

**MINISTÉRIO DA DEFESA
EXÉRCITO BRASILEIRO
DEPARTAMENTO DE CIÊNCIA E TECNOLOGIA
INSTITUTO MILITAR DE ENGENHARIA
CURSO DE GRADUAÇÃO EM ENGENHARIA CARTOGRÁFICA**

**1º Ten GABRIEL BOURDON FERNANDES
1º Ten THIAGO MENCK PFEIFER MACEDO**

**PLUGIN DE CONVERSÃO DO MODELO EDGV 3.0 PARA O MODELO
MGCP TRD4 V4.4**

**Rio de Janeiro
2020**

INSTITUTO MILITAR DE ENGENHARIA

1º Ten GABRIEL BOURDON FERNANDES
1º Ten THIAGO MENCK PFEIFER MACEDO

**PLUGIN DE CONVERSÃO DO MODELO EDGV 3.0 PARA O
MODELO MGCP TRD4 V4.4**

Projeto de Fim de Curso apresentado ao Curso de Graduação em Engenharia Cartográfica do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Engenheiro Cartógrafo.

Orientador: Major Felipe Ferrari - M.Sc.

Rio de Janeiro
2020

INSTITUTO MILITAR DE ENGENHARIA
Praça General Tibúrcio, 80 - Praia Vermelha
Rio de Janeiro - RJ CEP 22290-270

Este exemplar é de propriedade do Instituto Militar de Engenharia, que poderá incluí-lo em base de dados, armazenar em computador, microfilmar ou adotar qualquer forma de arquivamento.

É permitida a menção, reprodução parcial ou integral e a transmissão entre bibliotecas deste trabalho, sem modificação de seu texto, em qualquer meio que esteja ou venha a ser fixado, para pesquisa acadêmica, comentários e citações, desde que sem finalidade comercial e que seja feita a referência bibliográfica completa.

Os conceitos expressos neste trabalho são de responsabilidade do(s) autor(es) e do(s) orientador(es).

004.69 Fernandes, Gabriel Bourdon
S586e Plugin de Conversão do Modelo EDGV 3.0 para o Mo-
delo MGCP TRD4 v4.4 / Gabriel Bourdon Fernandes,
Thiago Menck Pfeifer Macedo, orientado por Felipe Fer-
rari - Rio de Janeiro: Instituto Militar de Engenharia,
2020.

47p.: il.

Projeto de Fim de Curso (graduação) - Instituto
Militar de Engenharia, Rio de Janeiro, 2020.

1. Curso de Graduação em Engenharia Cartográfica
- projeto de fim de curso. 1. Palavra 01. 2. Palavra 02.
3. Palavra 03. I. Ferrari, Felipe . II. Título. III.
Instituto Militar de Engenharia.

INSTITUTO MILITAR DE ENGENHARIA

1º Ten GABRIEL BOURDON FERNANDES
1º Ten THIAGO MENCK PFEIFER MACEDO

**PLUGIN DE CONVERSÃO DO MODELO EDGV 3.0 PARA O
MODELO MGCP TRD4 V4.4**

Projeto de Fim de Curso apresentado ao Curso de Graduação em Engenharia Cartográfica do Instituto Militar de Engenharia, como requisito parcial para a obtenção do título de Engenheiro de Cartográfico.

Orientador: Major Felipe Ferrari - M.Sc.

Aprovado em 31 de Fevereiro de 2020 pela seguinte Banca Examinadora:

Major Felipe Ferrari - M.Sc. do IME

Major Ivanildo Barbosa - D.Sc. do IME

Capitão Philipe Borba - Eng. Crtf. da DSG

Rio de Janeiro
2020

Ao Instituto Militar de Engenharia, alicerce da minha formação e aperfeiçoamento.

AGRADECIMENTOS

Agradeço a todas as pessoas que me incentivaram, apoiaram e possibilitaram esta oportunidade de ampliar meus horizontes.

Meus familiares, amigos e mestres.

Em especial ao meu Professor Orientador Major Felipe Ferrari por suas disponibilidades e atenções.

“Sem publicação, a ciência é morta. ”

GERARD PIEL

SUMÁRIO

LISTA DE ILUSTRAÇÕES	8
LISTA DE SIGLAS	9
1 INTRODUÇÃO	13
1.1 Objetivo	13
1.2 Especificações Técnicas para Estruturação de Dados Geoespaciais Ve- toriais (ET-EDGV)	13
1.3 Multinational Geospatial Co-production Program (MGCP)	14
1.3.1 Technical Reference Documentation (TRD)	15
1.4 Motivação da Pesquisa	15
2 FUNDAMENTAÇÃO TEÓRICA	16
2.1 Bancos de Dados	16
2.1.1 Entidades	16
2.1.2 Relacionamento de Entidades	17
2.1.3 Atributos	17
2.1.4 Modelo Conceitual	18
2.1.5 Diagrama de Classes	19
2.2 Formato JSON (<i>JavaScript Object Notation</i>)	20
2.3 QGIS - Multiplataforma de Sistema de Informação Geográfica	20
2.3.1 DSG Tools - Plugin	21
2.4 Python	22
2.5 Estruturas de Classes e Atributos dos Padrões Considerados	22
2.6 Conversões de Modelo	25
3 METODOLOGIA	27
3.1 Plugin - Parte Computacional	27
3.1.1 Plugin <i>EDGV Converter</i>	27
3.1.1.1 Interface do Plugin	27
3.1.1.2 Funcionamento do Plugin	28
3.1.2 Adaptações Necessárias no Plugin	32
3.2 Bancos de Dados Utilizados	32
3.2.1 Banco de Dados TRD4 v4.4	32

3.2.2	Banco de Dados ET-EDGV 3.0	34
3.3	Arquivo JSON Utilizado	35
3.4	Base de Dados Teste.....	38
3.5	Cronograma Proposto	39
4	RESULTADOS	40
4.0.1	Plugin Final	40
4.0.2	Resultado Obtido com a Base de Dados Teste	40
5	CONCLUSÃO	41
5.1	Continuidade da Pesquisa	41
6	REFERÊNCIAS BIBLIOGRÁFICAS	42
7	APÊNDICES	46
8	ANEXOS	47

LISTA DE ILUSTRAÇÕES

FIG.2.1	Notações Utilizadas em Modelos de Bancos de Dados	16
FIG.2.2	Exemplo Modelo Entidade-Relacionamento (FIDELIX, 2019)	19
FIG.2.3	Exemplo Aplicação UML (FONSECA, 2011)	19
FIG.2.4	Esquema de Funcionamento Formato JSON (INTERNACIONAL, 2013)	20
FIG.2.5	Plugin DSG Tools no QGIS	21
FIG.2.6	Classes de Hidrografia na EDGV 3.0 (CONCAR, 2017)	23
FIG.2.7	Classes e Atributos na MGCP (MGCP TRD4 v4.4, 2016)	24
FIG.2.8	Tabela de Comparação MGCP TRD4 v4.4 x ET-EDGV 3.0	25
FIG.3.1	Interface Plugin <i>EDGV Converter</i>	27
FIG.3.2	Sequenciamento de atividades do <i>plug-in</i> (SANTOS & WOLFGRAM & ALMEIDA, 2019)	29
FIG.3.3	Aba Mapeamento Domínio	30
FIG.3.4	Janela Domínio	30
FIG.3.5	Aba Mapeamento de Classe	31
FIG.3.6	Janela Atributos	31
FIG.3.7	Aba Mapeamentos (Log)	32
FIG.3.8	Parte do Código do Banco de Dados TRD4 v4.4 Utilizado	33
FIG.3.9	Upload do Banco de Dados no PgAdmin	33
FIG.3.10	Banco de Dados TRD4 Carregado no Plugin	34
FIG.3.11	Upload do Banco de Dados no PgAdmin	35
FIG.3.12	Arquivo JSON utilizado no desenvolvimento do <i>plugin</i>	36
FIG.3.13	Classes e atributos de ida e volta no arquivo JSON	37
FIG.3.14	Tradução de atributos no arquivo JSON	38
FIG.3.15	Cronograma Adotado para a Pesquisa	39

LISTA DE SIGLAS

EDGV	Especificações Técnicas para Estruturação de Dados Geoespaciais Vetoriais
MGCP	Multinational Geospatial Co-Production Program
CAD	Computer Aided Design
MTD	Mapoteca Topográfica Digital
TBCD	Tabela da Base Cartográfica Digital
DSG	Diretoria de Serviço Geográfico
IBGE	Instituto Brasileiro de Geografia e Estatística
CEPAD	Comitê Especializado para Estudo do Padrão de Intercâmbio de Dados Cartográficos Digitais
MMA	Ministério do Meio Ambiente
CONCAR	Comissão Nacional de Cartografia
CEMND	Comitê de Estruturação da Mapoteca Nacional Digital
EUA	Estados Unidos da América
NGA	National Geospatial Agency
IBM	International Business Machines Corporation
SIG	Sistema de Informação Geográfica
OSGeo	Open Source Geospatial Foundation
IDE	Integrated Development Environment
INDE	Infraestrutura Nacional de Dados Espaciais
SCN	Sistema Cartográfico Nacional
UML	Unified Modeling Language
JSON	JavaScript Object Notation

RESUMO

Atualmente no Brasil existem cartas representativas do território nacional que foram desenvolvidas com o padrão conhecido como EDGV (Especificação Técnica para a Estruturação de Dados Geoespaciais Vetoriais), na versão 2.1.3, e estão sendo atualizadas para a versão mais recente, a 3.0. Por outro lado, no cenário internacional, foi formado um grupo com diferentes nações participantes, que visa a cooperação para o mapeamento de todo o território do globo, e é conhecida como MGCP (Multinational Geospatial Co-production Program). O Brasil, como participante deste grupo internacional, necessita produzir cartas que sigam o padrão estabelecido pela MGCP para as classes e atributos, e para isso, esta pesquisa visa a construção de um plugin no software QGIS que realize a conversão entre as regras. Com isso, irá contribuir para que o país tenha participação relevante como membro do grupo.

