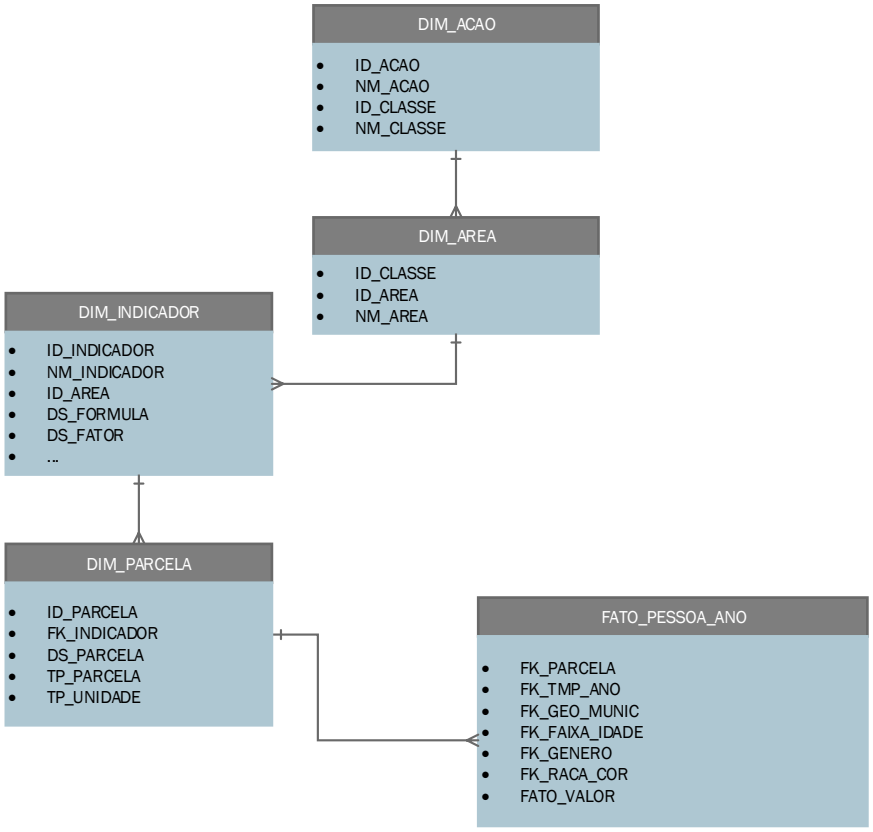

Figura 22 – Taxinomia aplicada ao modelo de dados.

4.5.4. Detalhamento das dimensões do projeto.

Nessa primeira fase do projeto, trabalharemos com um conjunto restrito de dimensões. Apesar do seu pequeno número, as dimensões escolhidas apresentam alto poder explicativo das desigualdades presentes na sociedade brasileira.

4.5.4.1. Dimensão Gênero

A dimensão gênero contém as descrições das classificações de gênero (sexo) das pessoas.

Campos:

ID_SEXO – chave candidata da tabela,

NM_SEXO – descrição do gênero/sexo,

SLG_SEXO – abreviatura do gênero/sexo.

id_sexo	nm_sexo	slg_sexo
1	Masculino	M
2	Feminino	F
99	Ignorado	I
999	Não se aplica	S

Tabela 8 – Dimensão Gênero.

O código 99 representa a situação em que o gênero/sexo não é conhecido. Já o código 999 representa a ausência dessa informação para o indicador. Esses códigos são válidos para todas as dimensões.

4.5.4.2. Dimensão Idade

A dimensão idade contém as descrições das classificações etárias, em suas diferentes agregações, permitindo analisar diferentes recortes etários ao longo da vida do indivíduo.

Campos:

ID_IDADE – chave candidata da tabela,

FAIXA_01 – faixa etária de maior desagregação. Permite detalhar as diferentes etapas da vida, segundo coortes etárias.

FAIXA_02 – faixa etária de desagregação média (4 em 4 anos).

FAIXA_03 – faixa etária de desagregação baixa (9 em 9 anos).

FAIXA_CICLO – Faixa agregada segundo ciclos etários (Crianças, adolescentes, jovens, adultos e idosos).

FAIXA_PIA – Faixa agregada segundo população em idade ativa

id_idade	faixa_01	faixa_02	faixa_03	faixa_ciclo	faixa_pia
0	Zero anos	0 a 3 anos	0 a 9 anos	Crianças	Não Economicamente Ativa
1	1 a 3 anos	0 a 3 anos	0 a 9 anos	Crianças	Não Economicamente Ativa
2	4 a 5 anos	4 a 5 anos	0 a 9 anos	Crianças	Não Economicamente Ativa
3	6 anos	6 a 9 anos	0 a 9 anos	Crianças	Não Economicamente Ativa
4	7 anos	6 a 9 anos	0 a 9 anos	Crianças	Não Economicamente Ativa
5	8 a 9 anos	6 a 9 anos	0 a 9 anos	Crianças	Não Economicamente Ativa
6	10 anos	10 a 14 anos	10 a 19 anos	Fase inicial da adolescência	Não Economicamente Ativa
7	11 a 13 anos	10 a 14 anos	10 a 19 anos	Fase inicial da adolescência	Não Economicamente Ativa
8	14 anos	10 a 14 anos	10 a 19 anos	Fase inicial da adolescência	Não Economicamente Ativa
9	15 a 17 anos	15 a 17 anos	10 a 19 anos	Fase final da adolescência	Economicamente Ativa
10	18 anos	18 a 24 anos	10 a 19 anos	Fase final da adolescência	Economicamente Ativa
11	...	...	...	...	...

Tabela 9 – Dimensão Idade

4.5.4.3. Dimensão Étnico-racial

A dimensão gênero contém as descrições das classificações de étnico-raciais e suas agregações.

Campos:

ID_COR – chave candidata da tabela,

RACA_COR_01 – classificação étnico-racial de maior desagregação. É condizente com a classificação de cinco categorias utilizada pelo instituto de estatística brasileiro (IBGE),

RACA_COR_02 – classificação étnico-racial de média desagregação (4 categorias),

RACA_COR_03 – classificação étnico-racial de baixa desagregação (3 categorias),

RACA_COR_04 – classificação étnico-racial dicotômica (2 categorias).

id_cor	raca_cor_01	raca_cor_02	raca_cor_03	raca_cor_04
1	Branca	Branca	Branca	Branca
2	Preta	Negra	Negra	Não-Branca
3	Parda	Negra	Negra	Não-Branca
4	Amarela	Amarela	Outras	Não-Branca
5	Indígena	Indígena	Outras	Não-Branca
99	Ignorado	Ignorado	Ignorado	Ignorado
999	Não se aplica	Não se aplica	Não se aplica	Não se aplica

Tabela 10 – Dimensão étnico-racial

4.5.4.4. Dimensão Indicador

A dimensão indicador contém as descrições dos indicadores e suas características (Figura 22). Diferentemente das demais dimensões, ela é uma dimensão normalizada e não tem a função de agregação, servindo somente para identificação e descrição dos indicadores.

DIM_PARCELA

Campos:

ID_PARCELA – chave candidata da tabela

DS_PARCELA – descrição da parcela

TP_PARCELA – descrição do tipo de parcela (numerador, denominador ou única)

TP_UNIDADE – define o tipo de unidade de análise

DIM_INDICADOR

Campos:

ID_INDICADOR – chave candidata da tabela,

NM_INDICADOR – nome do indicador,

DS_INDICADOR – descrição do indicador,

TP_INDICADOR – tipo de indicador,

ID_AREA – chave da taxonomia,

DS_FORMULA – descreve a fórmula de construção do indicador,

DS_FATOR – descreve o fator de multiplicação utilizado na fórmula de criação do indicador,

FLG_ATIVO – variável indicativa para a exibição do indicador.

Como foi dito na secção 4.5.1, a estrutura dos indicadores é composta por mais duas tabelas dentro do "floco de neve". "**Dimensão Área**" e a "**Dimensão Ação**" representam a taxonomia utilizada na instituição e possibilitam a recuperação da informação segundo essa classificação (Tabela 7).

4.5.4.5. Dimensões Geográficas

A dimensão geográfica contém as descrições da divisão territorial brasileira de forma reduzida para atender as necessidades do presente projeto. Como já dito na secção 4.5.1, ela é uma dimensão normalizada em suas hierarquias (Figura 20).

Ela é composta pelas seguintes tabelas:

DIM_GEO_ESTADO

Colunas:

ID_GEO_ESTADO – chave candidata da tabela,

COD_ESTADO – código IBGE para os estados,

NM_ESTADO – nome do estado,

SLG_ESTADO – sigla do estado,

COD_REGIAO – código IBGE para a macrorregião,

NM_REGIAO – nome da macrorregião,

SLG_REGIAO – sigla da macrorregião.

DIM_GEO_MUNICIPIO

Colunas:

ID_GEO_MUNIC – chave candidata da tabela,

FK_GEO_ESTADO – chave estrangeira da tabela estado,

COD_MUNIC – código IBGE para os municípios,

NM_MUNIC – nome dos municípios

COD_MESO – código IBGE para as mesorregiões geográficas,

NM_MESO – nome das mesorregiões geográficas,

COD_MICRO – código IBGE para as microrregiões geográficas,

NM_MICRO – nome das microrregiões geográficas,

COD_RM_RIDE – código IBGE para as regiões integradas de desenvolvimento,

NM_RM_RIDE – código IBGE para as regiões integradas de desenvolvimento,

DIM_GEO_DISTRITO

Colunas:

ID_GEO_DISTRITO – chave candidata da tabela,

FK_GEO_MUNIC – chave estrangeira da tabela município,

COD_DISTRITO – código IBGE para distritos,

NM_DISTRITO – nome do distrito,

DIM_GEO_SETOR

Colunas:

ID_GEO_SETOR – chave candidata da tabela,

FK_GEO_DISTRITO – chave estrangeira da tabela município,

COD_SETOR – código IBGE para setores censitários,

COD_SUBDISTRITO – código IBGE para os subdistritos,

NM_SUBDISTRITO – nome do subdistrito,

COD_BAIRRO – código IBGE para os bairros,

NM_BAIRRO – nome do bairro

SITUACAO_SETOR – situação do setor censitário

TP_SETOR – tipo de setor censitário,

4.5.4.6. Dimensões Temporais

A dimensão tempo contém as descrições das divisões temporais observadas nas fontes de dados. Como já dito na secção 4.5.1, ela é uma dimensão normalizada em suas hierarquias (Figura 20).

Ela é composta pelas seguintes tabelas:

DIM_ANO

Colunas:

ID_ANO – chave candidata da tabela,

FLG_BISSEXTO – indicativo de ano bissexto.

DIM_MES

Colunas:

ID_MES – chave candidata da tabela,

FK_ANO – chave estrangeira da tabela ano,

MES – número do mês

NM_MES – descrição do mês

SLG_MES – sigla do mês

DIM_TRIMESTRE

Colunas:

ID_TRIMESTRE – chave candidata da tabela,

FK_MES – chave estrangeira da tabela mês,

TRIMESTRE – número do trimestre

NM_TRIMESTRE – descrição do trimestre

SLG_TRIMESTRE – sigla do trimestre

4.5.4.7. Constelação de Fatos

Como foi dito na secção 4.5.2, o modelo de dados adotado foi o "Snowflake Schema" com a adição de múltiplas tabelas de factos (Fact Constellation) por melhor se adequar às necessidades do negócio.

Na Figura 23, é possível se ver parte do modelo dimensional para a granularidade municipal, desenvolvida para a presente solução.

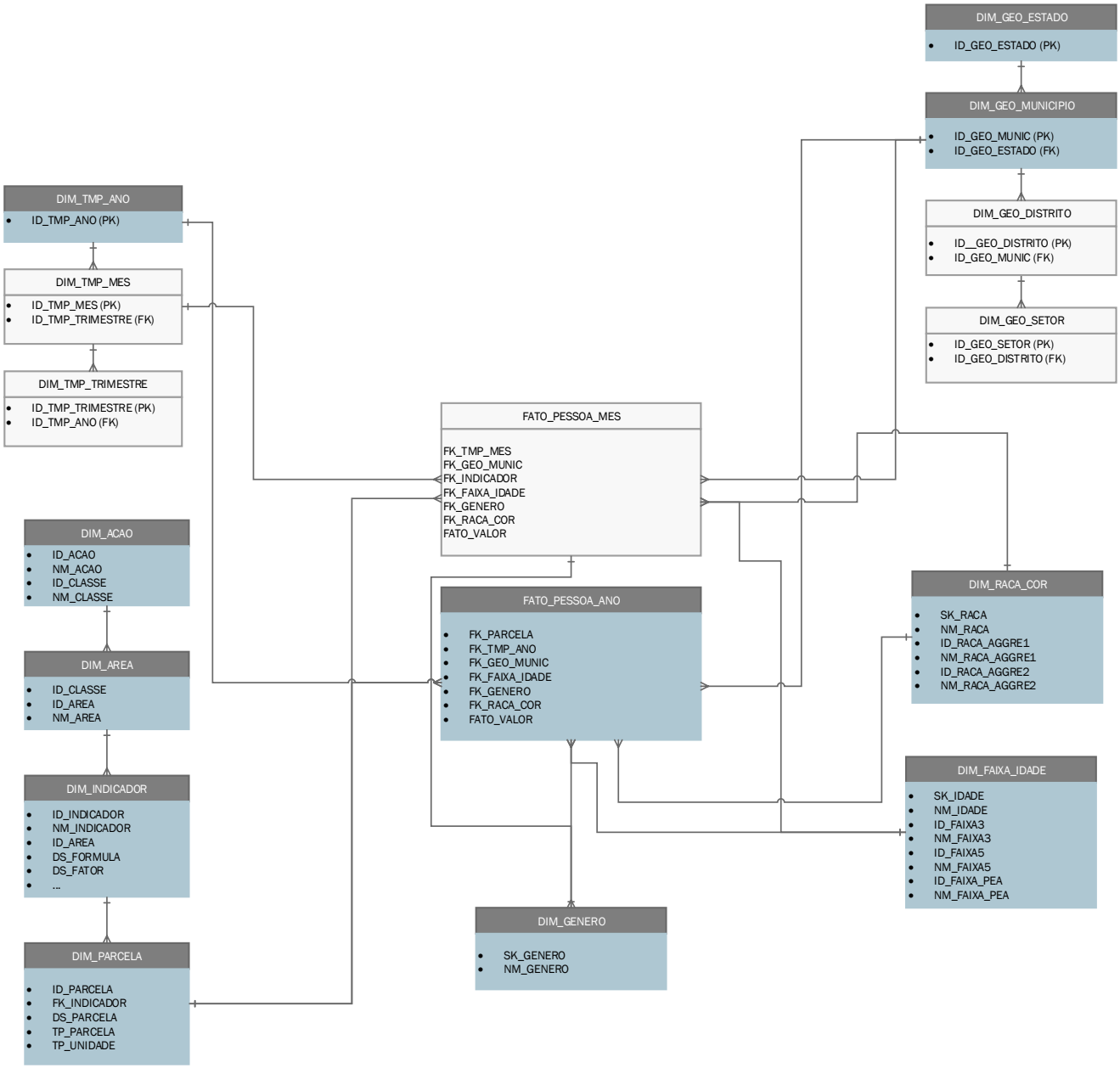

Figura 23 – Secção do modelo dimensional para os fatos de granularidade municipal

5. CONSUMO DA INFORMAÇÃO

Como foi dito na introdução, o desenvolvimento de soluções para o consumo da informação não faz parte do escopo do presente projeto. Com a finalidade de exemplificar as potencialidades do modelo dimensional desenvolvido, será apresentado abaixo um exemplo produzido a partir da ferramenta Power BI da Microsoft.