ABSTRACT

Currently, in Brazil, there are charts representing the national territory that were developed with the standard known as EDGV (Technical Specification for Structuring Vector Geospatial Data), in version 2.1.3, and are being updated to the latest version, 3.0. On the other hand, on the international scene, a group was formed with different participating nations, which aims to cooperate to map the entire territory of the globe, and is known as MGCP (Multinational Geospatial Co-production Program). Brazil, as a participant in this international group, needs to produce charts that follow the standard established by MGCP for classes and attributes, and for this, this research aims to build a plugin in the QGIS software that performs the conversion between the rules. This will help the country to have a relevant participation as a member of the group.

1 INTRODUÇÃO

1.1 OBJETIVO

Este projeto tem como objetivo, desenvolver um plugin para ser usado no software QGIS, que realizará a conversão entre classes e atributos do padrão ET-EDGV 3.0 (Especificações Técnicas para Estruturação de Dados Geoespaciais Vetoriais), para o MGCP (Multinational Geospatial Co-Production Program) TRD4 (Technical Reference Documentation) v4.4. Esse mapeamento será realizado utilizando como auxílio recursos de computação e programação, disponíveis no QGIS.

1.2 ESPECIFICAÇÕES TÉCNICAS PARA ESTRUTURAÇÃO DE DADOS GEOESPACIAIS VETORIAIS (ET-EDGV)

No fim dos anos 80, todos os processos de produção cartográfica no Brasil, eram analógicos, e foi no início da década de 1990 que surgiram as primeiras automações destes processos baseados em CAD (Computer Aided Design), ao mesmo tempo em que estavam sendo adaptadas as normas técnicas para o meio digital para que os dados produzidos fossem validados, de acordo com as regras topológicas (CONCAR, 2010).

A MTD (Mapoteca Topográfica Digital)/IBGE (Instituto Brasileiro de Geografia e Estatística) e a TBCD (Tabela da Base Cartográfica Digital)/DSG (Diretoria de Serviço Geográfico) iniciaram a estruturação de dados espaciais vetoriais, e a partir deste momento, cada órgão de mapeamento desenvolveu suas próprias estruturas para as atividades de mapeamento. Com isso, as bases geradas pela DSG e IBGE nesse período estavam em estruturas diferentes, tornando necessária a conversão entre os padrões para que os produtos sejam utilizados em conjunto (CONCAR, 2010).

Essa diferença entre os produtos geoespaciais da DSG e IBGE gerou certa dificuldade de compatibilização para um padrão único dos dados, pois existiam diferentes categorias, semânticas para certas categorias, feições e diferentes atributos por feição (CONCAR, 2010).

Identificadas as diferenças e a necessidade de se criar um padrão único, em junho de 1997, criou-se o CEPAD (Comitê Especializado para Estudo do Padrão de Intercâmbio de Dados Cartográficos Digitais), que visava desenvolver um padrão para os dados cartográficos digitais a serem gerados pelas organizações. As reuniões do CEPAD foram

realizadas até o final do ano de 1998 (GALVÃO, W. P. 2017).

Com o fim das reuniões do CEPAD, os próprios órgãos DSG e IBGE realizaram reuniões com o objetivo de obter um padrão único para os modelos MTD e TBCD, porém, a falta de recursos financeiros, para as despesas com deslocamento de pessoal, impediram que o objetivo final fosse alcançado (CONCAR, 2010).

O MMA (Ministério do Meio Ambiente), em 2004, junto ao CONCAR (Comissão Nacional de Cartografia), realizaram auditorias técnicas em bases digitais de empresas privadas, para verificar a adequação das mesmas ao SCN (Sistema Cartográfico Nacional). Essas auditorias visavam compor, com cartas na escala 1:100.000, a base cartográfica digital da Amazônia Legal (CONCAR, 2010).

A partir deste momento, iniciou-se a proposta da parceira do MMA com a DSG e IBGE, para que juntassem esforços e desenvolvessem um mesmo padrão de estrutura de dados (CONCAR, 2010). Esse padrão seria aplicável às bases digitais da Amazônia Legal, que na época, não seguiam padrão único.

Com o objetivo de definir um único padrão de estrutura de dados geoespaciais, usou-se como base a modelagem orientada a objetos desenvolvida pela DSG, a EGB2000. Nesta, foram incorporados conceitos da TBCD (Tabela da Base Cartográfica Digital) e MTD (Mapoteca Topográfica Digital), atendendo aos requisitos solicitados para a base cartográfica pretendida pelo MMA.

A estrutura desenvolvida foi proposta em 2005, para a estruturação da Mapoteca Nacional Digital. Em 2006, a CONCAR aprovou a versão 1.0 como norma provisória até que outra versão mais completa fosse desenvolvida, o que mais tarde, no ano de 2008, resultou na ET-EDGV 2.0 (CONCAR, 2010).

Com a publicação da nova versão da ET-EDGV, o SCN iniciou junto a seus órgãos, a adaptação dos dados geoespaciais para a versão 2.0. No mesmo período, a CEMND (Comitê de Estruturação da Mapoteca Nacional Digital) desenvolveu melhorias na estrutura de dados, no que culminou na versão 2.1 da ET-EDGV (CONCAR, 2010).

1.3 MULTINATIONAL GEOSPATIAL CO-PRODUCTION PROGRAM (MGCP)

O MGCP é um programa de mapeamento que tem por objetivo mapear todo o globo terrestre, organizado pela NGA. Conta atualmente com 32 nações membros, que elaboram produtos vetoriais nas escalas de 1:50.000 e 1:100.000, em células de 1° x 1°. O método de mapeamento difere em vários aspectos do feito pela DSG, pelo fato de seguir normas diferentes (CASTRO, 2019).

Esse banco de dados GIS conta com a especificação das escalas de 1: 50.000 e 1: 100.000, que atende às necessidades nacionais e internacionais (C. JÓZSEF & M. OLÍVIA, 2009).

Para a produção, as fontes das ortofotografias feitas com base em fotografias aéreas e materiais de sensoriamento remoto disponíveis precisam ser recentes, e todas as nações participantes financiam a implementação das premissas a partir de seu próprio orçamento (C. JÓZSEF & M. OLÍVIA, 2009).

1.3.1 TECHNICAL REFERENCE DOCUMENTATION (TRD)

O TRD é uma especificação técnica que visa a padronização de trabalho e metodologias para a produção de dados geospaciais, e atualmente, é de responsabilidade da França. Como referência para uso no trabalho, foi utilizada a TRD4 v4.4, que é constituída de toda a documentação necessária para o mapeamento, assim como todas as estruturas em formato .gml/.xml (MGCP_XSD_TRD4), modelo semântico (*Semantic Information Model*), regras de aquisição (*Extraction Guide*), especificações de metadados (*Metadata Specification*), padrões de qualidade (*QA Cookbook*), entre outras (CASTRO, 2019).

1.4 MOTIVAÇÃO DA PESQUISA

O Brasil, atualmente, é um dos países com a posição de membro da MGCP, e para que continue fazendo parte do grupo, é necessário fornecer cartas, nas especificações previamente explicitadas (DSG, 2019).

Para que essas entregas sejam feitas, é necessário que as cartas existentes hoje sejam convertidas do modelo da ET-EDGV para o MGCP, e para isso, deve ser feito um trabalho de mapeamento entre classes e atributos de um para o outro.

A elaboração de um plugin com esta funcionalidade, proporcionará considerável economia de tempo para as equipes de cartografia do Exército Brasileiro, pois onde seria usada a mão de obra disponível para executar a conversão de cada carta a ser enviada, será feito de forma automática e computacional.

A utilização desta solução proporcionará também participação mais ativa do Brasil dentro da comunidade do MGCP, pois simplificará a padronização das cartas, e com isso, ocorrerão mais entregas por parte do país.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 BANCOS DE DADOS

Um banco de dados é uma coleção de dados persistentes, usada pelos sistemas de aplicação de uma determinada empresa (DATE, 2004). Funciona como um local onde são armazenados dados para determinadas atividades, sendo fonte de dados para aplicações atuais e as que vierem a existir, sendo conjunto organizado de dados relacionados, criado com determinado objetivo e que atende uma comunidade de usuários (COSTA, 2011).

Pode ser descrito como uma coleção logicamente coerente de dados com algum significado inerente. Uma variedade aleatória de dados não pode ser corretamente chamada de banco de dados (ELMASRI & NAVATHE, 2011).

Um banco de dados é projetado, construído e projetado com dados para uma finalidade específica. Possui grupo definido de usuários e aplicações previamente concebidas, nas quais os usuários estão interessados (ELMASRI & NAVATHE, 2011).

Para que se entenda o conceito de Modelo Conceitual, é importante que se conheça primeiro os conceitos de Entidade, Atributo e Relacionamento. A figura abaixo mostra a representação usual desses conceitos, que são explicados de nos subcapítulos subsequentes.