O Power BI é um conjunto de ferramentas para a visualização de dados que possibilita a criação de visões de dados e dashboards com diferentes graus de complexidade. Por meio da ferramenta é possível aceder diretamente aos repositórios de dados, realizar transformações, criar tabelas e gráficos de forma rápida e fácil.

Uma das características do Power BI é a possibilidade de importar os modelos dimensionais diretamente para a ferramenta (Figura 24).

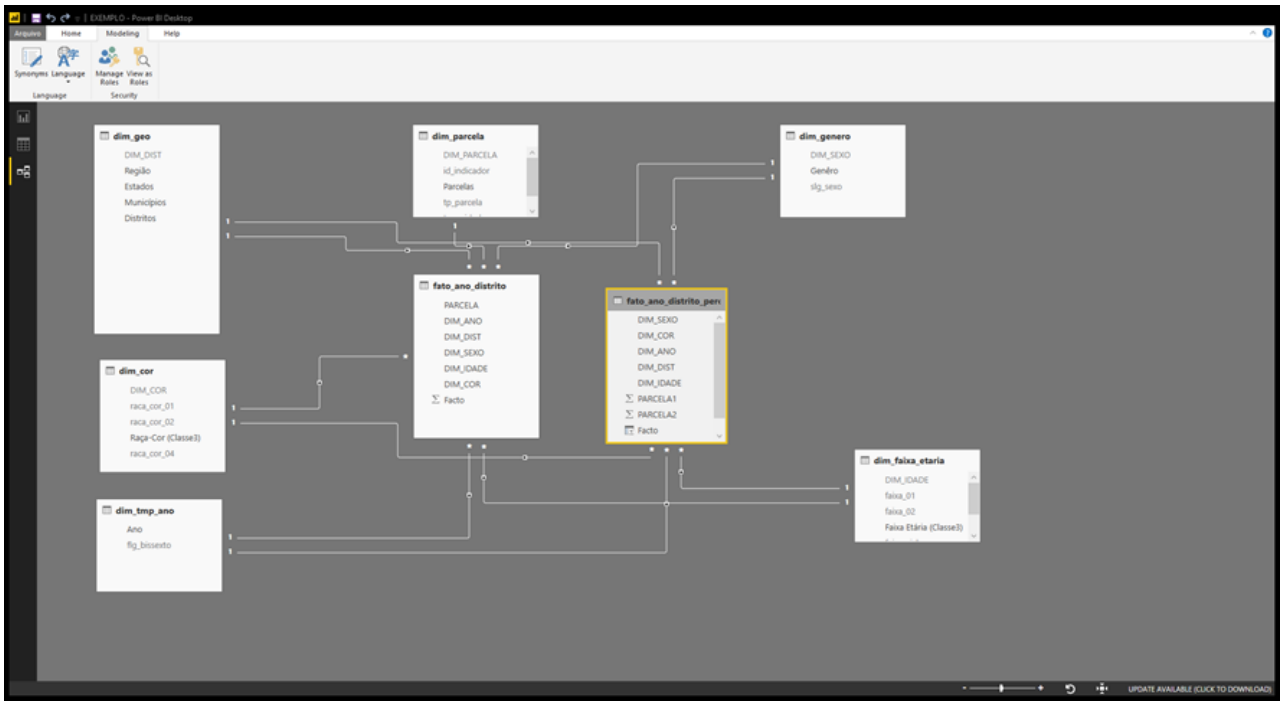

Figura 24 – Modelo Dimensional importado para o Power BI

o Power BI permite a realização de diferentes tipos de transformações nos dados importados. Um exemplo é a "desnomalização" das dimensões do modelo original (Snowflake Schema) em uma nova tabela "flat" e a criação de hierarquias, possibilitando assim a navegação direta entre os diferentes níveis de uma determinada dimensão (drill down e drill up).

Em termos de visualização de dados, o Power BI disponibiliza mais de duas dezenas de tipos de gráficos, cartogramas e tabelas que permitem a criação de visualizações interativas.

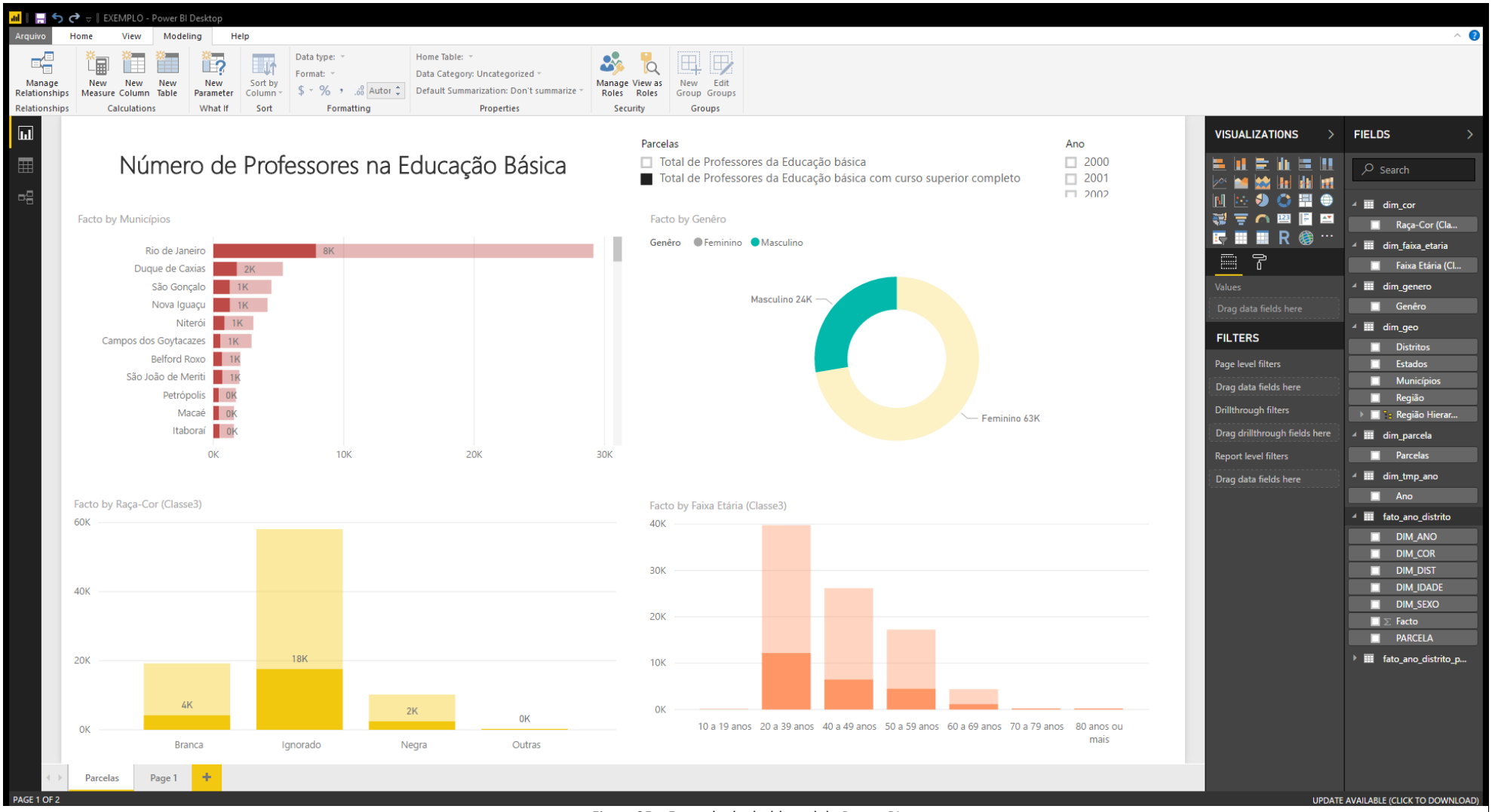

Figura 25 – Exemplo de dashboard do Power BI

6. CONCLUSÕES

O presente trabalho procurou demostrar as possibilidades de utilização das técnicas de Data Warehousing para a construção de uma solução de BI, de baixo custo, que utilizasse dados públicos abertos.

Com quase uma década de existência, o Movimento dos Dados Abertos tem demostrado a sua importância para a promoção da transparência pública, o combate à corrupção e o desenvolvimento da cidadania e, mesmo que tenha sofrido reveses ao longo do caminho, tem consolidado sua presença ao redor do mundo.

Apesar de sua relevância, a difusão dos dados públicos ainda apresenta problemas de utilização relacionados, na sua maior parte, à ausência de padrões para a divulgação dos dados e metadados. O que se observa, na prática, é que o desejo de disponibilizar as informações para a sociedade é maior que a capacidade dos órgãos produtores em uniformizar os processos de divulgação dessas mesmas informações. Mesmo tendo em mente que a função fim da maioria desses órgãos não é a divulgação de informações, ações simples como a unificação do formato dos metadados, de modo a torná-los totalmente legíveis por sistemas informatizados, representaria um grande avanço na direção de tornar mais fácil a vida dos utilizadores.

Nesse sentido, a utilização de técnicas de Data Warehousing tem demostrado ser de grande valia para o consumo dos dados públicos abertos. O avanço no desenvolvimento de ferramentas de extração, armazenamento e visualização de dados, que tem ocorrido nas últimas décadas, possibilitou a um maior número de pessoas ou instituições trabalhar com tais informações.

Mesmo assim, a grande diversidade de formatos e de temáticas – principalmente quando se trabalha as informações sob o enfoque multidisciplinar das Ciências Sociais – faz com que o nível de reutilização das soluções seja inferior ao observado em outras áreas do conhecimento.

A solução encontrada no presente projeto foi o desenvolvimento de uma plataforma para o mapeamento dos metadados das diferentes pesquisas, que possibilitasse a sua uniformização e permitisse a leitura e extração dos dados de forma mais rápida e fácil. Apesar da vantagem evidente do processo de mapeamento, já que possibilita uma maior reutilização das bases, este se mostrou mais difícil quando as informações dos metadados eram distribuídas em formatos não editáveis, tais como os arquivos PDF.

Em relação ao processamento dos dados para a produção de indicadores sociais, a solução desenvolvida, isto é, a combinação de uma interface para a cadastragem dos indicadores com a utilização de scripts em linguagem Python, mostrou-se bastante positiva, por possibilitar aos utilizadores finais uma maior autonomia em relação às formas anteriores de utilização (contratação de profissionais de TI para o processamento dos dados). Essa solução não elimina a necessidade de se conhecer as metodologias de cálculo e construção dos indicadores desejados, nem o conhecimento sobre os metadados utilizados. Dessa forma, a presente solução não elimina a necessidade de se dispor de utilizadores qualificados e especialistas, facilitando somente a relação desses com o ferramental técnico disponível.

6.1. OBJETIVOS REALIZADOS

O objetivo inicialmente proposto para o presente trabalho foi a construção de um repositório de dados públicos. A necessidade de utilizar ferramentas não comerciais, que reduzissem os custos de implementação da solução, mostrou-se um desafio durante o processo de desenvolvimento. Felizmente, a escolha da linguagem Python foi correta e possibilitou alcançar os objetivos desejados.

Durante o processo de desenvolvimento, foram cumpridas as seguintes etapas:

- Identificação dos requisitos dos utilizadores,
- Identificação das soluções existentes,
- Planejamento e execução da presente solução,

Mesmo se tratando da continuação de um projeto já existente, devemos pensar a solução apresentada como um passo dado na direção de dar autonomia ao IBASE no processo de produção e divulgação dos indicadores.

6.2. LIMITAÇÕES

Como já foi dito no decorrer do presente trabalho, a solução apresentada não descarta a necessidade de utilizadores capacitados e especialistas durante o processo de produção dos indicadores. Estes são e serão sempre necessários no processo de construção e na análise dessas métricas.

Ademais, devido à complexidade dos modelos de dados desenvolvidos, o processo de manutenção da solução deve ser planejado com atenção para se evitar a interrupção do funcionamento.

Além disso, não foi possível no presente projeto fazer a integração dos scripts Python com as soluções PHP em um mesmo ambiente de produção.

6.3. FUTUROS TRABALHOS

Espera-se, em breve, ser possível fazer a integração dos scripts Python com as soluções PHP em um mesmo ambiente de produção. Além de possibilitar uma melhor manutenção, tal integração possibilitaria o desenvolvimento de novas funcionalidades que tornariam a experiencia de utilização mais transparente.

Além disso, pretendemos construir uma nova solução de visualização e consumo das informações armazenadas que possibilite a utilização de técnicas de BI e OLAP.

Por último, já estamos a desenvolver a desenvolver de um repositório de dados municipais, criado por meio da presente solução, que possibilite a realização de análises mais avançadas e mineração de dados.

7. REFERÊNCIAS BIBLIOGRÁFICAS

Alexopoulos, C., Zuiderwijk, A., Charapabidis, Y., Loukis, E., & Janssen, M. (2014). Designing a Second Generation of Open Data Platforms: Integrating Open Data and Social Media. In M. Janssen, H. J. Scholl, M. A. Wimmer, & F. Bannister (Eds.), *Electronic Government: 13th IFIP WG 8.5 International Conference, EGOV 2014, Dublin, Ireland, September 1-3, 2014. Proceedings* (pp. 230–241). Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-662-44426-9_19

Attard, J., Orlandi, F., Scerri, S., & Auer, S. (2015). A systematic review of open government data initiatives. *Government Information Quarterly*, *32*(4), 399–418. https://doi.org/10.1016/j.giq.2015.07.006

B, R. B., Hanbury, A., Piroi, F., Haas, M., Berger, H., Lupu, M., & Dittenbach, M. (2015). Data Management Technologies and Applications, *178*(November), 45–61. https://doi.org/10.1007/978-3-319-25936-9

Berndt, D. J., Hevner, A. R., & Studnicki, J. (2003). The Catch data warehouse: Support for community health care decision-making. *Decision Support Systems*, *35*(3), 367–384. https://doi.org/10.1016/S0167-9236(02)0114-8

Berro, A., Megdiche, I., & Teste, O. (2015). A Content-Driven ETL Processes for Open Data. *Advances in Intelligent Systems and Computing*, *312*(April), 29–40. https://doi.org/10.1007/978-3-319-10518-5_3

Bertot, J. C., Jaeger, P. T., & Grimes, J. M. (2010). Using ICTs to create a culture of transparency: E-government and social media as openness and anti-corruption tools for societies. *Government Information Quarterly*, *27*(3), 264–271. https://doi.org/10.1016/j.giq.2010.03.001

Bonamino, A., & Franco, C. (2013). Avaliação e política educacional: o processo de institucionalização do SAEB. *Cadernos de Pesquisa*, (108), 101–132.

BRASIL. (2004). *SINAES - BASE PARA UMA NOVA PROPOSTA DE AVALIAÇÃO DA EDUCAÇÃO SUPERIOR*. *SINAES*. Brasília.

Capistrano, D., Cirotto, A. C., Nascimento, C. D., & Silva, J. (2016). Produção de Estatísticas Educacionais em Perspectiva Comparada. *Estatística E Sociedade*, (4).

Castro, M. H. G. de. (2000). Sistemas nacionais de avaliação e de informações educacionais. *São Paulo Em Perspectiva*, *14*(1), 121–128. https://doi.org/10.1590/S0102-88392000000100014

Cintrão, L. P., & Bizelli, J. L. (2013). Sistema de Monitoramento e Avaliação de Programas Sociais: revisitando mitos e recolocando premissas para sua maior efetividade na gestão. *Revista Brasileira de Monitoramento E Avaliação*, *5*, 48–59. Retrieved from http://aplicacoes.mds.gov.br/sagirmps/ferramentas/docs/RBMAs/RBMA_5.pdf

Coletta, R., Castanier, E., Valduriez, P., Frisch, C., Ngo, D., & Bellahsene, Z. (2012). Public data integration with WebSmatch. *Proceedings of the First International Workshop on Open Data - WOD '12*, (c), 5. https://doi.org/10.1145/2422604.2422606

Dawes, S., Vidiasova, L., & Parkhimovich, O. (2016). Planning and designing open government data programs: An ecosystem approach. *Government Information …*, *33*, 15–27. Retrieved from http://www.sciencedirect.com/science/article/pii/S0740624X1630003X

Diniz, E. H., Barbosa, A. F., Junqueira, A. R. B., & Prado, O. (2009). O governo eletrônico no Brasil: perspectiva histórica a partir de um modelo estruturado de análise. *Revista de Administração Pública*, *43*(1), 23–48. https://doi.org/10.1590/S0034-76122009000100003

Dos Santos Brito, K., Costa, M. A. S., Garcia, V. C., & De Lemos Meira, S. R. (2014). Experiences integrating heterogeneous Government open data sources to deliver services and promote transparency in Brazil. *Proceedings - International Computer Software and Applications Conference*, (475743), 606–607. https://doi.org/10.1109/COMPSAC.2014.87

dos Santos Brito, K., da Silva Costa, M. A., Garcia, V. C., & de Lemos Meira, S. R. (2014). Brazilian government open data. In *Proceedings of the 15th Annual International Conference on Digital Government Research - dg.o '14* (pp. 11–16). New York, New York, USA: ACM Press. https://doi.org/10.1145/2612733.2612770

Freitas, R. K. V. de, & Dacorso, A. L. R. (2014). Inovação aberta na gestão pública: análise do plano de ação brasileiro para a Open Government Partnership. *Revista de Administração Pública*, *48*(4), 869–888. https://doi.org/10.1590/0034-76121545

Gil, N. de L. (2009). A produção dos números escolares (1871-1931): contribuições para uma abordagem crítica das fontes estatísticas em História da Educação. *Revista Brasileira de História*, *29*(58), 341–358.

Golfarelli, M., & Rizzi, S. (1999). Designing the Data Warehouse: Key Steps and Crucial Issues. *Journal of Computer Science and Information Management*, *2*(3), 1–14. https://doi.org/10111241900

Gonzalez-Zapata, F., & Heeks, R. (2014). The multiple meanings of open government data: Understanding different stakeholders and their perspectives. *Government Information Quarterly*, *32*(4), 441–452. https://doi.org/10.1016/j.giq.2015.09.001

Gracioso, L. D. S. (2003). Disseminação de informações estatísticas no Brasil: práticas e políticas das agências estaduais de estatística. *Ciência Da Informação*, *32*(2), 69–76. https://doi.org/10.1590/S0100-19652003000200008

Gracioso, L. de S. G. (2004). Produção e disseminação da informação estatística brasileira: uma análise qualitativa. *Perspectivas Em Ciência Da Informação*, *9*(1), 34–47.

Group Kimball. (2013). Kimball Dimensional Modeling Techniques, 1–24.

Gruman, M. (2012). Lei de Acesso à Informação: notas e um breve exemplo. *Revista Debates*, *6*(3), 97. https://doi.org/1982-5269

Guimarães, C. B. dos S. (2014). Parceria para Governo Aberto e Relações Internacionais: oportunidades e desafios. Retrieved from http://hdl.handle.net/11449/121891

Guo, Y., Guo, Y., Tang, S., Tang, S., Tong, Y., Tong, Y., … Yang, D. (2006). Triple-Driven Data Modeling Methodology in Data Warehousing: A Case Study. *Proc. of the 9th ACM International Workshop on Data Warehousing and OLAP*, 59–66. https://doi.org/10.1145/1183512.1183524

Heise, A., & Naumann, F. (2012). Integrating open government data with stratosphere for more transparency. *Journal of Web Semantics*, *14*, 45–56. https://doi.org/10.1016/j.websem.2012.02.002

Inmon, W. H. (2005). *Building the data warehouse*. (R. Elliott, Ed.) (3ª Edition). New York: John

Wiley & Sons, Inc.

Jannuzzi, P. D. M. (2001). *Indicadores sociais no Brasil: conceitos, fonte de dados e aplicações.* Campinas: Alínea.

Jardim, J. M. (2004). A construção do e-gov no Brasil : configurações político-informacionais. *Cinform*, 1–25. Retrieved from http://www.cinform.ufba.br/v_anais/artigos/josemariajardim.html

Kimball, R. (1998). The Data Warehouse Lifecycle Toolkit: Expert Methods for Designing, Developing, and Deploying Data Warehouses, 771. Retrieved from http://books.google.com/books?hl=fr&lr=&id=abEwJJLewDAC&pgis=1

Kimball, R., & Caserta, J. (2015). *The Data Warehouse ETL Toolkit*. *The effects of brief mindfulness intervention on acute pain experience: An examination of individual difference* (Vol. 1). https://doi.org/10.1017/CBO9781107415324.004

Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit The Definitive Guide to Dimensional Modeling* (3th Editio). New York: John Wiley & Sons, Inc.

Lacerda, L. L. V. de, Ferri, C., & Duarte, B. K. da C. (2016). SINAES: avaliação, accountability e desempenho. *Avaliação: Revista Da Avaliação Da Educação Superior (Campinas)*, *21*(3), 975–992. https://doi.org/10.1590/S1414-40772016000300015

List, B., Bruckner, R. M., Machaczek, K., & Schiefer, J. (2002). A comparison of data warehouse development methodologies case study of the process warehouse. *Database and Expert Systems Applications*, 203–215. https://doi.org/10.1007/3-540-46146-9_21

Matheus, R., & Ribeiro, M. M. (2014). Open Data in the Legislature: The Case of São Paulo City Council. Retrieved from http://www.opendataresearch.org/content/2014/665/open-data-legislature-case-são-paulo-city-council

McHugh, B. (2013). *Beyond Transparency: Open Data and the Future of Civic Innocation*. *Beyond Transparency - Open Data and Future of Civic Innovation*. https://doi.org/10.1017/CBO9781107415324.004

Meijer, a J., Curtin, D., & Hillebrandt, M. (2012). Open government: connecting vision and voice. *International Review of Administrative Sciences*, *78*(1), 10–29. https://doi.org/10.1177/0020852311429533

Moura, R., & Barion, M. I. (2006). INSTITUCIONALIZAÇÃO DE REGIÕES METROPOLITANAS : qual o sentido ?, (i), 129–143.

OGP. (2011). *Plano de ação brasileiro – versão em português brazilian action plan – english version*. Retrieved from www.opengovpartnership.org/countries/brazil

Oliveira, L. A. P. de, & Simões, C. C. da S. (2005). O IBGE e as pesquisas populacionais. *Revista Brasileira de Estudos Populacionais*, *22*(2), 291–302. https://doi.org/10.1590/S0102-30982005000200007

Paiva, E. De, Revoredo, K., & Baião, F. (2016). DW-CGU: Integração dos Dados do Portal da Transparência do Governo Federal Brasileiro. *Revista Brasileira de Sistemas de Informação*, *9*(1), 6–32.

Paterson, A. (2003). The Design and Development of a social Science Data Warehouse: A Case

Study of the Human Resources Development Data Warehouse Project of the Human Sciences Research Council, South Africa. *Data Science Journal*, *2*(12–24), 12–24. https://doi.org/http://doi.org/10.2481/dsj.2.12

Pereira, A. K., Pires, P. S., & Pinto, A. (2014). Pesquisas de Avaliação e Confidencialidade da Informação: Limites e Conflitos. *Revista Brasileira de Monitoramento E Avaliação*, *7*, 82–99.

Petruccelli, J. L., & Saboia, A. L. (2013). *Características Étnico-raciais da População. Classificação e identidades*. Retrieved from http://servicodados.ibge.gov.br/Download/Download.ashx?http=1&u=biblioteca.ibge.gov.br/visualizacao/livros/liv49891.pdf

Polidori, M. M., Marinho-Araujo, C. M., & Barreyro, G. B. (2006). Perspectivas e desafios na avaliação da educação superior brasileira. *Ensaio: Aval. Pol. Públ. Educ*, *1453*, 425–436. https://doi.org/10.1590/S0104-40362006000400002

Polo, M. G. de O. (2015). Governo , sociedade civil e os desafios na publicação de dados abertos : o caso da base de dados do Programa Nacional de Apoio à Cultura no Brasil .

Ponniah, P. (2001). *Data Warehousing Fundamentals: A Comprehensive Guide for IT Professionals* (Vol. 6). John Wiley & Sons, Inc.

Post, D. (2015). Does Watching Help ? In Search of the Theory of Change for Education Monitoring. *Current Issues in Comparative Education*, *17*(1), 72–86. Retrieved from https://eric.ed.gov/?id=EJ1061022

Pronko, M. (2015). O Banco Mundial no campo internacional da educação. In J. M. M. Pereira & M. Pronko (Eds.), *A demolição de direitos: um exame das políticas do Banco Mundial para a educação e a saúde (1980-2013)* (pp. 89–112). Rio de Janeiro: EPSJV. Retrieved from http://www.epsjv.fiocruz.br/sites/default/files/l240.pdf

Resende, W. D. C., & Nassif, M. E. (2015). Aplicação da lei de acesso à informação em portais de transparência governamentais brasileiros. *Revista Eletrônica de Biblioteconomia E Ciência Da Informação*, *20*(42), 1. https://doi.org/10.5007/1518-2924.2015v20n42p1

Rigo, S. J., Cambruzzi, W., Barbosa, J. L. V., & Cazella, S. C. (2014). Minerando Dados Educacionais com foco na evasão escolar: oportunidades, desafios e necessidades. *Revista Brasileira de Informática Na Educação*, *22*(1), 132. https://doi.org/10.5753/RBIE.2014.22.01.132

Rigotti, J. I. R. (2004). Variáveis de educação dos censos demográficos brasileiros de 1960 a 2000. In E. L. G. Rios-Neto & J. L. de R. Riani (Eds.), *Introdução à demografia da educação* (pp. 129–142). Campinas: Associação Brasileira de Estudos Populacionais - ABEP.

RIPSA, R. I. de I. para a S. (2008). *Indicadores básicos de Saúde no Brasil: conceitos e aplicações*. (R. Astorino, Ed.) (1ª edição). Brasília: Organização Pan-Americana da Saúde. Retrieved from http://www.ripsa.org.br

Sá, J. V. de O., Carvalho, J. Á., & Kaldeich, C. (2012). A multi-driven approach to requirements analysis of data warehouse schema: a case study. *IADIS 2013*, 8. Retrieved from http://hdl.handle.net/1822/22145

Saviani, D. (2012). O Inep, o diagnóstico da educação brasileira e a Rbep. *Revista Brasileira de Estudos Pedagógicos*, *93*(234).

Schaefer, B. C., Tanrikulu, E., & Breiter, A. (2011). Eliciting user requirements when there is no

organization: A mixed method for an educational data warehouse project. *Procedia - Social and Behavioral Sciences*, *28*, 743–748. https://doi.org/10.1016/j.sbspro.2011.11.137

Senra, N. de C. (2009). *Uma Breve história das estatísticas brasileiras (1822-2002)*. Rio de Janeiro: Centro de Documentação e Informações e Disseminação de Informações - CDDI. Retrieved from www.ibge.gov.br

Shen, C., Riaz, Z., Palle, M. S., Jin, Q., & Peña-Mora, F. (2015). Open Data Landscape: A Global Perspective and a Focus on China. In M. Janssen, M. Mäntymäki, J. Hidders, B. Klievink, W. Lamersdorf, B. van Loenen, & A. Zuiderwijk (Eds.), *Open and Big Data Management and Innovation : 14th IFIP WG 6.11 Conference on e-Business, e-Services, and e-Society, I3E 2015, Delft, The Netherlands, October 13-15, 2015, Proceedings* (pp. 247–260). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-319-25013-7_20

Silva, A. B. de O. (2003). O sistema de informações estatísticas no Brasil e as relações entre seus produtores e usuários. *Ciência Da Informação*, *34*(2), 62–69. https://doi.org/10.1590/S0100-19652005000200007

Souza, R. M. de O., & Oliveira, E. A. M. (2012). O censo escolar no contexto da democratização da educação básica e do pacto federativo brasileiro, 1464–1473.

Steibel, F. (2014). *Independent Reporting mechanism - Brasil : relatório do progresso Brasil 2013-2014*. Washington, DC. Retrieved from http://www.opengovpartnership.org/sites/default/files/Brasil_Relatório2013-14_Final_0.pdf

Strand, M. (2005). *External Data Incorporation into Data Warehouses*. *Knowledge Creation Diffusion Utilization*. Retrieved from file:///C:/Users/TUL/Documents/Litteratur/Strand - 2005 - External Data Incorporation into Data Warehouses.pdf

Susha, I., Zuiderwijk, A., Janssen, M., Ke, Å., & Nlund, G. (2015). Benchmarks for evaluating the Pprogress of open Data adoption: usage, limitations, and lessons learned. *Social Science Computer Review*, *33*(5), 613–630. https://doi.org/10.1177/0894439314560852

Ubaldi, B. (2013). *Towards Empirical Analysis of Open Government Data Initiatives*. *OECD Working Papers on Public Governance*. Paris. https://doi.org/http://dx.doi.org/10.1787/5k46bj4f03s7-en

Verger, A., Lubienski, C., & Steiner-Khamsi, G. (2016). The Emergence and Structuring of the Global Education Industry: Towards an Analytical Framework. In *World Yearbook of Education 2016: The Global Education Industry* (pp. 1–26). https://doi.org/10.1063/1.857816

Viana, G., & Lima, J. F. de. (2010). Capital humano e crescimento econômico. *Interações (Campo Grande)*, *11*(2), 137–148. https://doi.org/10.1590/S1518-70122010000200003

Weinstein, J., & Goldstein, J. (2012). The benefits of a big tent: Opening up government in developing countries a response to Yu & Robinson's the new ambiguity of "Open Government." *UCLA Law Review Discourse*, *60*(2012), 38–48. Retrieved from http://www.uclalawreview.org/pdf/discourse/60-3.pdf

Zuiderwijk, A., & Janssen, M. (2014). Open data policies, their implementation and impact: A framework for comparison. *Government Information Quarterly*, *31*(1), 17–29. https://doi.org/10.1016/j.giq.2013.04.003

8. ANEXOS

Documentação do banco “db_dados_metadados”:

Tabela: “tbl_base_ano_layout”

Descrição: armazena a informação sobre o ano que um determinado layout de base foi utilizado em uma determinada pesquisa. Diferentes edições (anos) de uma pesquisa podem compartilhar um mesmo layout.

Objetivo: evitar a duplicação de informações nas tabelas "tbl_base_variaveis" e tbl_base_variavel_values".