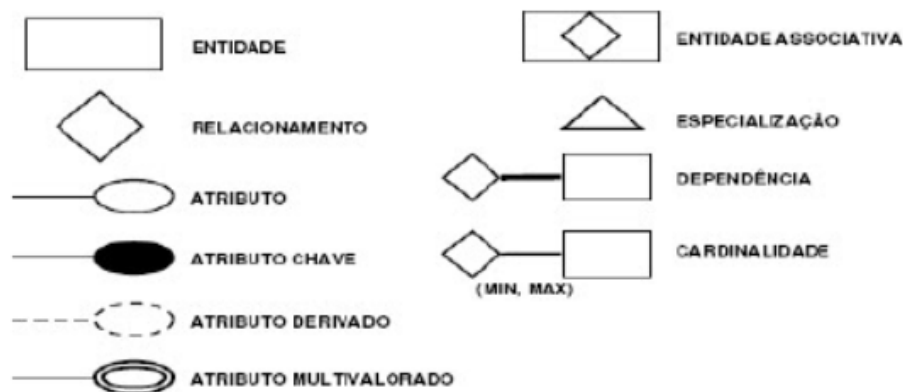


FIG. 2.1: Notações Utilizadas em Modelos de Bancos de Dados

2.1.1 ENTIDADES

Define-se entidade como um objeto que existe no mundo real com uma identificação distinta e com um significado próprio. São objetos e partes envolvidas em um domínio, e classificados de acordo com sua existência. As nomenclaturas adotadas para as entidades,

são substantivos concretos ou abstratos, que representem de forma clara sua função dentro do domínio (SIQUEIRA, 2016).

As entidades podem se classificar em 3 (três) grupos, de acordo com o motivo de sua existência, que são eles (BALAGUER, 2016):

- Entidades Fortes: aquelas que independem de outras, que sozinhas possuem sentido completo.
- Entidades Fracas: aquelas que dependem de outras, pois individualmente, não fazem possuem sentido completo.
- Entidades Associativas: quando há a necessidade de associar uma entidade a um relacionamento. Na modelagem Entidade-Relacionamento, se faz isso para tornar um relacionamento, uma entidade associativa, que se relaciona com outras entidades.

2.1.2 RELACIONAMENTO DE ENTIDADES

Para a montagem de um banco de dados, é preciso que sejam definidos os relacionamentos entre as entidades consideradas. De acordo com a quantidade de objetos envolvidos em cada lado do relacionamento, eles podem ser classificados de 3 (três) formas (BALAGUER, 2016):

- Relacionamento 1..1 (um para um): acontece quando uma das duas entidades envolvidas, referenciam uma, e apenas uma, unidade da outra.
- Relacionamento 1..n ou 1..* (um para muitos): é um relacionamento em que uma entidade referencia muitas unidades da outra, e essas várias, podem referenciar apenas uma unidade da origem.
- Relacionamento n..n ou *..* (muitos para muitos): cada entidade, de ambos os lados, podem referenciar múltiplas unidades da outra.

2.1.3 ATRIBUTOS

São interpretados como característica de uma entidade que tem valor para a aplicação, que deve ser observada pelo usuário. A definição dos atributos devem ser discutidas antes de se começar o desenvolvimento, para que todo atributo que necessário, no presente ou no futuro, seja incluído. Estas discussões devem acontecer também, para que não se incluam dados desnecessários nos bancos de dados (SIQUEIRA, 2016).

Os atributos, assim como as entidades, podem ser classificados dependendo do tipo de informação que representam. São eles (SIQUEIRA, 2016):

- Atributo Monovalorado: assume valor único para cada elemento.
- Atributo Composto: formado por um ou mais sub-atributos. Nesse tipo de situação, é necessário que se analise a possibilidade de fragmentar este atributo em outros.
- Atributo Multivalorado: uma entidade possui diversos valores para este atributo.
- Atributo Determinante: identifica cada entidade de um conjunto (conhecido como atributo chave).

2.1.4 MODELO CONCEITUAL

A elaboração do desenho inicial do banco de dados, fase qual não se prende a detalhes da forma como será implementado, são utilizados modelos de dados. Um modelo de dados é uma coleção de conceitos que podem ser usados para descrever a estrutura de um banco de dados (COSTA, 2011).

Existem certos modelos conhecidos e amplamente utilizados, que são o Modelo Conceitual, o Modelo de Implementação e o Modelo Físico, sendo que para esta pesquisa, é importante que se entenda o conceito de modelagem conceitual. Este, é o mais próximo do usuário final. Entidades, atributos e relacionamentos são alguns dos conceitos utilizados (ELMASRI & NAVATHE, 2011).

Um exemplo de modelo conceitual, é o conhecido como Modelo Entidade-Relacionamento. Este, foi desenvolvido para simplificar o projeto de banco de dados, permitindo especificação e representação da estrutura e lógica geral de um banco de dados (SILBERSCHATZ & KORTH & SUDARSHAN, 2006).

Peter Chen, em 1976, definiu o Modelo Entidade-Relacionamento, usando como base a teoria relacional criada por E.F.Codd em 1970. Segundo Chen, a visão da realidade se baseia no relacionamento entre os conceitos desta, que retratam os fatos dessa mesma realidade. Afirmou também que cada conceito (entidade ou relacionamento), pode possuir atributos (SIQUEIRA, 2016).

A figura abaixo ilustra uma aplicação dos conceitos da modelagem descrita neste capítulo, onde pessoas são relacionadas a projetos nos quais irão trabalhar. Neste, são aplicados as representações visuais contidas nas figura 2.1, e os conceitos dos subcapítulos 2.1.1, 2.1.2 e 2.1.3.

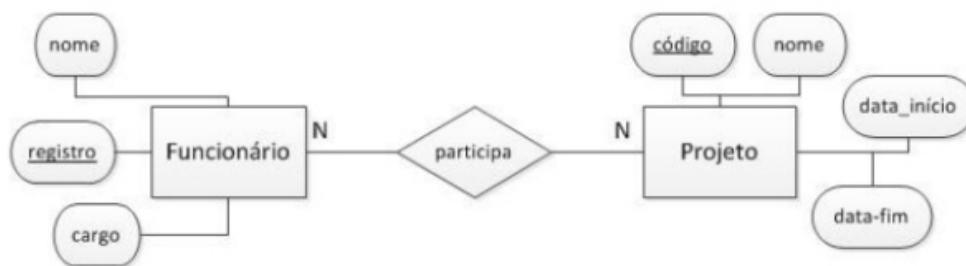


FIG. 2.2: Exemplo Modelo Entidade-Relacionamento (FIDELIX, 2019)

2.1.5 DIAGRAMA DE CLASSES

Um diagrama de classes é uma representação da estrutura e relações das classes elaboradas para o modelo. De acordo com (MELO, 2004), é um conjunto de objetos com as mesmas características, e assim, fica possível identificar objetos e agrupá-los, de forma a encontrar suas respectivas classes.

A UML (Unified Modeling Language), em diagrama de classes, é uma linguagem única, de modelagem de um software ou de uma aplicação. A UML gera artifícios visuais, que são diagramas que auxiliam na modelagem de uma classe. Sua representação se dá por um retângulo com três divisões, que são: O nome da classe, seus atributos e por fim os métodos (DEV MEDIA, 2018).

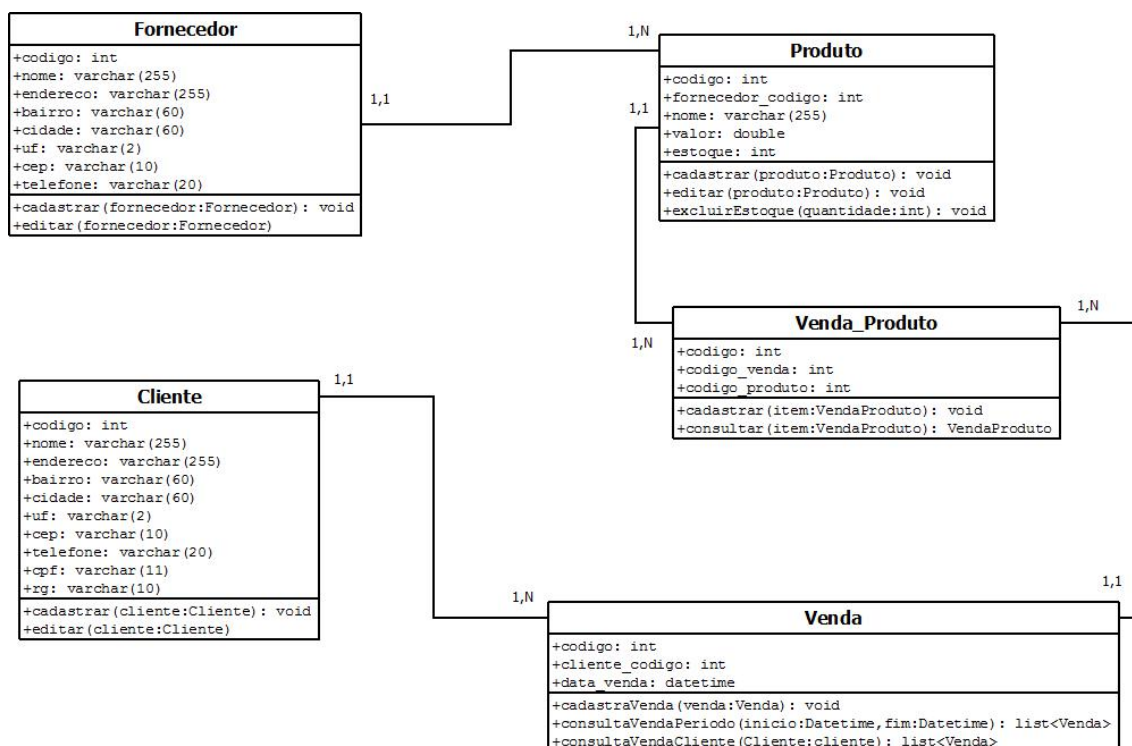


FIG. 2.3: Exemplo Aplicação UML (FONSECA, 2011)

Então, de acordo com (FONSECA, 2011), um Diagrama de Classes, pode ser definido como a representação de uma coleção de classes e seus inter-relacionamentos. A figura acima mostra um exemplo de uma aplicação do diagrama de classes UML. para o caso do caminho seguido por um produto, desde o fornecedor, até o cliente.