DDL:

```
CREATE TABLE `tbl_base_ano_layout` (
`fk_base` INT(11) NOT NULL,
`id_ano` INT(11) NOT NULL,
`fk_tp_file` INT(11) NULL DEFAULT NULL,
`flg_header` SET('Y','N') NULL DEFAULT NULL,
`fk_tp_separador` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`fk_base`, `id_ano`),
INDEX `FK_tbl_base_pesquisa_tbl_tp_file` (`fk_tp_file`),
INDEX `FK_tbl_base_pesquisa_tbl_tp_separador` (`fk_tp_separador`),
CONSTRAINT `FK_tbl_base_ano_layout_tbl_base_pesquisa` FOREIGN KEY
(`fk_base`) REFERENCES `tbl_base_pesquisa` (`id_base`) ON UPDATE NO
ACTION ON DELETE NO ACTION,
CONSTRAINT `FK_tbl_base_pesquisa_tbl_tp_file` FOREIGN KEY (`fk_tp_file`)
REFERENCES `tbl_tp_file` (`id_tp_file`) ON UPDATE NO ACTION ON DELETE NO
ACTION,
CONSTRAINT `FK_tbl_base_pesquisa_tbl_tp_separador` FOREIGN KEY
(`fk_tp_separador`) REFERENCES `tbl_tp_separador` (`id_tp_separador`) ON
UPDATE NO ACTION ON DELETE NO ACTION
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;
```

Tabela: “tbl_base_arquivo”

Descrição: armazena os nomes dos arquivos que estão relacionados a uma determinada pesquisa. Uma pesquisa deve conter no mínimo um arquivo cadastrado para poder ser utilizada no processamento. Ela também informa o caminho onde o arquivo deve ser achado e qual o tamanho desse.

Objetivo: informar à aplicação o local e nome dos arquivos que devem ser processados.

DDL:

```
CREATE TABLE `tbl_base_arquivo` (
`id_arquivo` INT(11) NOT NULL AUTO_INCREMENT,
`fk_base_pesquisa` INT(11) NOT NULL DEFAULT '0',
`nm_arquivo` VARCHAR(150) NOT NULL DEFAULT '0',
```

```
`caminho_arquivo` VARCHAR(500) NOT NULL DEFAULT '0',
`tamanho_arquivo` VARCHAR(50) NOT NULL DEFAULT '0',
PRIMARY KEY (`id_arquivo`),
INDEX `FK_tbl_base_arquivo_tbl_base_pesquisa` (`fk_base_pesquisa`),
CONSTRAINT `FK_tbl_base_arquivo_tbl_base_pesquisa` FOREIGN KEY
(`fk_base_pesquisa`) REFERENCES `tbl_base_ano_layout` (`fk_base`) ON
UPDATE NO ACTION ON DELETE NO ACTION
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=17
;
```

Tabela: "tbl_base_pesquisa"

Descrição: armazena as bases que compõem uma determinada pesquisa. Uma pesquisa deve conter no mínimo uma base. Além disso são armazenadas informações sobre tipo de layout que (existência de header ou o tipo de separador) que os arquivos da pesquisa possuem. Por último, são armazenadas as unidades de análise da base, já que uma pesquisa pode ter diferentes unidades de analise/investigação.

Objetivo: fazer a conexão entre arquivos, pesquisas e indicadores.

DDL:

```
CREATE TABLE `tbl_base_pesquisa` (
`id_base` INT(11) NOT NULL AUTO_INCREMENT,
`nm_base` VARCHAR(50) NOT NULL,
`fk_pesquisa` INT(11) NOT NULL,
`ds_base` TEXT NULL,
`fk_tp_unidade_analise` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`id_base`),
INDEX `FK_tbl_base_pesquisa_tbl_pesquisa` (`fk_pesquisa`),
INDEX `FK_tbl_base_pesquisa_tbp_tp_unidade_analise`
(`fk_tp_unidade_analise`),
CONSTRAINT `FK_tbl_base_pesquisa_tbl_pesquisa` FOREIGN KEY
(`fk_pesquisa`) REFERENCES `tbl_pesquisa` (`id_pesquisa`) ON UPDATE NO
ACTION ON DELETE NO ACTION,
CONSTRAINT `FK_tbl_base_pesquisa_tbp_tp_unidade_analise` FOREIGN KEY
(`fk_tp_unidade_analise`) REFERENCES `tbl_tp_unidade_analise`
(`id_tp_unidade_analise`) ON UPDATE NO ACTION ON DELETE NO ACTION
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=6
;
```

Tabela: "tbl_base_variaveis"

Descrição: armazena as informações básicas das variáveis (campos) que uma base possui.

Objetivo: informa à aplicação as variáveis e suas informações para a execução do processamento. Também pode ser utilizada para a criação automática de tabelas em um banco de dados (ex. criação de tabelas staging).

DDL:

```
CREATE TABLE `tbl_base_variaveis` (
`id_var` INT(11) NOT NULL AUTO_INCREMENT,
`fk_base` INT(11) NOT NULL,
`nm_var` VARCHAR(50) NOT NULL,
`fk_varflag` INT(11) NULL DEFAULT NULL,
`fk_tp_var` INT(11) NULL DEFAULT NULL,
`ds_var` TEXT NULL,
`pos_inicial` INT(11) NULL DEFAULT NULL,
`pos_final` INT(11) NULL DEFAULT NULL,
`tamanho` VARCHAR(8) NULL DEFAULT NULL,
`var_missing` VARCHAR(50) NULL DEFAULT NULL COMMENT 'armazena os valores
que devem ser excluidos do processamento',
PRIMARY KEY (`id_var`),
INDEX `FK_tbl_base_variaveis_tbl_varflag` (`fk_varflag`),
INDEX `FK_tbl_base_variaveis_tbl_tp_variavel` (`fk_tp_var`),
INDEX `FK_tbl_base_variaveis_tbl_base_pesquisa` (`fk_base`),
CONSTRAINT `FK_tbl_base_variaveis_tbl_base_pesquisa` FOREIGN KEY
(`fk_base`) REFERENCES `tbl_base_ano_layout` (`fk_base`) ON UPDATE NO
ACTION ON DELETE NO ACTION,
CONSTRAINT `FK_tbl_base_variaveis_tbl_tp_variavel` FOREIGN KEY
(`fk_tp_var`) REFERENCES `tbl_tp_variavel` (`id_tp_variavel`) ON UPDATE
NO ACTION ON DELETE NO ACTION,
CONSTRAINT `FK_tbl_base_variaveis_tbl_varflag` FOREIGN KEY (`fk_varflag`)
REFERENCES `tbl_tp_varflag` (`id_var_flag`) ON UPDATE NO ACTION ON DELETE
NO ACTION
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
```

Tabela: "tbl_base_variavel_values"

Descrição: armazena as informações das codificações das variáveis categóricas.

Objetivo: facilitar a interação dos utilizadores com a interface de criação dos indicadores no momento do filtro, já que traduz os códigos em conceitos inteligíveis ao utilizador.

DDL:

```
CREATE TABLE `tbl_base_variavel_values` (
`fk_var` INT(11) NULL DEFAULT NULL,
`cod_value` VARCHAR(50) NULL DEFAULT NULL,
`ds_value` VARCHAR(400) NULL DEFAULT NULL,
INDEX `FK_tbl_base_variavel_value_tbl_base_variavel` (`fk_var`),
CONSTRAINT `FK_tbl_base_variavel_value_tbl_base_variavel` FOREIGN KEY
(`fk_var`) REFERENCES `tbl_base_variaveis` (`id_var`) ON UPDATE NO ACTION
ON DELETE NO ACTION
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
```

Tabela: "tbl_bases_link"

Descrição: armazena as informações que permitem a interligação das bases.

Objetivo: permitir que a aplicação realize "joins" entre as bases.

DDL:

```
CREATE TABLE `tbl_bases_link` (
`id_link` INT(11) NOT NULL AUTO_INCREMENT,
`fk_base_pai` INT(11) NULL DEFAULT NULL,
`fk_chave_pai` INT(11) NULL DEFAULT NULL,
`fk_base_filho` INT(11) NULL DEFAULT NULL,
`fk_chave_filho` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`id_link`),
INDEX `fk_base_pai_fk_base_filho` (`fk_base_pai`, `fk_base_filho`),
INDEX `FK_tbl_base_link_tbl_var_1` (`fk_chave_pai`),
INDEX `FK_tbl_base_link_tbl_var_2` (`fk_chave_filho`),
INDEX `FK_tbl_base_link_tbl_base_pesquisa_2` (`fk_base_filho`),
CONSTRAINT `FK_tbl_base_link_tbl_base_pesquisa_1` FOREIGN KEY
(`fk_base_pai`) REFERENCES `tbl_base_ano_layout` (`fk_base`) ON UPDATE NO
ACTION ON DELETE NO ACTION,
CONSTRAINT `FK_tbl_base_link_tbl_base_pesquisa_2` FOREIGN KEY
(`fk_base_filho`) REFERENCES `tbl_base_ano_layout` (`fk_base`) ON UPDATE
NO ACTION ON DELETE NO ACTION,
CONSTRAINT `FK_tbl_base_link_tbl_var_1` FOREIGN KEY (`fk_chave_pai`)
REFERENCES `tbl_base_variaveis` (`id_var`) ON UPDATE NO ACTION ON DELETE
NO ACTION,
CONSTRAINT `FK_tbl_base_link_tbl_var_2` FOREIGN KEY (`fk_chave_filho`)
REFERENCES `tbl_base_variaveis` (`id_var`) ON UPDATE NO ACTION ON DELETE
NO ACTION
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=6
;
```

Tabela: “tbl_grao_geo”

Descrição: armazenas os tipos de granularidade espacial

Objetivo: Tabela Auxiliar - informa à aplicação qual a menor dimensão geográfica que é possível em determinada base.

DDL:

```
CREATE TABLE `tbl_grao_geo` (
`id_grao_geo` INT(11) NOT NULL AUTO_INCREMENT,
`nm_grao_geo` VARCHAR(50) NOT NULL,
PRIMARY KEY (`id_grao_geo`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
ROW_FORMAT=DYNAMIC
AUTO_INCREMENT=7
;
```

Tabela: “tbl_grao_tempo”

Descrição: armazenas os tipos de granularidade temporal

Objetivo: Tabela Auxiliar- informa à aplicação qual a menor dimensão temporal que é possível em determinada base.

```
CREATE TABLE `tbl_grao_tempo` (
`id_grao_tempo` INT(11) NOT NULL AUTO_INCREMENT,
`nm_grao_tempo` VARCHAR(50) NOT NULL,
PRIMARY KEY (`id_grao_tempo`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=2
;
```

Tabela: "tbl_pesquisa"

Descrição: armazena as informações das pesquisas ou dos registros administrativos.

Objetivo: identificar os tipos (censitário, amostral, registro adm, etc), temáticas (Educação, Mercado de Trabalho, etc) e a granularidade de tempo e espaço em as informações são produzidas

DDL:

```
CREATE TABLE `tbl_pesquisa` (
`id_pesquisa` INT(11) NOT NULL AUTO_INCREMENT,
`fk_produtor` INT(11) NULL DEFAULT NULL,
`nm_pesquisa` VARCHAR(250) NULL DEFAULT NULL,
`sigla_pesquisa` VARCHAR(10) NULL DEFAULT NULL,
`url_pesquisa` VARCHAR(500) NULL DEFAULT NULL,
`ds_pesquisa` TEXT NULL,
`fk_tematica` INT(11) NULL DEFAULT NULL,
`fk_tp_pesquisa` INT(11) NULL DEFAULT NULL,
`grao_tempo` INT(11) NULL DEFAULT NULL,
`grao_geo` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`id_pesquisa`),
INDEX `FK_tbl_pesquisa_tbl_produtor_pesquisa` (`fk_produtor`),
INDEX `FK_tbl_pesquisa_tbl_tp_pesquisa` (`fk_tp_pesquisa`),
INDEX `FK_tbl_pesquisa_tbl_grao_tempo` (`grao_tempo`),
INDEX `FK_tbl_pesquisa_tbl_tgrao_geo` (`grao_geo`),
INDEX `FK_tbl_pesquisa_tbl_tematica_pesquisa` (`fk_tematica`),
CONSTRAINT `FK_tbl_pesquisa_tbl_grao_tempo` FOREIGN KEY (`grao_tempo`)
REFERENCES `tbl_grao_tempo` (`id_grao_tempo`) ON UPDATE NO ACTION ON
DELETE NO ACTION,
CONSTRAINT `FK_tbl_pesquisa_tbl_produtor_pesquisa` FOREIGN KEY
(`fk_produtor`) REFERENCES `tbl_produtor_pesquisa` (`id_produtor`) ON
UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT `FK_tbl_pesquisa_tbl_tematica_pesquisa` FOREIGN KEY
(`fk_tematica`) REFERENCES `tbl_tematica_pesquisa` (`id_tematica`) ON
UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT `FK_tbl_pesquisa_tbl_tgrao_geo` FOREIGN KEY (`grao_geo`)
REFERENCES `tbl_grao_geo` (`id_grao_geo`) ON UPDATE NO ACTION ON DELETE
NO ACTION,
CONSTRAINT `FK_tbl_pesquisa_tbl_tp_pesquisa` FOREIGN KEY
(`fk_tp_pesquisa`) REFERENCES `tbl_tp_pesquisa` (`id_tp_pesquisa`) ON
UPDATE NO ACTION ON DELETE NO ACTION
)
COLLATE='utf8_general_ci'
```

```
ENGINE=InnoDB
AUTO_INCREMENT=3
;
```

Tabela: “tbl_produtor_pesquisa”

Descrição: armazena as informações da instituição produtora dos dados.

Objetivo: Tabela Auxiliar - seu principal objetivo é agregar as pesquisas segundo o órgão produtor.

DDL:

```
CREATE TABLE `tbl_produtor_pesquisa` (
`id_produtor` INT(11) NOT NULL AUTO_INCREMENT,
`nm_produtor` VARCHAR(250) NULL DEFAULT NULL,
`sigla_produtor` VARCHAR(10) NULL DEFAULT NULL,
PRIMARY KEY (`id_produtor`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=4
;
```

Tabela: “tbl_tematica_pesquisa”

Descrição: armazena as possíveis temáticas de uma pesquisa.

Objetivo: Tabela Auxiliar - serve para agregar as pesquisas de acordo com as temáticas.

DDL:

```
CREATE TABLE `tbl_tematica_pesquisa` (
`id_tematica` INT(11) NOT NULL AUTO_INCREMENT,
`tp_tematica` VARCHAR(50) NULL DEFAULT NULL,
PRIMARY KEY (`id_tematica`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=4
;
```

Tabela: “tbl_tp_file”

Descrição: armazena os tipos de arquivo (extensão – csv, txt, dbf, por exemplo), nos quais os dados são disponibilizados

Objetivo: Tabela Auxiliar - é utilizada pela aplicação para definir como a leitura do arquivo deve ser processada.

DDL:

```
CREATE TABLE `tbl_tp_file` (
```

```
`id_tp_file` INT(11) NOT NULL AUTO_INCREMENT,
`nm_tp_file` VARCHAR(3) NULL DEFAULT '0',
PRIMARY KEY (`id_tp_file`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=3
;
```

Tabela: “tbl_tp_pesquisa”

Descrição: armazena os tipos de pesquisa possíveis ("Censo", "Registro Administrativo", "Amostra", etc).

Objetivo: Tabela Auxiliar - é utilizada para definir a forma de processamento dos dados, já que nos censo e registros administrativos, não há a necessidade de utilização de ponderações.