2.2 FORMATO JSON (*JAVASCRIPT OBJECT NOTATION*)

O JSON é um formato baseado em texto que possibilita a troca de dados independente da linguagem de programação utilizada (INTERNACIONAL, 2013). Quando se esta analisando troca de informações entre sistemas, que é o caso da presente pesquisa, onde foram convertidas classes e atributos no padrão da ET-EDGV 3.0 para o padrão TRD4, é evidente a importância da existência dessa tecnologia.

As estruturas deste formato, são flexíveis, e permitem trabalhar com dados sem a necessidade de se definir esquemas rígidos. No geral, um arquivo JSON obedece uma sequência de *tokens* do padrão *Unicode*. As estruturas existentes no formato são os colchetes ([]), chaves ({ }), dois pontos (:), e vírgula (,), além de *true*, *false* e *null* (MACHADO, 2017).



FIG. 2.4: Esquema de Funcionamento Formato JSON (INTERNACIONAL, 2013)

2.3 QGIS - MULTIPLATAFORMA DE SISTEMA DE INFORMAÇÃO GEOGRÁFICA

O software hoje chamado de QGIS, e anteriormente de Quantum GIS, é um *software* livre, que funciona como uma multiplataforma de SIG (Sistema de Informação Geográfica), que possui código-fonte aberto e licença pública geral. Foi projetado oficialmente pela OSGeo (Open Source Geospatial Foundation), e contém suporte para Linux, Unix, Mac OSX, Windows e Android (QGIS Project, 2020). É capaz de processar e interpretar inúmeros formatos de vetores, rasters, bases de dados e funcionalidades.

O desenvolvimento inicial do programa começou no ano de 2002, e após 5 anos, passou por processo de incubação por uma fundação ligada a projetos Open Source Geoespaciais, o que possibilitou que a primeira versão fosse lançada em 2009 (QGIS Project, 2020).

Desde então, o software passou a ser ajustado para que sua interface fosse mais simples e intuitiva para o usuário. Outro objetivo adotado pelo projeto na época, foi a independência quanto ao sistema operacional utilizado.

Diferentemente dos outros *softwares* disponíveis no mercado, o uso do QGIS não requer um computador com configurações avançadas e processador de geração recente. O programa é menos exigente quanto à composição técnica das máquinas, quando comparado com similares (VITAL, 2019).

Uma das características mais marcantes do QGIS é a extensibilidade de seus recursos através de plugins. Estes, são desenvolvidos em ambiente de linguagem Python, podem possuir diversas funcionalidades de geoprocessamento e são elaborados para ter funções que o QGIS não supre nativamente.

2.3.1 DSG TOOLS - PLUGIN

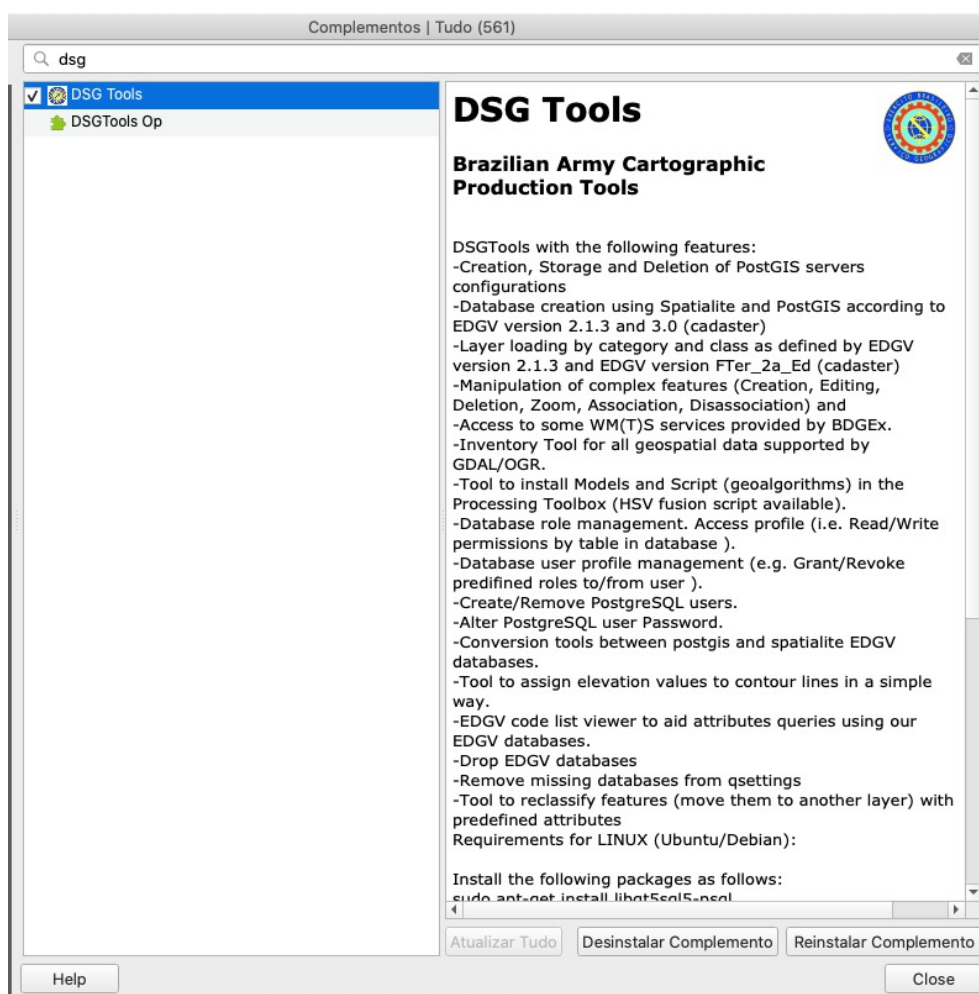


FIG. 2.5: Plugin DSG Tools no QGIS

A DSG Tools é um plugin hospedado no *software* QGIS, e é totalmente gratuito e aberto. Foi desenvolvido pela DSG, e permite a produção de dados geoespaciais nos conformes da INDE (Infraestrutura Nacional de Dados Espaciais). Visa atender não apenas o Exército Brasileiro, mas também produtores e usuários de geoinformação da sociedade.

O plugin possibilita as atividades de aquisição de feições vetoriais a partir de imagens matriciais. As ferramentas do plugin deverão receber atualizações ser utilizado em outras áreas, como por exemplo, no processamento digital de imagens, na vetorização semiautomática, na validação topológica e de atributos e na impressão de cartas (PRATES, 2015).

2.4 PYTHON

Em 1991, Guido Van Rossum lançou a primeira versão da linguagem de programação Python, e o primeiro intuito era desenvolver linguagem de scripts, aplicada ao sistema operacional Amoeba, desenvolvido pelo CWI, empresa onde trabalhava na época (DEV-MEDIA, 2018).

O Python é uma linguagem fracamente tipada, o que significa que não é necessário declarar tipos das variáveis. Pode ser utilizada tanto no ambiente desktop, web ou mobile (SILVA, 2019).

Para criar e desenvolver projetos, é necessário que se use programas com ambientes e funcionalidades próprias para tal função, e são conhecidos como IDE. Algumas das mais utilizadas estão: Spyder, PyCharm, Visual Studio, entre outras (DEVMEDIA, 2018).

Sobre suas aplicações, onde o Python possui vantagens em relação a outras linguagens, são por exemplo, desenvolvimento Web, scripting e ciência de dados. Todos os exemplos citados, são área da computação amplamente explorados nos dias atuais, pois são aplicáveis à robótica, segurança da informação e outros assuntos relevantes para sociedade contemporânea (BORGES, 2010).

Como o objetivo do presente projeto é o desenvolvimento de um plugin no *software* QGIS, e este disponibiliza ambiente de programação em *Python*, o conhecimento de suas funcionalidades e aplicações, é relevante para a realização do projeto.

2.5 ESTRUTURAS DE CLASSES E ATRIBUTOS DOS PADRÕES CONSIDERADOS

Conforme consta no objetivo deste documento, a finalidade é a construção de um plugin que realize o mapeamento de classes e atributos do padrão EDGV 3.0 para o MGCP

TRD4 v4.4. Para isso, nessa seção, serão apresentadas as formas que cada um desses padrões oferece para os usuários.

No caso da EDGV 3.0, os dados geoespaciais foram organizados nas chamadas Classes de Objetos, de acordo com semelhanças nas funcionalidades, e que são compostas por Atributos que as caracterizam. Quando agrupadas, essas classes formam Categorias de Informação (CONCAR, 2017).