DDL:

```
CREATE TABLE `tbl_tp_pesquisa` (
`id_tp_pesquisa` INT(11) NOT NULL AUTO_INCREMENT,
`nm_tp_pesquisa` VARCHAR(50) NOT NULL,
PRIMARY KEY (`id_tp_pesquisa`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=4
;
```

Tabela: “tbl_tp_separador”

Descrição: armazena os tipos separadores de campos possíveis em uma base ("vírgula", "ponto e vírgula", "pipe", "Marca de Tabulação", etc).

Objetivo: Tabela Auxiliar - é utilizada pela aplicação para definir como a leitura do arquivo deve ser processada.

DDL:

```
CREATE TABLE `tbl_tp_separador` (
`id_tp_separador` INT(11) NOT NULL AUTO_INCREMENT,
`nm_separador` VARCHAR(25) NOT NULL,
`separador` CHAR(1) NOT NULL,
PRIMARY KEY (`id_tp_separador`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=5
;
```

Tabela: **“**tbl_tp_varflag”

Descrição: armazena as informações dos tipos especiais que uma determinada variável pode assumir.

Objetivo: Tabela Auxiliar - informa à aplicação o papel dessas variáveis especiais no momento do processamento. Os possíveis papeis são: Linkage de bases ("Chave Primária" e "Chave Estrangeira"), Agregação ("Chave Geográfica" e "Chave Temporal") ou Cômputo ("Peso" e "Valor Monetário").

DDL:

```
CREATE TABLE `tbl_tp_varflag` (
`id_var_flag` INT(11) NOT NULL AUTO_INCREMENT,
`nm_var_flag` VARCHAR(50) NULL DEFAULT '0',
`sigla_var_flag` VARCHAR(2) NULL DEFAULT '0',
PRIMARY KEY (`id_var_flag`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=11
;
```

Tabela: “tbl_tp_variavel”

Descrição: armazena as informações dos tipos de dados de uma determinada variável ("Numérico", "String", "Data", "Monetário", "Hora", "Ignorado").

Objetivo: Tabela Auxiliar – informar à aplicação como uma determinada variável pode ser utilizada.

DDL:

```
CREATE TABLE `tbl_tp_variavel` (
`id_tp_variavel` INT(11) NOT NULL AUTO_INCREMENT,
`nm_tp_variavel` VARCHAR(50) NULL DEFAULT NULL,
`tp_var_sql` VARCHAR(50) NULL DEFAULT NULL,
PRIMARY KEY (`id_tp_variavel`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=100
;
```

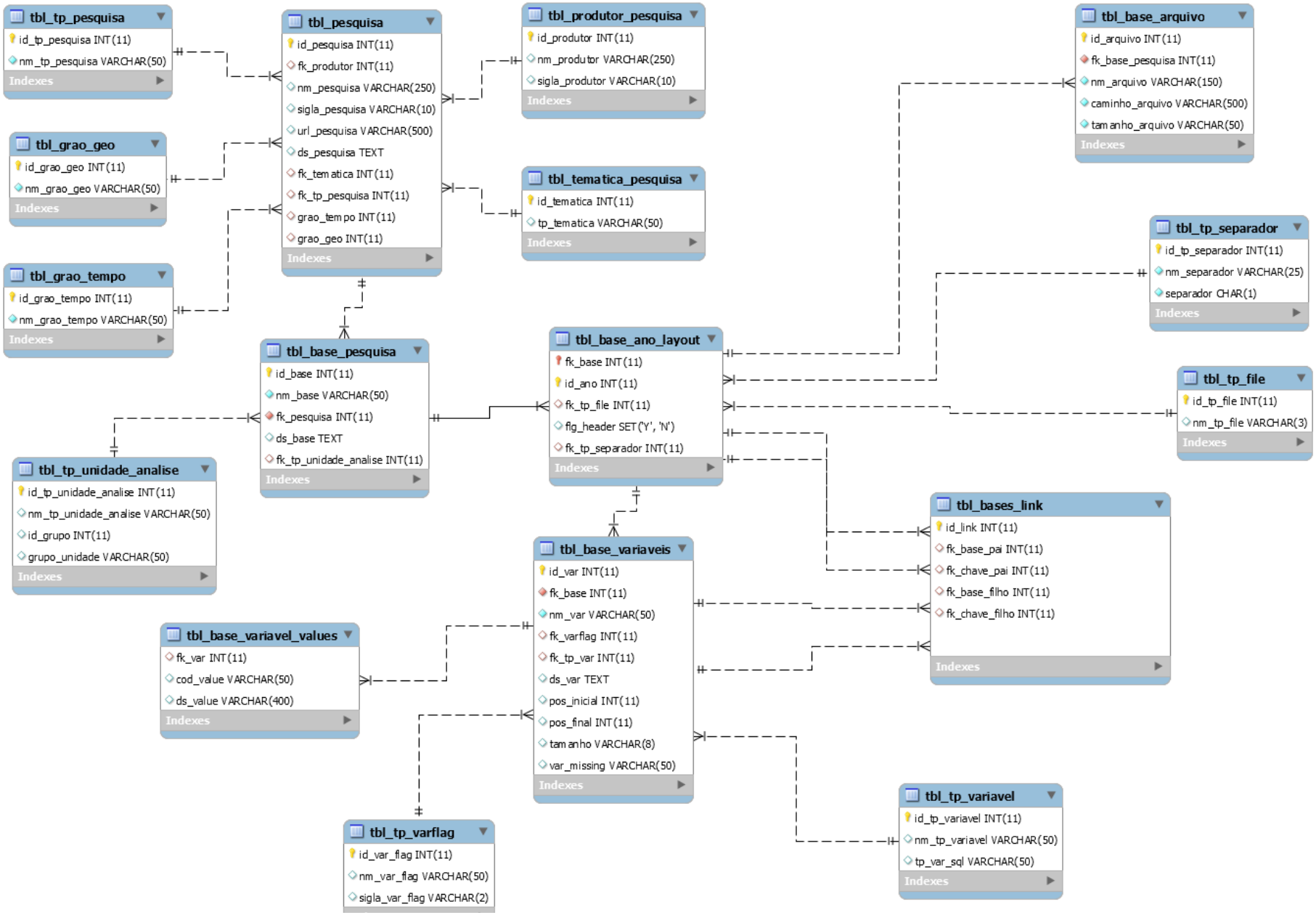

Figura 26 – DER – Banco de Metadados

Documentação do banco “db_indicadores”:

Tabela: `tbl_indicadores`

Descrição: armazena as informações básicas do indicador (nome, descrição, tipo e unidade de análise).

Objetivo: fazer o cadastramento inicial dos indicadores no sistema.

DDL:

```
CREATE TABLE `tbl_indicadores` (
`id_indicador` INT(11) NOT NULL AUTO_INCREMENT,
`nm_indicador` VARCHAR(300) NULL DEFAULT NULL,
`ds_indicador` TEXT NULL,
`fk_tp_indicdor` INT(11) NULL DEFAULT NULL,
`fk_tp_unidade_analise` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`id_indicador`),
INDEX `FK_tbl_indicador_tbl_tp_indicador` (`fk_tp_indicdor`),
CONSTRAINT `FK_tbl_indicador_tbl_tp_indicador` FOREIGN KEY
(`fk_tp_indicdor`) REFERENCES `tbl_tp_indicador` (`id_tp_indicador`) ON
UPDATE CASCADE ON DELETE CASCADE
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=9
;
```

Tabela: `tbl_indicador_agregacao`

Descrição: armazena as variáveis pelas quais os indicadores serão agregados. Na prática, as agregações correspondem as dimensões do modelo dimensional.

Objetivo: possibilitar a visualização dos indicadores pode diferentes dimensões.

DDL:

```
CREATE TABLE `tbl_indicador_agregacao` (
`id_agregacao` INT(11) NOT NULL AUTO_INCREMENT,
`fk_intancia` INT(11) NULL DEFAULT NULL,
`fk_parcela` INT(11) NULL DEFAULT NULL,
`id_dimensao` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`id_agregacao`),
UNIQUE INDEX `fk_intancia_fk_parcela_id_dimensao` (`fk_intancia`,
`fk_parcela`, `id_dimensao`),
INDEX `FK_01` (`fk_parcela`),
CONSTRAINT `FK_01` FOREIGN KEY (`fk_parcela`) REFERENCES
`tbl_parcelas_indicador` (`id_parcela`) ON UPDATE CASCADE ON DELETE CASCADE
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=23
;
```

Tabela: `tbl_indicador_ano`

Descrição: representa a instancia temporal do indicador, referenciado pela edição da base de dados.

Objetivo: além de possibilitar a construção de diferentes versões de um mesmo indicador, a tabela funciona como controle de processamento do ETL.

DDL:

```
CREATE TABLE `tbl_indicador_ano` (
`id_instancia` INT(11) NOT NULL AUTO_INCREMENT,
`id_indicador` INT(11) NOT NULL DEFAULT '0',
`id_ano` INT(11) NOT NULL DEFAULT '0',
`flg_proc` INT(11) NOT NULL DEFAULT '0',
PRIMARY KEY (`id_instancia`),
UNIQUE INDEX `id_indicador_id_ano` (`id_indicador`, `id_ano`),
CONSTRAINT `FK_tbl_indicador_tbl_indicador_ano` FOREIGN KEY
(`id_indicador`) REFERENCES `tbl_indicadores` (`id_indicador`) ON UPDATE
CASCADE ON DELETE CASCADE
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=6
;
```

Tabela: `tbl_indicador_totalizacao`

Descrição: armazena a variável pela qual o indicador será totalização ou contabilizado.

Objetivo: possibilitar a realização das contagens e somatório durante a execução do ETL/SQL

DDL:

```
CREATE TABLE `tbl_indicador_totalizacao` (
`id_totalizacao` INT(11) NOT NULL AUTO_INCREMENT,
`fk_intancia` INT(11) NULL DEFAULT NULL,
`fk_parcela` INT(11) NULL DEFAULT NULL,
`ds_totalizacao` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`id_totalizacao`),
UNIQUE INDEX `fk_parcela_ds_totalizacao` (`fk_parcela`, `ds_totalizacao`),
CONSTRAINT `FK_02` FOREIGN KEY (`fk_parcela`) REFERENCES
`tbl_parcelas_indicador` (`id_parcela`) ON UPDATE NO ACTION ON DELETE NO
ACTION
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
ROW_FORMAT=DYNAMIC
AUTO_INCREMENT=4
;
```

Tabela: `tbl_operador_comparacao`

Descrição: armazena os principais operadores de comparação utilizados no processamento.

Objetivo: Tabela Auxiliar - possibilitar à aplicação PHP exibir aos utilizadores os operadores de comparação. Além disso, fornece as versões desses operadores em Pyhton e SQL, para a utilização das rotinas de ETL.

DDL:

```
CREATE TABLE `tbl_operador_comparacao` (
`id_comparacao` INT(11) NOT NULL AUTO_INCREMENT,
`ds_comparacao` VARCHAR(50) NULL DEFAULT NULL,
`operador_python` VARCHAR(50) NULL DEFAULT NULL,
`operador_sql` VARCHAR(50) NULL DEFAULT NULL,
PRIMARY KEY (`id_comparacao`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=9
;
```

Tabela: `tbl_operador_logico`

Descrição: armazena os principais operadores lógicos utilizados no processamento.

Objetivo: Tabela Auxiliar - possibilitar à aplicação PHP exibir aos utilizadores os operadores lógicos. Além disso, fornece as versões desses operadores em Pyhton e SQL, para a utilização das rotinas de ETL.

DDL:

```
CREATE TABLE `tbl_operador_logico` (
`id_logico` INT(11) NOT NULL AUTO_INCREMENT,
`ds_logico` VARCHAR(50) NULL DEFAULT NULL,
`operador_python` VARCHAR(50) NULL DEFAULT NULL,
`operador_petl` VARCHAR(50) NULL DEFAULT NULL,
`operador_sql` VARCHAR(50) NULL DEFAULT NULL,
PRIMARY KEY (`id_logico`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=4
;
```

Tabela: `tbl_parcelas_indicador`

Descrição: armazena as parcelas que compõem um determinado indicador. Além disso identifica o tipo de totalização que deverá ser realizado em cada uma delas.

Objetivo: possibilitar a construção de indicadores de factos não aditivos.

DDL:

```
CREATE TABLE `tbl_parcelas_indicador` (
`id_parcela` INT(11) NOT NULL AUTO_INCREMENT,
`fk_instancia` INT(11) NULL DEFAULT '0',
`fk_pesquisa` INT(11) NULL DEFAULT '0',
```

```
`fk_tp_parcela` SET('Numerador','Denominador','Parcela Única') NULL DEFAULT
NULL,
`ds_parcela` VARCHAR(300) NULL DEFAULT '0',
`ds_unidade_totalizacao` SET('Ponderação','Contagem','Cálculo') NULL
DEFAULT NULL,
PRIMARY KEY (`id_parcela`),
INDEX `FK_tbl_parcela_indicador_tbl_indicador_ano` (`fk_instancia`),
CONSTRAINT `FK_tbl_parcela_indicador_tbl_indicador_ano` FOREIGN KEY
(`fk_instancia`) REFERENCES `tbl_indicador_ano` (`id_instancia`) ON UPDATE
NO ACTION ON DELETE NO ACTION
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=5
;
```

Tabela: `tbl_parcela_atributo`

Descrição: armazena as características de cada uma das parcelas do indicador, dentre elas, a fonte de dados utilizada, as variáveis utilizadas, os operadores (lógico e de comparação) e os valores de filtro que deverm ser atribuídos em cada parcela.

Objetivo: possibilitar a construção dos indicadores pelo processamento ETL.

DDL:

```
CREATE TABLE `tbl_parcela_atributo` (
`id_atributo` INT(11) NOT NULL AUTO_INCREMENT,
`fk_parcela` INT(11) NOT NULL,
`fk_base_pesquisa` INT(11) NOT NULL,
`fk_variavel` INT(11) NOT NULL,
`fk_comparacao` INT(11) NOT NULL,
`ds_valores` VARCHAR(250) NOT NULL,
`fk_logica` INT(11) NOT NULL,
PRIMARY KEY (`id_atributo`),
INDEX `FK_tbl_parcela_indicador_tbl_parcela_atributo` (`fk_parcela`),
INDEX `FK_tbl_parcela_indicador_tbl_operador_comparacao` (`fk_comparacao`),
INDEX `FK_tbl_parcela_indicador_tbl_operador_logico` (`fk_logica`),
CONSTRAINT `FK_tbl_parcela_indicador_tbl_operador_comparacao` FOREIGN KEY
(`fk_comparacao`) REFERENCES `tbl_operador_comparacao` (`id_comparacao`) ON
UPDATE NO ACTION ON DELETE NO ACTION,
CONSTRAINT `FK_tbl_parcela_indicador_tbl_operador_logico` FOREIGN KEY
(`fk_logica`) REFERENCES `tbl_operador_logico` (`id_logico`) ON UPDATE NO
ACTION ON DELETE NO ACTION,
CONSTRAINT `FK_tbl_parcela_indicador_tbl_parcela_atributo` FOREIGN KEY
(`fk_parcela`) REFERENCES `tbl_parcelas_indicador` (`id_parcela`) ON UPDATE
NO ACTION ON DELETE NO ACTION
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=11
```

Tabela: `tbl_tp_indicador`

Descrição: indicada a unidade de medida na qual o indicador está representado (“Valor absoluto”, “Média”, “Taxa”, “Percentual”, “Razão”).

Objetivo: Tabela Auxiliar – indicadar no sistama PHP as opções de medida.