A referida versão da EDGV, conta com 180 classes e 1706 atributos. Tais classes estão divididas em dois conjuntos: Mapeamento Topográfico para Pequenas Escalas (Map-TopoPE) para escalas iguais ou menores que 1:25.000 e Mapeamento Topográfico para Grandes Escalas (MapTopoGE) para escalas entre 1:25.000 e 1:1.000 (CASTRO, 2019).

Como exemplo, na figura abaixo, está apresentado exemplo da categoria Hidrografia e suas classes contidas na ET-EDGV 3.0, e que possui um total de 20 (vinte) classes.

03. Hidrografia

Classe Código na RCO	1:1.000	1:10.000	1:25.000	1:50.000	1:100.000	1:250.000
Area_Umida 1.3.1	X	X	X	X	X	X
Banco_Areia 1.3.2	X	X	X	X	X	X
Barragem 1.3.3	X	X	X	X	X	X
Canal 1.3.4	X	X	X	X	X	X
Canal_Vala 1.3.5	X	X	X	X	X	X
Comporta 1.3.6	X	X	X	-	-	-
Corredeira 1.3.7	X	X	X	X	X	-
Dique 1.3.8	X	X	X	X	X	X
Fonte_Dagua 1.3.9	X	X	X	X	X	-
Foz_Maritima 1.3.10	X	X	X	X	X	X
Ilha 1.3.11	X (1)	X (1)	X (1)	X (1)	X (1)	X (1)
Massa_Dagua 1.3.12	X	X	X	X	X	X
Quebramar_Molhe 1.3.13	X	X	X	X	X	X
Queda_Dagua 1.3.14	X	X	X	X	X	X
Recife 1.3.15	X	X	X	X	X	X
Rocha_Em_Agua (2) 1.3.16	X	X	X	X	-	-
Sumidouro_Vertedouro 1.3.17	X	X	X	X	X	X
Terreno_Sujeito_ Inundacao 1.3.18	X	X	X	X	X	X
Trecho_Drenagem 1.3.19	X	X	X	X	X	X
Vala 1.3.20	X	X	X	X	X	X

FIG. 2.6: Classes de Hidrografia na EDGV 3.0 (CONCAR, 2017)

De acordo com o contido nas especificações publicadas pelo CONCAR em 2017, observa-se que cada classe possui código numérico único, com 4 dígitos, que as individualiza, e os relaciona com sua respectiva categoria.

No caso da TRD, os dados geoespaciais são arranjados de forma similar, ainda em classes e atributos, porém com suas particularidades em relação à EDGV. A primeira grande diferença entre as duas é o idioma, pois o adotado para o programa foi o inglês, enquanto que, para o padrão brasileiro, o português. Como consequência deste fato, as classes e atributos acabam por possuir nomes totalmente diferentes, dificultando o relacionamento entre os modelos.

Uma segunda diferença relevante, é que assim como na EDGV, as classes possuem códigos, cada atributo também recebe uma codificação específica, porém os atributos possuem apenas três dígitos textuais, conforme pode ser observado na figura abaixo:

Beach Area Feature	
Definition	On a shore, the area on which the waves break and over which shore debris (for example: sand, shingle, and/or pebbles) accumulate. (A beach includes backshore and foreshore.)
DFDD Code	BA050
Spatial representation	Area

LIST OF FEATURE ATTRIBUTES			
INHERITED ATTRIBUTES (goto supertype)			
DFDD [MGCP] Code	Name	Type	Unit
[GEOM]	Spatial representation of the feature	GM_Surface	
[NAM]	Name	CharacterString	
[NFI]	Named Feature Identifier	CharacterString	
[NFI]	Name Identifier	CharacterString	
[SMC]	Surface Material Type	CodeList	

FIG. 2.7: Classes e Atributos na MGCP (MGCP TRD4 v4.4, 2016)

A referida versão da MGCP, possui 196 classes e 93 atributos, sendo que este valor não leva em conta a primitiva geométrica de cada uma delas. Após analisar o catálogo de classes e atributos da TRD4, é importante que se entenda como as nomenclaturas foram estruturadas. O primeiro dígito textual determina a qual categoria a classe pertence. A divisão acontece da seguinte forma (CASTRO, 2019):

- Classes iniciadas com o dígito textual A são da categoria que representa construções que não envolvam hidrografia.
- Classes iniciadas com o dígito textual B são da categoria de hidrografia.
- Classes iniciadas com o dígito textual D são da categoria que representa elementos fisiográficos.
- Classes iniciadas com o dígito textual E são da categoria que representa vegetação.

- Classes iniciadas com o dígito textual F são da categoria que representa campos e estandes de tiro.
- Classes iniciadas com o dígito textual G são da categoria que representa objetos geográficos aeroespaciais.
- Classes iniciadas com o dígito textual S são da categoria que representa instalações militares.
- Classes iniciadas com o dígito textual Z são da categoria que representa vazios cartográfico e nome local.

Com essas informações acerca das estruturas de cada um dos padrões considerados, é possível perceber que existem diferentes números de atributos e classes entre estes, assim como as escalas consideradas. Estas diferenças, influenciam diretamente na correspondência entre os mapeamentos considerados, podendo conter classes e atributos na EDGV 3.0 sem representação existente na MGCP TRD4 v4.4.

Segue abaixo uma tabela resumo de comparação entre os padrões considerados, que identifica as principais diferenças apontadas.

Informação	MGCP TRD4 v4.4	ET-EDGV 3.0
Idioma	Inglês	Português
# Classes	196	180
# Atributos	93	1706
Codificação de Classes	2 dígitos textuais + 3 dígitos numéricos	Código numérico de até 4 dígitos
Codificação dos Atributos	3 dígitos textuais	Sem regra específica
Escalas	1:50.000 e 1:100.000	Possui especificações para escalas grandes e pequenas

FIG. 2.8: Tabela de Comparação MGCP TRD4 v4.4 x ET-EDGV 3.0

2.6 CONVERSÕES DE MODELO

Ao longo da realização dessa pesquisa, foram utilizados produtos gerados em outros projetos como forma de embasamento e auxílio ao desenvolvimento da pesquisa. Os produtos utilizados foram um mapeamento de classes e atributos entre dois modelos, e um plugin de conversão, previamente desenvolvido.

O mapeamento de classes e atributos utilizado, foi o produto gerado na pesquisa "Mapeamento Entre as Especificações MGCP TRD4 v4.4 e ET-EDGV 3.0", elaborado por (CASTRO, 2019). A finalidade da pesquisa foi elaborar um arquivo de mapeamento

que permita, de forma estruturada, a conversão entre as especificações da TRD4 v4.4 e a ET-EDGV 3.0 (CASTRO, 2019).

O plugin de conversão utilizado foi o *EDGV Converter*, desenvolvido por (SANTOS & WOLFGRAM & ALMEIDA, 2019), hospedado no *software* QGIS. Este plugin realiza a conversão de dados entre a EDGV versão 2.1.3 e a EDGV versão 3.0, de modo que o usuário possa realizar a conversão entre estas versões da EDGV em seu próprio banco de dados em um tempo reduzido e com o uso de softwares livres. A estrutura de código previamente existente no plugin foi usada como base para o desenvolvimento do proposto nesta pesquisa.

3 METODOLOGIA

3.1 PLUGIN - PARTE COMPUTACIONAL

A parte computacional da presente pesquisa envolveu o desenvolvimento de um plugin, hospedado no repositório de plugins do QGIS, que realize a conversão de um banco de dados com classes e atributos no padrão da ET-EDGV 3.0, para o padrão adotado no TRD4 v4.4 da MGCP.

Para que não fosse necessário o desenvolvimento completo de um novo código, foi utilizado como base o plugin chamado *EDGV Converter*, desenvolvido por (SANTOS & WOLFGRAM & ALMEIDA, 2019). O código foi adaptado para interpretar os arquivos JSON de conversão entre os modelos EDGV 3.0 e TRD4 utilizado.

3.1.1 PLUGIN *EDGV CONVERTER*

3.1.1.1 INTERFACE DO PLUGIN

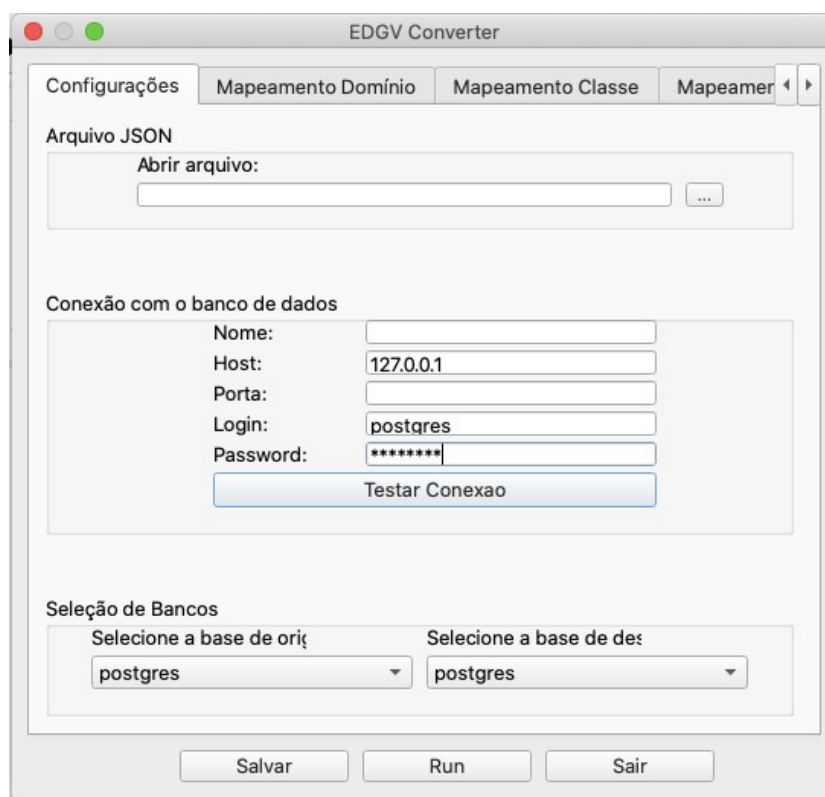


FIG. 3.1: Interface Plugin *EDGV Converter*

A interface deste plugin possui 4 (quatro) abas. Na aba de configurações, é possível incluir o arquivo JSON que será usado na conversão, assim como estabelecer a conexão com o banco de dados contendo as informações a serem convertidas.

Ainda na aba configurações, é possível escolher em qual padrão está a base de origem, e qual será a base de destino a ser considerada para a conversão.

As outras 3 (três) abas são destinadas à realização do mapeamento entre as classes e atributos das bases consideradas. No subcapítulo a seguir, estão explicados os funcionamentos das atividades relacionadas com cada aba.

3.1.1.2 FUNCIONAMENTO DO PLUGIN

Primeiramente, é importante que se saiba o que é necessário que o usuário tenha em posse para que seja possível a utilização do plugin. São 6 (seis) requisitos a serem atendidos (SANTOS & WOLFGRAM & ALMEIDA, 2019):

- *Software* QGIS versão 3.0 ou mais recente
- Python 3
- Versão 9.0 ou superior do PostgreSQL
- PostGIS
- Qt Designer
- Deve possuir 2 bancos de dados com as estruturas dos modelos a serem convertidos de forma automática. No caso do plugin original, a conversão é feita entre as versões 2.1.3 e 3.0 da ET-EDGV

Checados os 6 (seis) requisitos acima, é preciso entender como é o funcionamento do *plugin*, que obedece um sequenciamento de atividades que explicam o passo-a-passo para utilização deste.

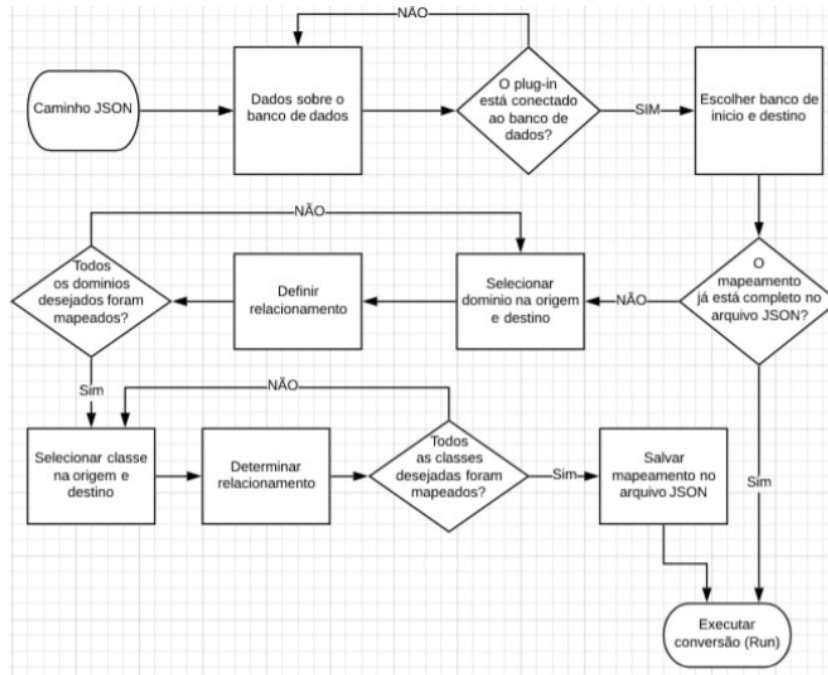


FIG. 3.2: Sequenciamento de atividades do *plug-in* (SANTOS & WOLFGRAM & ALMEIDA, 2019)

De acordo com a figura acima, é possível que o usuário utilize arquivos JSON com mapeamentos prontos, pois o *plugin* é capaz de fazer as alterações no arquivo fornecido, e torna possível fazer mapeamentos em partes, sem a necessidade de diversas versões do arquivo original. Esta funcionalidade facilita a realização de mapeamentos extensos.

A interface original, contém 4 (quatro) abas, onde cada uma possui funcionalidades a serem realizadas, que estão relacionadas abaixo (SANTOS & WOLFGRAM & ALMEIDA, 2019):

- A aba "Configurações" possui 3 atividades a serem realizadas, e são elas: seleção do arquivo JSON a ser utilizado, teste de conexão com banco de dados e determinar das bases de dados de origem e destino
- A aba "Mapeamento Domínio" possibilita mapear algo que não foi contemplado no arquivo JSON inserido.

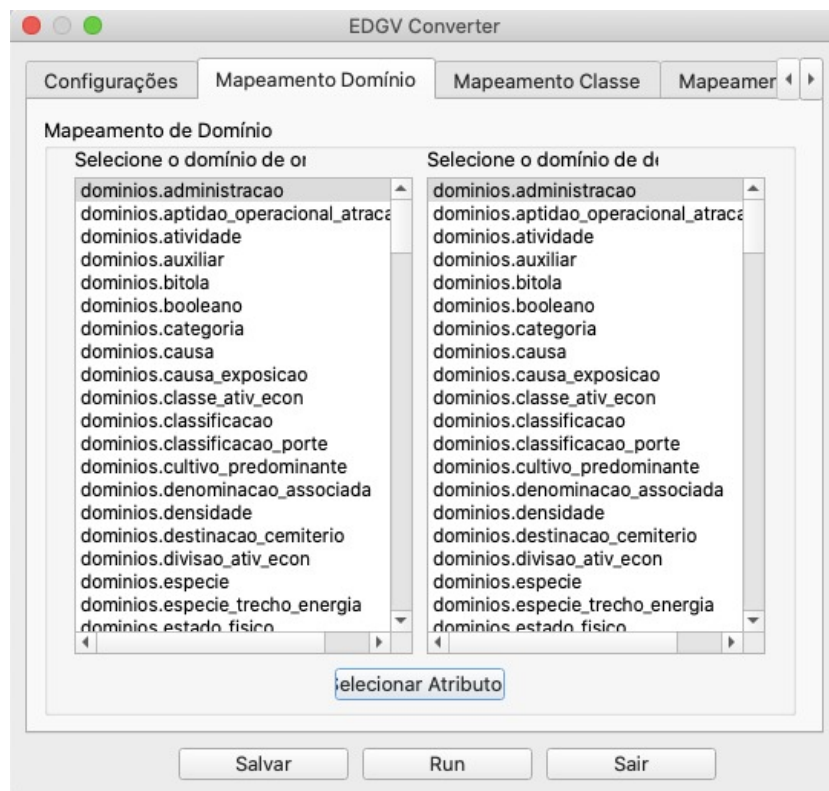


FIG. 3.3: Aba Mapeamento Domínio

Após o clique em "Selecionar Atributo", irá abrir uma janela chamada "Domínio", onde é possível que o usuário faça a relação entre os valores do domínio da base de origem para base de destino.

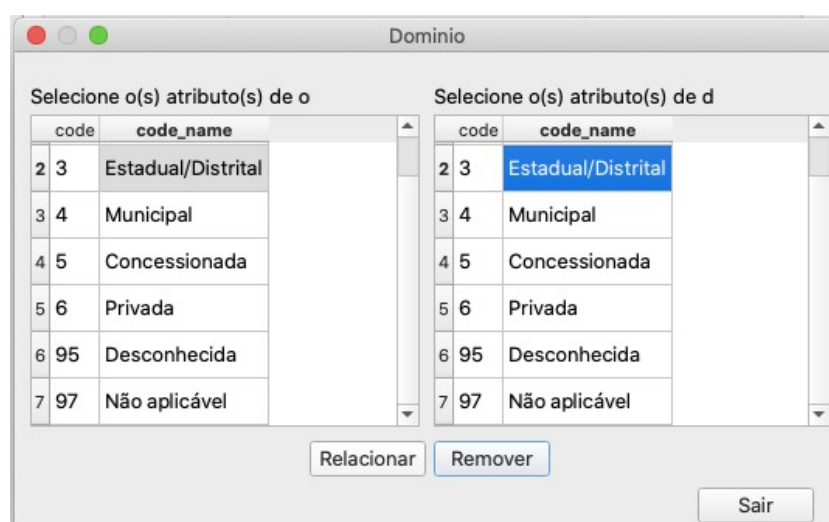


FIG. 3.4: Janela Domínio

- Na aba "Mapeamento de Classe", o usuário deverá fazer o relacionamento de classes, onde irá selecionar uma classe no campo de origem e sua correspondente no de

destino.