DDL:

```
CREATE TABLE `tbl_tp_indicador` (
`id_tp_indicador` INT(11) NOT NULL AUTO_INCREMENT,
`nm_tp_indicador` VARCHAR(50) NULL DEFAULT NULL,
PRIMARY KEY (`id_tp_indicador`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=6
;
```

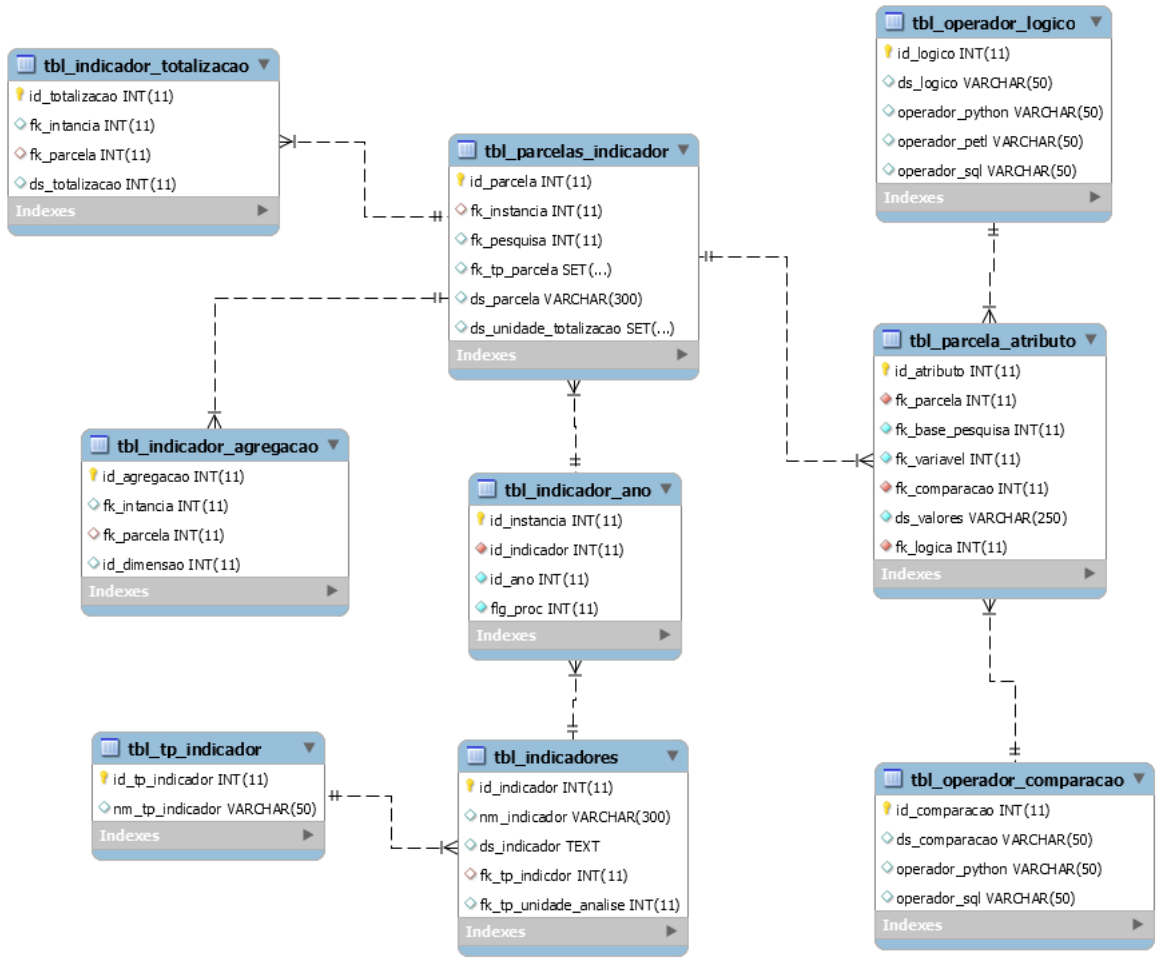

Figura 27 – DER – Banco de Indicadores

Documentação do banco “db_dimensional”:

Dimensão Gênero:

```
CREATE TABLE `dim_genero` (
`id_sexo` INT(11) NOT NULL AUTO_INCREMENT,
`nm_sexo` VARCHAR(25) NULL DEFAULT '0',
`slg_sexo` VARCHAR(1) NULL DEFAULT '0',
PRIMARY KEY (`id_sexo`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
AUTO_INCREMENT=1000
;
```

Dimensão Étinco-racial:

```
CREATE TABLE `dim_cor` (
`id_cor` INT(11) NOT NULL,
`raca_cor_01` VARCHAR(50) NULL DEFAULT NULL,
`raca_cor_02` VARCHAR(50) NULL DEFAULT NULL,
`raca_cor_03` VARCHAR(50) NULL DEFAULT NULL,
`raca_cor_04` VARCHAR(50) NULL DEFAULT NULL,
PRIMARY KEY (`id_cor`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;
```

Dimensão Faixa Etária:

```
CREATE TABLE `dim_faixa_etaria` (
`id_idade` INT(11) NOT NULL,
`faixa_01` VARCHAR(50) NULL DEFAULT NULL,
`faixa_02` VARCHAR(50) NULL DEFAULT NULL,
`faixa_03` VARCHAR(50) NULL DEFAULT NULL,
`faixa_ciclo` VARCHAR(50) NULL DEFAULT NULL,
`faixa_pia` VARCHAR(50) NULL DEFAULT NULL,
PRIMARY KEY (`id_idade`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;
```

Dimensão Indicador:

```
CREATE TABLE `dim_parcela` (
`id_parcela` INT(11) NOT NULL,
`id_indicador` INT(11) NULL DEFAULT NULL,
`ds_parcela` VARCHAR(50) NULL DEFAULT NULL,
`tp_parcela` VARCHAR(50) NULL DEFAULT NULL,
`tp_unidade` VARCHAR(50) NULL DEFAULT NULL,
PRIMARY KEY (`id_parcela`),
INDEX `FK3_2` (`id_indicador`),
CONSTRAINT `FK3_2` FOREIGN KEY (`id_indicador`) REFERENCES `dim_indicador`
(`id_indicador`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;
```

```
CREATE TABLE `dim_indicador` (
`id_indicador` INT(11) NOT NULL,
`nm_indicador` VARCHAR(250) NULL DEFAULT NULL,
`ds_indicador` VARCHAR(250) NULL DEFAULT NULL,
`id_area` INT(11) NULL DEFAULT NULL,
`ds_formula` VARCHAR(50) NULL DEFAULT NULL,
`df_fator` VARCHAR(50) NULL DEFAULT NULL,
PRIMARY KEY (`id_indicador`),
INDEX `FK3_1` (`id_area`),
CONSTRAINT `FK3_1` FOREIGN KEY (`id_area`) REFERENCES `dim_area` (`id_area`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;


CREATE TABLE `dim_area` (
`id_area` INT(11) NOT NULL,
`id_classe` INT(11) NOT NULL,
`nm_area` INT(11) NOT NULL,
PRIMARY KEY (`id_area`),
INDEX `FK1_1a` (`id_classe`),
CONSTRAINT `FK1_1a` FOREIGN KEY (`id_classe`) REFERENCES `dim_acao` (`id_classe`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;


CREATE TABLE `dim_acao` (
`id_classe` INT(11) NOT NULL,
`nm_classe` VARCHAR(250) NOT NULL,
`id_acao` INT(11) NOT NULL,
`nm_acao` VARCHAR(250) NOT NULL,
PRIMARY KEY (`id_classe`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;
```

Dimensão Geográfica:

```
CREATE TABLE `dim_geo_distrito` (
`id_geo_distrito` INT(11) NOT NULL,
`id_geo_munic` INT(11) NULL DEFAULT NULL,
`cod_distrito` BIGINT(20) NULL DEFAULT NULL,
`nm_distrito` VARCHAR(250) NULL DEFAULT NULL,
PRIMARY KEY (`id_geo_distrito`),
INDEX `FK2_3a` (`id_geo_munic`),
CONSTRAINT `FK2_3a` FOREIGN KEY (`id_geo_munic`) REFERENCES `dim_geo_municipio`
(`id_geo_munic`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;

CREATE TABLE `dim_geo_estado` (
`id_geo_estado` INT(11) NOT NULL,
`cod_estado` INT(11) NULL DEFAULT NULL,
`nm_estado` VARCHAR(50) NULL DEFAULT NULL,
`slg_estado` VARCHAR(2) NULL DEFAULT NULL,
`cod_regiao` INT(11) NULL DEFAULT NULL,
`nm_regiao` VARCHAR(50) NULL DEFAULT NULL,
`slg_regiao` VARCHAR(2) NULL DEFAULT NULL,
PRIMARY KEY (`id_geo_estado`)
)
```

```
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;

CREATE TABLE `dim_geo_municipio` (
`id_geo_munic` INT(11) NOT NULL,
`id_geo_estado` INT(11) NULL DEFAULT NULL,
`cod_munic` INT(11) NULL DEFAULT NULL,
`nm_munic` VARCHAR(250) NULL DEFAULT NULL,
`cod_meso` INT(11) NULL DEFAULT NULL,
`nm_meso` VARCHAR(250) NULL DEFAULT NULL,
`cod_micro` INT(11) NULL DEFAULT NULL,
`nm_micro` VARCHAR(250) NULL DEFAULT NULL,
`cod_rm_ride` INT(11) NULL DEFAULT NULL,
`nm_rm_ride` VARCHAR(250) NULL DEFAULT NULL,
PRIMARY KEY (`id_geo_munic`),
INDEX `FK2_4a` (`id_geo_estado`),
CONSTRAINT `FK2_4a` FOREIGN KEY (`id_geo_estado`) REFERENCES `dim_geo_estado`
(`id_geo_estado`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;

CREATE TABLE `dim_geo_setor` (
`id_geo_setor` BIGINT(20) NOT NULL,
`id_geo_distrito` INT(11) NULL DEFAULT NULL,
`cod_setor` BIGINT(20) NULL DEFAULT NULL,
`cod_subdistrito` BIGINT(20) NULL DEFAULT NULL,
`nm_subdistrito` VARCHAR(250) NULL DEFAULT NULL,
`cod_bairro` BIGINT(20) NULL DEFAULT NULL,
`nm_bairro` VARCHAR(250) NULL DEFAULT NULL,
`situacao_setor` INT(11) NULL DEFAULT NULL,
`tp_setor` INT(11) NULL DEFAULT NULL,
PRIMARY KEY (`id_geo_setor`),
INDEX `FK2_2_a` (`id_geo_distrito`),
CONSTRAINT `FK2_2_a` FOREIGN KEY (`id_geo_distrito`) REFERENCES `dim_geo_distrito`
(`id_geo_distrito`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;
```

Dimensão Tempo:

```
CREATE TABLE `dim_tmp_ano` (
`id_ano` INT(11) NOT NULL,
`flg_bissexto` VARCHAR(50) NULL DEFAULT NULL,
PRIMARY KEY (`id_ano`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;

CREATE TABLE `dim_tmp_mes` (
`id_mes` INT(11) NOT NULL,
`id_ano` INT(11) NULL DEFAULT NULL,
`mes` VARCHAR(2) NULL DEFAULT NULL,
`nm_mes` VARCHAR(20) NULL DEFAULT NULL,
`slg_mes` VARCHAR(3) NULL DEFAULT NULL,
PRIMARY KEY (`id_mes`),
INDEX `FK1_2a` (`id_ano`),
CONSTRAINT `FK1_2a` FOREIGN KEY (`id_ano`) REFERENCES `dim_tmp_ano` (`id_ano`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
```

```
;

CREATE TABLE `dim_tmp_trimestre` (
`id_trimestre` INT(11) NOT NULL,
`id_mes` INT(11) NULL DEFAULT NULL,
`trimestre` VARCHAR(1) NULL DEFAULT NULL,
`nm_trimestre` VARCHAR(20) NULL DEFAULT NULL,
`slg_trimestre` VARCHAR(6) NULL DEFAULT NULL,
PRIMARY KEY (`id_trimestre`),
INDEX `FK1_3a` (`id_mes`),
CONSTRAINT `FK1_3a` FOREIGN KEY (`id_mes`) REFERENCES `dim_tmp_mes` (`id_mes`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;
```