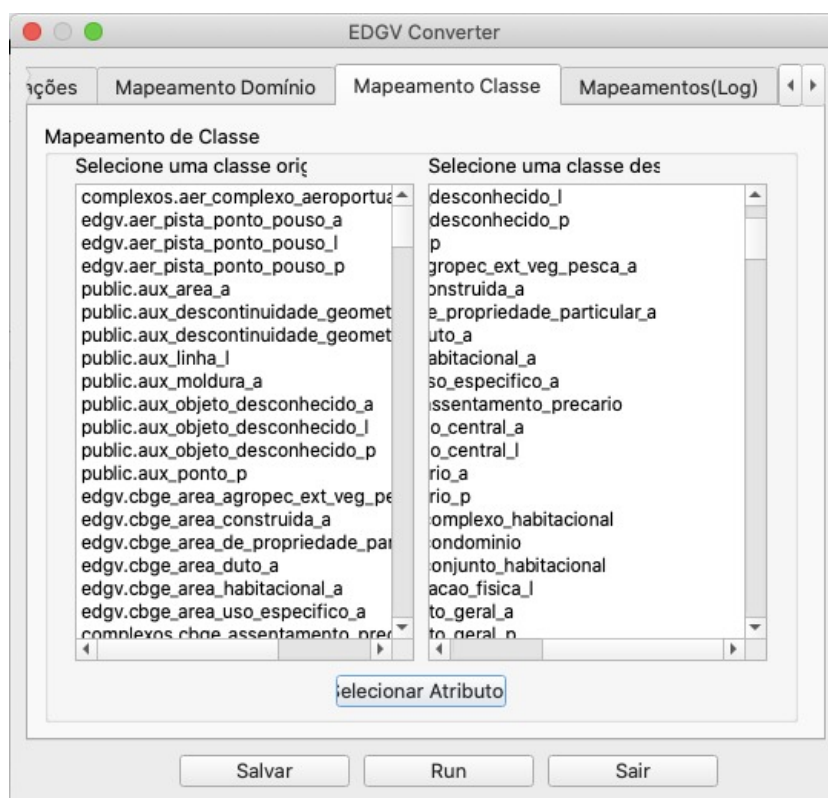


FIG. 3.5: Aba Mapeamento de Classe

Após clicar em "Selecionar Atributos", abrirá uma janela chamada "Atributos", onde serão relacionados os atributos que compõem as classes.

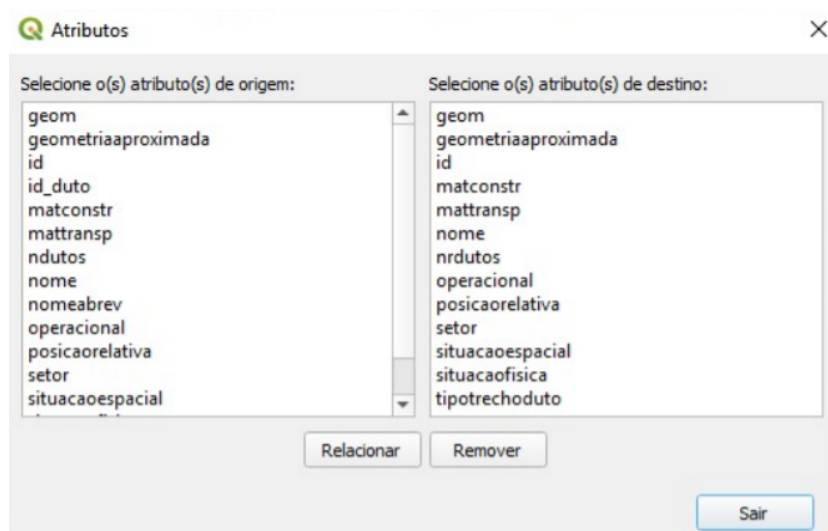


FIG. 3.6: Janela Atributos

■ A quarta e última aba é a chamada "Mapeamentos (Log)", que é a tela de registros,

onde o usuário pode checar os mapeamentos previamente executados. Essa aba auxilia na utilização do *plugin*.

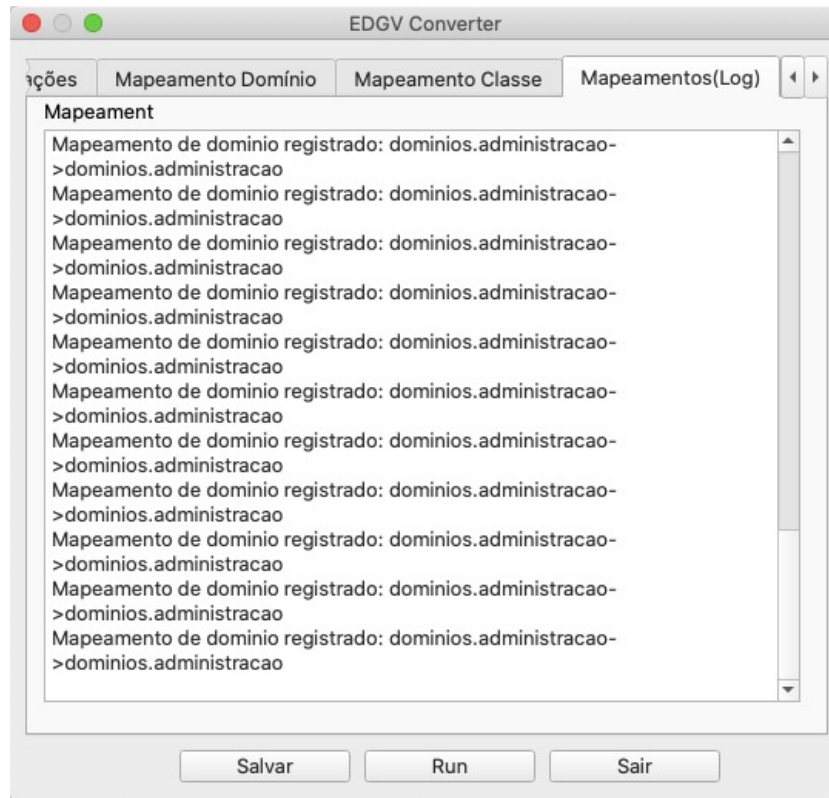


FIG. 3.7: Aba Mapeamentos (Log)

3.1.2 ADAPTAÇÕES NECESSÁRIAS NO PLUGIN

Abc.

3.2 BANCOS DE DADOS UTILIZADOS

3.2.1 BANCO DE DADOS TRD4 V4.4

O banco de dados da TRD4 v4.4 utilizado, foi desenvolvido em SQL, linguagem computacional destinada a desenvolvimento de bancos de dados, pela DSG. Este foi criado usando esta linguagem, e as tabelas necessárias inseridas, conforme segue abaixo:

```

CREATE TABLE domains.tipo_veg (
    code smallint NOT NULL,
    code_name text NOT NULL,
    CONSTRAINT tipo_veg_pk PRIMARY KEY (code)
);

INSERT INTO domains.tipo_veg (code,code_name) VALUES (101,'Extração mineral');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (102,'Brejo (sem árvores, pouca humidade)');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (103,'Brejo (sem árvores, muita humidade)');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (104,'Pântano (com árvores, muita humidade)');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (105,'Arroz');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (106,'Terreno exposto');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (107,'Vegetação cultivada');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (108,'Pomar');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (109,'Vinhedo');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (110,'Lúpulo');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (111,'Campo');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (112,'Cana');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (113,'Arbustos');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (114,'Floresta');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (115,'Desmatamento');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (116,'Mangue');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (117,'Clareira');
INSERT INTO domains.tipo_veg (code,code_name) VALUES (9999,'A SER PREENCHIDO');

```

FIG. 3.8: Parte do Código do Banco de Dados TRD4 v4.4 Utilizado

Para que se possa utilizar este no plugin, é preciso que seja feito o *upload* no pgAdmin, que é um software gráfico para administração de banco de dados.

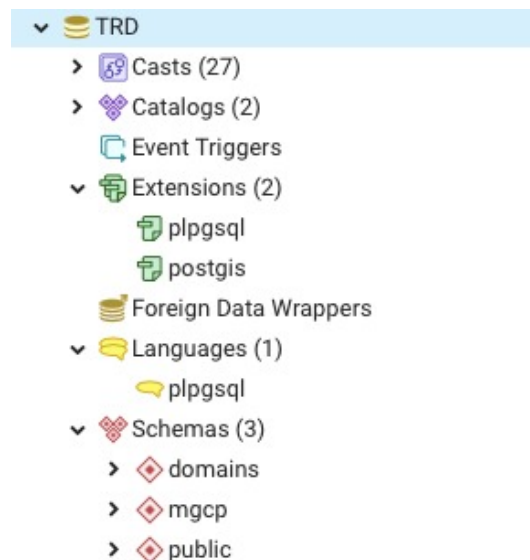


FIG. 3.9: Upload do Banco de Dados no PgAdmin

Após feito o *upload* do banco de dados no pgAdmin, seguindo os passos descritos no capítulo 3.1.1.2, é possível realizar o login através do *plugin*, e realizar o carregamento do banco de dados neste.