Factos:

```
CREATE TABLE `fato_ano_distrito` (
`PARCELA` INT(1) NOT NULL,
`DIM_ANO` INT(11) NOT NULL,
`DIM_DIST` INT(11) NOT NULL,
`DIM_SEXO` INT(3) NOT NULL,
`DIM_IDADE` INT(3) NOT NULL,
`DIM_COR` INT(3) NOT NULL,
`VALUE` INT(11) NOT NULL,
PRIMARY KEY (`PARCELA`, `DIM_ANO`, `DIM_DIST`, `DIM_SEXO`, `DIM_IDADE`, `DIM_COR`),
INDEX `FK5_2` (`DIM_ANO`),
INDEX `FK5_3` (`DIM_DIST`),
INDEX `FK5_4` (`DIM_SEXO`),
INDEX `FK5_5` (`DIM_IDADE`),
INDEX `FK5_6` (`DIM_COR`),
CONSTRAINT `FK5_1` FOREIGN KEY (`PARCELA`) REFERENCES `dim_parcela` (`id_parcela`),
CONSTRAINT `FK5_2` FOREIGN KEY (`DIM_ANO`) REFERENCES `dim_tmp_ano` (`id_ano`),
CONSTRAINT `FK5_3` FOREIGN KEY (`DIM_DIST`) REFERENCES `dim_geo_distrito`
(`id_geo_distrito`),
CONSTRAINT `FK5_4` FOREIGN KEY (`DIM_SEXO`) REFERENCES `dim_genero` (`id_sexo`),
CONSTRAINT `FK5_5` FOREIGN KEY (`DIM_IDADE`) REFERENCES `dim_faixa_etaria`
(`id_idade`),
CONSTRAINT `FK5_6` FOREIGN KEY (`DIM_COR`) REFERENCES `dim_cor` (`id_cor`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;

CREATE TABLE `fato_ano_municipio` (
`PARCELA` INT(1) NOT NULL,
`DIM_ANO` INT(11) NOT NULL,
`DIM_MUNIC` INT(11) NOT NULL,
`DIM_SEXO` INT(3) NOT NULL,
`DIM_IDADE` INT(3) NOT NULL,
`DIM_COR` INT(3) NOT NULL,
`VALUE` INT(11) NOT NULL,
PRIMARY KEY (`PARCELA`, `DIM_ANO`, `DIM_MUNIC`, `DIM_SEXO`, `DIM_IDADE`,
`DIM_COR`),
INDEX `FK7_2` (`DIM_ANO`),
INDEX `FK7_3` (`DIM_MUNIC`),
INDEX `FK7_4` (`DIM_SEXO`),
INDEX `FK7_5` (`DIM_IDADE`),
INDEX `FK7_6` (`DIM_COR`),
CONSTRAINT `FK7_1` FOREIGN KEY (`PARCELA`) REFERENCES `dim_parcela` (`id_parcela`),
CONSTRAINT `FK7_2` FOREIGN KEY (`DIM_ANO`) REFERENCES `dim_tmp_ano` (`id_ano`),
CONSTRAINT `FK7_3` FOREIGN KEY (`DIM_MUNIC`) REFERENCES `dim_geo_municipio`
(`id_geo_munic`),
CONSTRAINT `FK7_4` FOREIGN KEY (`DIM_SEXO`) REFERENCES `dim_genero` (`id_sexo`),
CONSTRAINT `FK7_5` FOREIGN KEY (`DIM_IDADE`) REFERENCES `dim_faixa_etaria`
(`id_idade`),
```

```
CONSTRAINT `FK7_6` FOREIGN KEY (`DIM_COR`) REFERENCES `dim_cor` (`id_cor`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;

CREATE TABLE `fato_ano_setor` (
`PARCELA` INT(1) NOT NULL,
`DIM_ANO` INT(11) NOT NULL,
`DIM_SETOR` BIGINT(11) NOT NULL,
`DIM_SEXO` INT(3) NOT NULL,
`DIM_IDADE` INT(3) NOT NULL,
`DIM_COR` INT(3) NOT NULL,
`VALUE` INT(11) NOT NULL,
PRIMARY KEY (`PARCELA`, `DIM_ANO`, `DIM_SETOR`, `DIM_SEXO`, `DIM_IDADE`,
`DIM_COR`),
INDEX `FK6_2` (`DIM_ANO`),
INDEX `FK6_3` (`DIM_SETOR`),
INDEX `FK6_4` (`DIM_SEXO`),
INDEX `FK6_5` (`DIM_IDADE`),
INDEX `FK6_6` (`DIM_COR`),
CONSTRAINT `FK6_1` FOREIGN KEY (`PARCELA`) REFERENCES `dim_parcela` (`id_parcela`),
CONSTRAINT `FK6_2` FOREIGN KEY (`DIM_ANO`) REFERENCES `dim_tmp_ano` (`id_ano`),
CONSTRAINT `FK6_3` FOREIGN KEY (`DIM_SETOR`) REFERENCES `dim_geo_setor`
(`id_geo_setor`),
CONSTRAINT `FK6_4` FOREIGN KEY (`DIM_SEXO`) REFERENCES `dim_genero` (`id_sexo`),
CONSTRAINT `FK6_5` FOREIGN KEY (`DIM_IDADE`) REFERENCES `dim_faixa_etaria`
(`id_idade`),
CONSTRAINT `FK6_6` FOREIGN KEY (`DIM_COR`) REFERENCES `dim_cor` (`id_cor`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;

CREATE TABLE `fato_ano_uf` (
`PARCELA` INT(1) NOT NULL,
`DIM_ANO` INT(11) NOT NULL,
`DIM_UF` INT(11) NOT NULL,
`DIM_SEXO` INT(3) NOT NULL,
`DIM_IDADE` INT(3) NOT NULL,
`DIM_COR` INT(3) NOT NULL,
`VALUE` INT(11) NOT NULL,
PRIMARY KEY (`PARCELA`, `DIM_ANO`, `DIM_UF`, `DIM_SEXO`, `DIM_IDADE`, `DIM_COR`),
INDEX `FK4_2` (`DIM_ANO`),
INDEX `FK4_3` (`DIM_UF`),
INDEX `FK4_4` (`DIM_SEXO`),
INDEX `FK4_5` (`DIM_IDADE`),
INDEX `FK4_6` (`DIM_COR`),
CONSTRAINT `FK4_1` FOREIGN KEY (`PARCELA`) REFERENCES `dim_parcela` (`id_parcela`),
CONSTRAINT `FK4_2` FOREIGN KEY (`DIM_ANO`) REFERENCES `dim_tmp_ano` (`id_ano`),
CONSTRAINT `FK4_3` FOREIGN KEY (`DIM_UF`) REFERENCES `dim_geo_estado`
(`id_geo_estado`),
CONSTRAINT `FK4_4` FOREIGN KEY (`DIM_SEXO`) REFERENCES `dim_genero` (`id_sexo`),
CONSTRAINT `FK4_5` FOREIGN KEY (`DIM_IDADE`) REFERENCES `dim_faixa_etaria`
(`id_idade`),
CONSTRAINT `FK4_6` FOREIGN KEY (`DIM_COR`) REFERENCES `dim_cor` (`id_cor`)
)
COLLATE='utf8_general_ci'
ENGINE=InnoDB
;
```

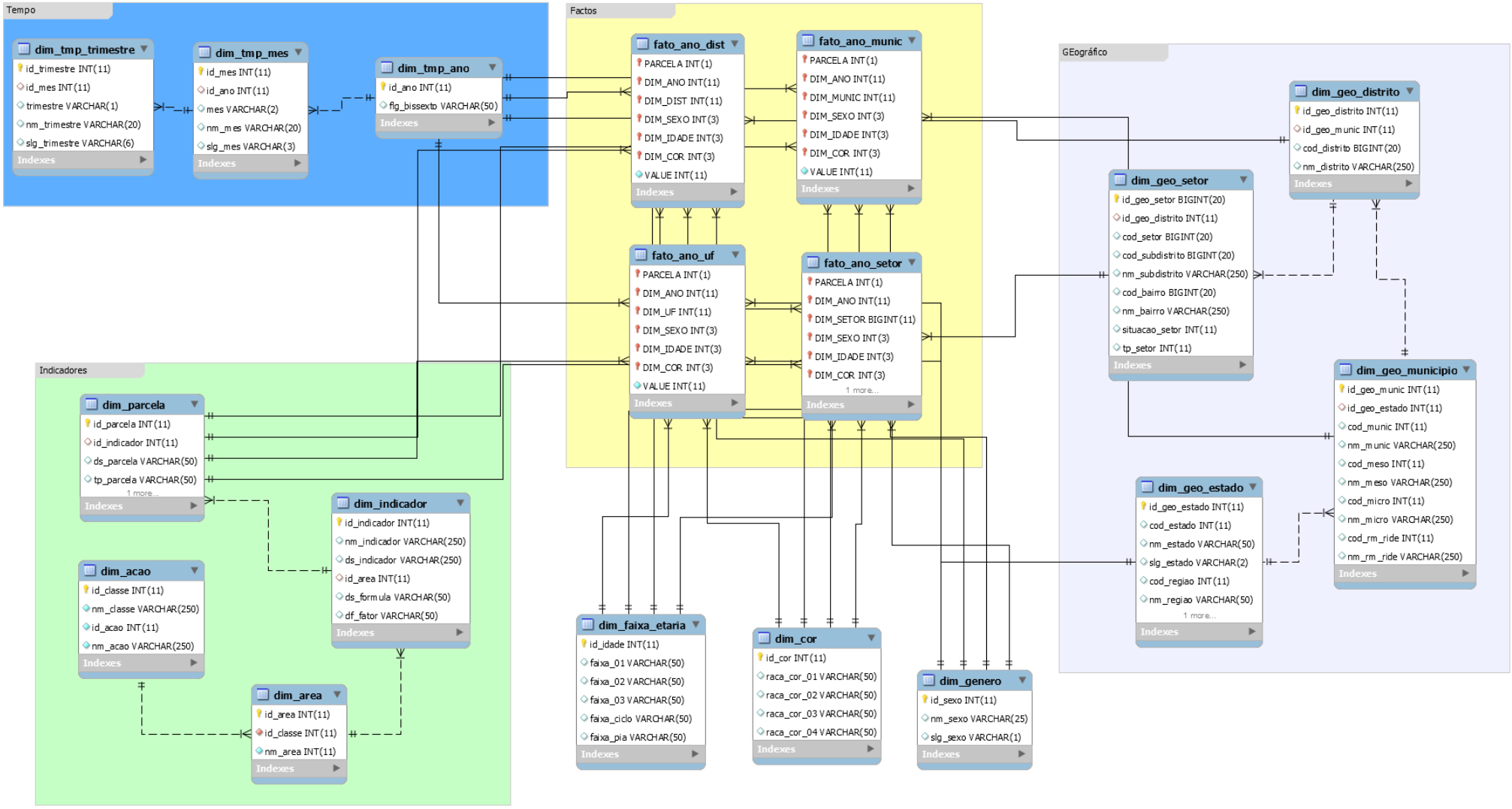

Figura 28 – DER Banco Dimensional

Codificação Python:

```
################################################
#  PROGRAMAÇÃO DE CRIAÇÃO E CARGAS DAS
#  TABELAS STAGE
################################################

import pandas as pd
from datetime import datetime
import logging

from MyFunctions import conexao as cn, sql_script as sq, func_agregacao as
fa, drop_table as dt, move_fato as mf, finaliza_proc as fc, info

now = datetime.now()
log_file = 'logfile_{0}.log'.format(now.strftime("%Y-%m-%d_%H%M%S"))
logger = logging.getLogger(__name__)
hdlr = logging.FileHandler("{0}\{1}".format(info.caminho,log_file))
formatter =
logging.Formatter('%(asctime)s;%(name)s;%(levelname)s;%(message)s')
hdlr.setFormatter(formatter)
logger.addHandler(hdlr)
logger.setLevel(logging.INFO)

logger.info('Início do Processamento')

print('+ Início do Processamento: {0}'.format(str(datetime.now())))
alter_table = []
monta_proc = []

try:
    dt.drop()
    logger.info('Limpa Banco Staging')

except:
    logger.error('Erro ao limpar o Banco Staging')

# Retorna os indicadores para processamento: id_indicador; nm_indicador;
tp_indicador; parcelas; base: {base_pesquisa; variáveis};
atributos:{variáveis; comparação; logica}

try:
    conn_01 = cn.conn_01
    cursor_01 = conn_01.cursor()
    sql_04 = sq.sql_04()
    cursor_01.execute(sql_04)
    dados = cursor_01.fetchall()
    logger.info('Leitura dos Indicadores para Processamento (SQL_04)')

except:

    logger.error('Erro na Leitura dos Indicadores para Processamento
(SQL_04)')


if len(dados)> 0:
    for dado in dados:
        v_ind = dado[0]
        v_par = dado[3]
        v_base = dado[4].split(',')[0]
```

```
        v_vars = dado[4].split(',')[1]
        v_atri = dado[5]
        v_instan = dado[6]
        parcela_base = str(v_par) + '_' + str(v_base)

logger.info('@Indicador={0}|cod={1}|instancia={2}'.format(dado[1],v_ind,v_i
nstan))
        try:
            sql_05 = sq.sql_05(v_base , v_vars)
            conn_02 = cn.conn_02
            cursor_02 = conn_02.cursor()
            cursor_02.execute(sql_05)
            base_info = cursor_02.fetchall()
            logger.info('Leitura das informações para montagem das tabelas
stage (SQL_05)')
        except:
            logger.info('Erro na Leitura das informações para montagem das
tabelas stage (SQL_05)')

        list_var = []
        mnt_sql = ''
        var_chave = ''
        var_index = ''
        lista_proc = []
        if len(base_info) > 0:
            for q in base_info:
                nm_var = q[1]
                chave = q[2]
                tp_var = q[3]
                tam_var = q[4]

                if chave in ['PK' , 'FK']:
                    if tp_var == 'varchar':
                        tp_var = tp_var + '(' + str(tam_var) + ')'
                        var_index = var_index + 'ADD INDEX
`{0}`(`{0}`),'.format(nm_var)
                        mnt_sql = mnt_sql + '`{0}` {1} DEFAULT
NULL,\n'.format(nm_var , tp_var)
                    else:
                        var_index = var_index + 'ADD INDEX
`{0}`(`{0}`),'.format(nm_var)
                        mnt_sql = mnt_sql + '`{0}` {1} DEFAULT
NULL,\n'.format(nm_var , tp_var)
                    flg_chave = 1
                elif chave in ['GK']:
                    var_index = var_index + 'ADD INDEX
`{0}`(`{0}`),'.format(nm_var)
                    mnt_sql = mnt_sql + '`{0}` {1} unsigned DEFAULT
NULL,\n'.format(nm_var , tp_var)
                    chave_geo = nm_var

                elif chave in ['TK']:
                    var_index = var_index + 'ADD INDEX
`{0}`(`{0}`),'.format(nm_var)
                    mnt_sql = mnt_sql + '`{0}` {1} unsigned DEFAULT
NULL,\n'.format(nm_var , tp_var)
                    chave_tmp = nm_var

                else:
                    if tp_var == 'varchar':
                        tp_var = tp_var + '(' + str(tam_var) + ')'
```

```
                    else:
                        tp_var = tp_var
                    mnt_sql = mnt_sql + '`{0}` {1} unsigned DEFAULT
NULL,\n'.format(nm_var , tp_var)

                list_var.append(nm_var)

            var = list_var

            if flg_chave == 1:
                mnt_sql = mnt_sql[0:len(mnt_sql) - 2]
            else:
                mnt_sql = mnt_sql[0:len(mnt_sql) - 2]

            cabecalho = 'DROP TABLE IF EXISTS `tbl_{0}`;\n CREATE TABLE IF
NOT EXISTS `tbl_{0}` (\n'.format(parcela_base)
            rodape = '\n) ENGINE=MyISAM DEFAULT CHARSET=utf8;'

            chave_geo = 'tbl_{0}.{1}'.format(parcela_base,chave_geo)
            chave_tmp = 'tbl_{0}.{1}'.format(parcela_base , chave_tmp)

            mnt_sql = cabecalho + mnt_sql + rodape

            try:
                conn_03 = cn.conn_03
                cursor_03 = conn_03.cursor()
                cursor_03.execute(mnt_sql)
                logger.info('Monta tabela Staging (MNT_SQL)')
            except:
                logger.error('Erro ao Montar tabela Staging (MNT_SQL)')

            try:
                sql_06 = sq.sql_06(v_base)
                cursor_02.execute(sql_06)
                base_atributos = cursor_02.fetchall()
                logger.info('Retorna informações sobre os arquivos para
processamento (SQL_06)')
            except:
                logger.error('Erro ao retornar informações sobre os
arquivos para processamento (SQL_06)')



            if len(base_atributos)> 0:
                for f in base_atributos:

                    tipo_file = f[3]
                    header_ = f[5]
                    separador = f[6]
                    caminho = f[7]
                    file = f[8]

                    if header_ == "Y":
                        header_ = 0
                    else:
                        header_ = None

                    filtro = ''
                    for t2 in v_atri.split('-'):
                        variavel = t2.split('|')[0]
                        operador = t2.split('|')[1]
```

```
                        valor = t2.split('|')[2]
                        logica = t2.split('|')[3]
                        sql_07 = sq.sql_07(variavel)
                        cursor_02.execute(sql_07)
                        variavel = cursor_02.fetchone()[0]
                        sql_08 = sq.sql_08(operador)
                        cursor_01.execute(sql_08)
                        operador = cursor_01.fetchone()[0]
                        sql_09 = sq.sql_09(logica)
                        cursor_01.execute(sql_09)
                        logica = cursor_01.fetchone()[0]

                        y = ''
                        valor = valor.split(',')
                        for x in valor:
                            y = y + '"' + x + '",'
                        valor = y[0:len(y) - 1]

                        if operador not in ["not in" , "in"]:
                            x = "{0} {1} {2} {3} ".format(variavel ,
operador , valor , logica)
                            filtro = filtro + x
                        elif operador in ["not in" , "in"]:
                            x = "{0} {1} ({2}) {3} ".format(variavel ,
operador , valor , logica)
                            filtro = filtro + x

                    print(filtro)
                    if tipo_file.lower() == 'csv':
                        try:
                            df = pd.read_csv(caminho + '\\' + file ,
sep=separador , header=header_ , encoding='cp1252' , dtype=object ,
usecols=var)
                            df = df.query(filtro.strip(' '))
                            logger.info('Leitura do arquivo de dados
(PANDAS)')
                        except:
                            logger.error('Erro na leitura do arquivo de
dados (PANDAS)')

                        try:
                            engine = cn.engine_03
                            conn = engine.connect()
                            tbl_nm = 'tbl_{0}'.format(parcela_base)
                            df.to_sql(tbl_nm , con=conn ,
if_exists='append' , index=False , chunksize=10)
                            print('tabela: {0} - {1}
registros'.format(tbl_nm , len(df)))
                            logger.info('++ Carga do arquivo de dados
(PANDAS) tabela: {0} - {1} registros'.format(tbl_nm , len(df)))
                        except:
                            logger.error('Erro na carga do arquivo de dados
(PANDAS) tabela: {0} '.format(tbl_nm))

                    elif tipo_file.lower() == 'txt':
                        pass

                tbl_nm = 'tbl_{0}'.format(parcela_base)
                var_index = var_index[0:len(var_index) - 1]
                var_index = "ALTER TABLE `{0}` {1};".format(tbl_nm ,
var_index)
```

```
                alter_table.append(var_index)
            else:
                print('|-- Indicadores sem atributos configurados')
                logger.error('Error! - Indicadores sem atributos
configurados')

        else:
            print('|-- Indicadores sem parametros configurados')
            logger.error('Error! - Indicadores sem parametros
configurados')


        lista_proc.append(parcela_base)
        monta_proc.append(lista_proc)

else:
    print('|-- Não há indicadores para serem processados!')
    logger.info('Não há indicadores para serem processados! - Fim do
Processamento')



print('|-- Fim Carga dos Dados: {0}'.format(str(datetime.now())))

#EXECUTA OS INDICES DAS TABELAS
for b in alter_table:
    try:
        conn.execute(b)
        logger.info('Cria index na tabela staging')
    except:
        logger.error('Erro ao criar index na tabela staging')



try:
    print(monta_proc,v_instan)
    entrada = fa.agregacao(monta_proc,v_instan)
    logger.info('Processa agregação')
except:
    logger.error('Erro ao processar agregação')

try:
    mf.move_registros(entrada)
    logger.info('Carrega modelo Dimensional')
except:
    logger.erro('Erro ao carregar modelo Dimensional')

try:
    fc.final_proc(v_instan,0)
    logger.info('Finaliza Processamento - Altera o status do indicador')
except:
    logger.info('Erro ao finalizar Processamento')

#FECHA CONEXAO
try:
    conn_01.close()
    cursor_01.close()

    cursor_02.close()
    conn_02.close()
```

```
    cursor_03.close()
    conn_03.close()

except:
    pass


print('|-- Fim Processamento: {0}'.format(str(datetime.now())))
logger.info('Fim do Processamento')


############################################################
# PROGRAMA PARA A CRIAÇÃO DOS DDL DE AGRAGAÇÃO
############################################################


import itertools
from MyFunctions import conexao as cn

conn_022 = cn.conn_02
cursor_022 = conn_022.cursor()

conn_011 = cn.conn_01
cursor_011 = conn_011.cursor()

conn_033 = cn.conn_03
cursor_033 = conn_033.cursor()



def variaveis(v_instan,parcela):
    var_t = set()
    sql_11 = """select
                c.id_dimensao
                from
                tbl_indicadores as a left join tbl_indicador_ano as b on
a.id_indicador = b.id_indicador
                left join tbl_indicador_agregacao as c on b.id_instancia =
c.fk_intancia
                where b.flg_proc = 0 and  b.id_instancia ={0} and
c.fk_parcela = {1}""".format(v_instan,parcela)

    cursor_011.execute(sql_11)
    result = cursor_011.fetchall()

    if result:
        for i in result:
            sql_12 = "select concat('_',fk_base,'.',nm_var) from
tbl_base_variaveis where id_var = {0}".format(i[0])
            cursor_022.execute(sql_12)
            t = cursor_022.fetchall()
            var_t.add(t[0][0])

        return var_t


############################################################################
##


def totaliza(v_instan,parcela):
```

```
    var_g = list()
    sql_13 = """select
                c.ds_totalizacao
                from
                tbl_indicadores as a left join tbl_indicador_ano as b on
a.id_indicador = b.id_indicador
                left join tbl_indicador_totalizacao as c on b.id_instancia
= c.fk_intancia
                where b.flg_proc = 0 and  b.id_instancia = {0} and
c.fk_parcela = {1}""".format(v_instan,parcela)

    cursor_011.execute(sql_13)
    result = cursor_011.fetchall()

    if result:
        for i in result:
            sql_12 = "select concat('_',fk_base,'.',nm_var) from
tbl_base_variaveis where id_var = {0}".format(i[0])
            cursor_022.execute(sql_12)
            g = cursor_022.fetchall()
            var_g.append(g[0])
        return var_g[0][0]


##########################################################################
##

def tipo_operacao(parcela):

    sql_14 = """select ds_unidade_totalizacao from tbl_parcelas_indicador
where id_parcela = {0}""".format(parcela)
    cursor_011.execute(sql_14)
    result = cursor_011.fetchall()

    tp_op = ''
    if result[0][0] == 'Contagem':

        tp_op = ', COUNT(DISTINCT '

    elif result[0][0] == 'Ponderação':

        tp_op = ', SUM( '

    elif result[0][0] == 'Cálculo':

        print('FALTA OPERAÇÃO!!!!')

    return tp_op



##########################################################################
##


def define_grao_tmp(parcela):

    sql_11a = "select fk_pesquisa from tbl_parcelas_indicador where
id_parcela = {0}".format(parcela)
    cursor_011.execute(sql_11a)
```

```
    result = cursor_011.fetchall()
    for i in result:
        sql_11a1 = "select grao_tempo from tbl_pesquisa where id_pesquisa =
{0}".format(i[0])
        cursor_022.execute(sql_11a1)
        result = cursor_022.fetchall()
        for x in result:
            return x[0]


##############################################################################
##

def define_grao_geo(parcela):

    sql_11a = "select fk_pesquisa from tbl_parcelas_indicador where
id_parcela = {0}".format(parcela)
    cursor_011.execute(sql_11a)
    result = cursor_011.fetchall()
    for i in result:
        sql_11a1 = "select grao_geo from tbl_pesquisa where id_pesquisa =
{0}".format(i[0])
        cursor_022.execute(sql_11a1)
        result = cursor_022.fetchall()
        for x in result:
            return x[0]


##############################################################################
##

def monta_sql(v_instan,parcela):
    geo = define_grao_geo(parcela)
    tmp = define_grao_tmp(parcela)

    var_t = set()
    v_list = ''
    g_list = ''
    join = ''

    sql_11 = """select
                c.id_dimensao
                from
                tbl_indicadores as a left join tbl_indicador_ano as b on
a.id_indicador = b.id_indicador
                left join tbl_indicador_agregacao as c on b.id_instancia =
c.fk_intancia
                where b.flg_proc = 0 and  b.id_instancia ={0} and
c.fk_parcela = {1}""".format(v_instan,parcela)
    cursor_011.execute(sql_11)
    result = cursor_011.fetchall()
    if result:
        v_lista = []
        for i in result:
            sql_12a = 'select fk_varflag,nm_var from tbl_base_variaveis
where id_var ={0}'.format(i[0])
            cursor_022.execute(sql_12a)
            v = cursor_022.fetchall()
            v_lista.append(v[0][0])

            if v[0][0] == 4: #chave geo
```

```
                if geo == 4: #município
                    grao_g = " JOIN (select distinct cod_munic,
id_geo_munic from tbl_dim_geo ) AS geo ON (geo.cod_munic = tbl_"
                    nm_geo = 'DIM_MUNIC'
                    v_geo = 'geo.id_geo_munic'
                elif geo == 2: #estado
                    grao_g = " JOIN (select distinct cod_estado,
id_geo_estado from tbl_dim_geo ) AS geo ON (geo.cod_estado = tbl_"
                    nm_geo = 'DIM_UF'
                    v_geo = 'geo.id_geo_estado'
                elif geo == 5: #distrito
                    grao_g = " JOIN (select distinct
cod_distrito,id_geo_distrito from tbl_dim_geo ) AS geo ON (geo.cod_distrito
= tbl_"
                    nm_geo = 'DIM_DIST'
                    v_geo = 'geo.id_geo_distrito'
                elif geo == 6: #setor
                    grao_g = " JOIN (select distinct cod_setor,
id_geo_setor from tbl_dim_geo ) AS geo ON (geo.cod_setor = tbl_"
                    nm_geo = 'DIM_SETOR'
                    v_geo = 'geo.id_geo_setor'
                else: # erro
                    print("ERRO-41!")


                sql_12b = "select concat('_',fk_base,'.',nm_var) from
tbl_base_variaveis where id_var = {0}".format(
                    i[0])
                cursor_022.execute(sql_12b)
                t = cursor_022.fetchall()
                join = join + "{0}{1}{2}) ".format(grao_g,parcela,t[0][0])
                v_list = v_list + ",{0} as {1} ".format(v_geo,nm_geo)
                g_list = g_list+ ",{0} ".format(v_geo)


            elif v[0][0] == 3: #chave tmp

                if tmp == 1: #ano
                    grao_t = " JOIN (select distinct id_ano from
tbl_dim_tmp ) AS tmp ON (tmp.id_ano = tbl_"
                    nm_tmp = "DIM_ANO"
                    v_tmp = 'tmp.id_ano'
                elif tmp == 2: #trimestre
                    pass
                elif tmp == 3: #mês
                    pass
                else: # erro
                    print("ERRO-42!")

                sql_12b = "select concat('_',fk_base,'.',nm_var) from
tbl_base_variaveis where id_var = {0}".format(
                    i[0])
                cursor_022.execute(sql_12b)
                t = cursor_022.fetchall()
                join = join + "{0}{1}{2}) ".format(grao_t,parcela,t[0][0])
                v_list = v_list + ",{0} as {1} ".format(v_tmp,nm_tmp)
                g_list = g_list +  ",{0} ".format(v_tmp)


            elif v[0][0] in [8]: # dim idade
```

```
                grao_i = " JOIN (SELECT DISTINCT idade from tbl_dim_idade)
as tb_idade ON (tb_idade.idade = tbl_"
                nm_dim = "DIM_IDADE"
                v_tmp = "tb_idade.idade"

                sql_12b = "select concat('_',fk_base,'.',nm_var) from
tbl_base_variaveis where id_var = {0}".format(
                    i[0])
                cursor_022.execute(sql_12b)
                t = cursor_022.fetchall()
                join = join + "{0}{1}{2}) ".format(grao_i,parcela,t[0][0])
                v_list = v_list + ",{0} as {1} ".format(v_tmp,nm_dim)
                g_list = g_list +  ",{0} ".format(v_tmp)

            elif v[0][0] in [9, 10]: # dim sexo e raça

                if v[0][0] == 9:
                    nm_sexo = "DIM_SEXO"
                else:
                    nm_sexo = "DIM_COR"


                sql_12b = "select concat('_',fk_base,'.',nm_var) from
tbl_base_variaveis where id_var = {0}".format(
                    i[0])
                cursor_022.execute(sql_12b)
                t = cursor_022.fetchall()
                var = t[0][0]
                script01 = ''
                sql_12c =  """select value_de, value_para from
tbl_base_variavel_values_dimensao where id_var ={0}""".format(i[0])
                cursor_022.execute(sql_12c)
                result = cursor_022.fetchall()

                for v in result:
                    de =v[0]
                    para = v[1]
                    script02 =  "WHEN {0} THEN {1} ".format(de,para)
                    script01 = script01+script02

                v_list = v_list + ',CASE tbl_{0}{1} '.format(parcela,var) +
script01 + ' ELSE 99 END AS {0} '.format(nm_sexo)
                g_list = g_list + ',CASE tbl_{0}{1} '.format(parcela,var) +
script01 + ' ELSE 99 END'


            sql_12b = "select concat('_',fk_base,'.',nm_var) from
tbl_base_variaveis where id_var = {0}".format(i[0])
            cursor_022.execute(sql_12b)
            t = cursor_022.fetchall()
            var_t.add(t[0][0])


        return v_list, g_list, join


##########################################################################
##
```

```
def agregacao(lista,v_instan):

    condicao_on = set()
    parcela_set = set()
    tabelas = set()
    bases = []
    tab_out = []

    for l in lista:
        bases.append(int(l[0].split('_')[1]))
    opcoes = set(itertools.combinations(bases, 2))

    for l in lista: #CRIA A CONDIÇÃO FROM E ON
        parcela = (int(l[0].split('_')[0]))
        parcela_set.add(parcela)
        for o in opcoes:
            sql_10 = """select
                        a.fk_base_pai,
                        (select nm_var from tbl_base_variaveis where id_var
= a.fk_chave_pai) v_pai,
                        a.fk_base_filho,
                        (select nm_var from tbl_base_variaveis where id_var
= a.fk_chave_filho ) v_filho
                        from tbl_bases_link as a where a.fk_base_pai = {0}
and a.fk_base_filho = {1}""".format(o[0],o[1])
            cursor_022.execute(sql_10)
            result = cursor_022.fetchall()
            if result:
                condicao_on.add('ON (tbl_{0}_{1}.{2} =
tbl_{0}_{3}.{4})'.format(parcela,result[0][0],result[0][1],result[0][2],res
ult[0][3]))
                tabelas.add(" FROM tbl_{0}_{1} INNER JOIN tbl_{0}_{2}
".format(parcela,result[0][0],result[0][2]))

    for i, x in enumerate(list(parcela_set)):

        a = ''
        a = "CREATE TABLE tbl_{1} SELECT {0} AS PARCELA
".format(v_instan,x)
        b = list(monta_sql(v_instan,x))
        a = a + b[0]
        a = a + tipo_operacao(x)
        a = a + "tbl_{0}{1} ".format(x,totaliza(v_instan,x))
        a = a + ") as VALUE "
        a = a + sorted(tabelas)[i]
        a = a + sorted(condicao_on)[i]
        a = a + b[2]
        a = a + ' GROUP BY '
        a = a +  b[1][1:]
        tab_out.append("tbl_{0}".format(x))
        geo = define_grao_geo(x)
        tmp = define_grao_tmp(x)

        try:
            s = "DROP TABLE IF EXISTS `tbl_{0}`;".format(x)
            cursor_033.execute(s)
            cursor_033.execute(a[0:len(a)-1])
            print(a)

            print("Processamento OK")
```

```
        except Exception as e:

            print(e)

    return tab_out, geo, tmp



##############################################
# POVOA O MODELO DIMENSIONAL
##############################################


import pandas as pd
import sqlalchemy
from MyFunctions import conexao as cn

#DB STAGE
engine_03 = cn.engine_03
conn_03 = engine_03.connect()

#DB DIMENSÃO
engine_04 = cn.engine_04
conn_04 = engine_04.connect()

#return tab_out, geo, tmp

def move_registros(entrada):

    #entrada = (['tbl_1','tbl_2'], 5, 1)

    bases = entrada[0]
    geo = entrada[1]
    tmp = entrada[2]

    for base in bases:
        sql_tabela = base
        df = pd.read_sql_table(sql_tabela,conn_03)
        tamanho = df.size
        if geo == 2 and tmp == 1: #uf & ano
            tab_destino = 'fato_ano_uf'
        elif geo == 4 and tmp == 1: #munic & ano
            tab_destino = 'fato_ano_municipio'
        elif geo == 5 and tmp == 1: #dist & ano
            tab_destino = 'fato_ano_distrito'
        elif geo == 6 and tmp == 1: #setor & ano
            tab_destino = 'fato_ano_setor'

        df.to_sql(tab_destino , con=conn_04 , if_exists='append' ,
index=False , chunksize=10)
        print("Origem: {0} ; Destino:  {1} ; Registros movidos:
{2}".format(base,tab_destino,tamanho))
```