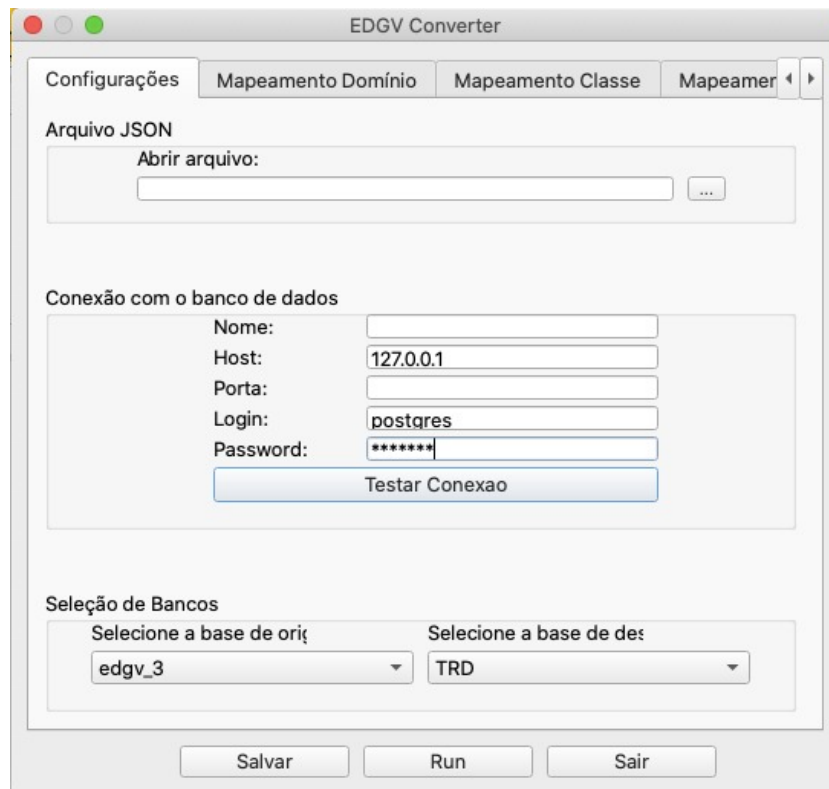


FIG. 3.10: Banco de Dados TRD4 Carregado no Plugin

Para o desenvolvimento deste banco de dados, foram criadas 392 tabelas, separadas em 3 *schemas*, e está disponível no repositório do GitHub, no link https://github.com/thiago-mpm/pfc_2020_MB.

3.2.2 BANCO DE DADOS ET-EDGV 3.0

O banco de dados da ET-EDGV 3.0 utilizado, foi desenvolvido por (SANTOS & WOLFGAM & ALMEIDA, 2019), e serviu de base para esta pesquisa. Para sua criação foi usada também a linguagem SQL, em estrutura semelhante ao banco de dados da TRD4 v4.4.

Após realizar a conexão com o banco de dados através do pgAdmin, é possível visualizar informações acerca deste:

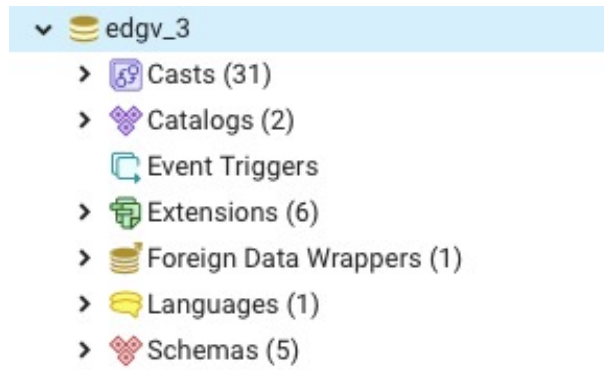


FIG. 3.11: Upload do Banco de Dados no PgAdmin

Este banco de dados conta com 508 tabelas, separadas em 5 *schemas*. Após realizar a conexão com o banco de dados, será possível selecionar este como base de origem ou destino no plugin, de acordo com o mostrado na figura 3.10. O banco de dados está disponível no repositório do GitHub, no link https://github.com/thiago-mpm/pfc_2020_MB.

3.3 ARQUIVO JSON UTILIZADO

Para o desenvolvimento do plugin proposto na presente pesquisa, foi usado um arquivo JSON que auxilia a conversão entre uma base de dados no padrão da EDGV 3.0 para uma em TRD4, que pertence ao programa MGCP.

O arquivo utilizado foi o desenvolvido por (CASTRO, 2019), utilizado na funcionalidade "Conversão de Modelos", disponível no plugin chamado de *Dsg Tools*, previamente citado no presente texto.

```

1  {
2    "__comment": "UTF-8",
3    "schema_ida": "mgcp",
4    "schema_volta": "edgv",
5    "afixo_geom_ida": {
6      "tipo": "prefixo",
7      "POINT": "p",
8      "LINESTRING": "l",
9      "POLYGON": "a"
10   },
11   "afixo_geom_volta": {
12     "tipo": "sufixo",
13     "POINT": "_p",
14     "LINESTRING": "_l",
15     "POLYGON": "_a"
16   },
17   "agregar_geom_ida": true,
18   "agregar_geom_volta": true,
19   "mapeamento_classes": [
20     {
21       "__comment": "AL020 Built-UP Area | Residencial area <-> lml_area_densamente_edificada",
22       "classe_ida": "al020",
23       "classe_volta": "lml_area_densamente_edificada",
24       "filtro_ida": {
25         "$and": [

```

FIG. 3.12: Arquivo JSON utilizado no desenvolvimento do *plugin*

O arquivo completo pode ser encontrado no GitHub usado como repositório para os arquivos gerados por essa pesquisa, através do seguinte link: https://github.com/thiago-mpm/pfc_2020_MB.

O mapeamento é dividido em duas partes: mapeamento de classes e mapeamento de atributos. Sobre o de atributos, este permite mudar o nome deles, de forma independentemente das classes. Realiza também a tradução de valores dos atributos (DINIZ, 2019).

O arquivo JSON foi estruturado de forma que são definidas classes de ida e de volta, para que seja possível realizar a conversão entre as bases. São também definidos os atributos pertencentes a cada uma das classes a serem convertidas, conforme segue na figura abaixo:

```

22     "classe_ida": "a1020",
23     "classe_volta": "lml_area_densamente_edificada",
24     "filtro_ida": {
25         "$and": [
26             {
27                 "$or": [
28                     {
29                         "nome_atributo": "fuc",
30                         "valor": 0
31                     },
32                     {
33                         "nome_atributo": "fuc",
34                         "valor": 4
35                     },
36                     {
37                         "nome_atributo": "fuc",
38                         "valor": 19
39                     }
40                 ]
41             },
42             {
43                 "nome_atributo": "$GEOM_TYPE",
44                 "valor": "POLYGON"
45             }
46         ]
47     },
48     "atributos_default_volta": [
49         {
50             "nome_atributo": "bac",
51             "valor": 0
52         },
53         {
54             "nome_atributo": "fuc",
55             "valor": 0
56         },
57         {
58             "nome_atributo": "fun",
59             "valor": 0
60         },
61         {
62             "nome_atributo": "ord",
63             "valor": 0
64         }

```

FIG. 3.13: Classes e atributos de ida e volta no arquivo JSON

Determinada a classe a qual será convertida a que está sendo analisada, são então traduzidos os atributos destas, e o arquivo em JSON contém o mapeamento entre as nomenclaturas e valores de suas características.

```

281     "classe_ida": "bd120",
282     "classe_volta": "hid_recife",
283     "mapeamento_atributos": [
284         {
285             "attr_ida": "wle",
286             "attr_volta": "situacaoemagua",
287             "traducao": [
288                 {
289                     "valor_ida": 1,
290                     "valor_volta": 0,
291                     "sentido": "ida"
292                 },
293                 {
294                     "valor_ida": 2,
295                     "valor_volta": 4
296                 },
297                 {
298                     "valor_ida": 3,
299                     "valor_volta": 5
300                 },
301                 {
302                     "valor_ida": 4,
303                     "valor_volta": 7
304                 },
305                 {
306                     "valor_ida": 1,
307                     "valor_volta": 0,
308                     "sentido": "ida"
309                 },
310                 {
311                     "valor_ida": 8,
312                     "valor_volta": 0,
313                     "sentido": "ida"
314                 },
315                 {
316                     "valor_ida": 998,
317                     "valor_volta": 0,
318                     "sentido": "ida"
319                 },
320                 {
321                     "valor_ida": 999,
322                     "valor_volta": 0,
323                     "sentido": "ida"
324                 }
325             ]
326         }
327     ]

```

FIG. 3.14: Tradução de atributos no arquivo JSON

Conforme previamente abordado neste texto, os padrões da EDGV 3.0 e TRD4 possuem inúmeras diferenças, e portanto, o mapeamento é extenso e complexo. Sua estrutura completa e detalhada está explicada em (DINIZ, 2019).

3.4 BASE DE DADOS TESTE

Abc.

3.5 CRONOGRAMA PROPOSTO



FIG. 3.15: Cronograma Adotado para a Pesquisa

De acordo com o cronograma acima, foram separadas 08 (oito) semanas para a realização da pesquisa. Abaixo estão descritas as atividades contidas neste:

- Conexão com os bancos de dados da EDGV 3.0 e TRD4 v4.4 no pgAdmin e no Plugin EDGV Converter (semanas 1 e 2): Atividades realizadas e descritas no capítulo 3.2 deste texto
- Realização das adaptações necessárias no código do plugin (semanas 3 e 4): Atividade realizada e descrita no capítulo 3.1.2 deste texto
- Decisão da base de dados teste a ser utilizada nos testes (semanas 5 e 6): Atividade realizada e descrita no capítulo 3.4 deste texto
- Realização dos testes utilizando o plugin e das conclusões acerca dos resultados (semanas 7 e 8): Atividades realizadas e descritas nos capítulos 4 e 5 deste texto

4 RESULTADOS

4.0.1 PLUGIN FINAL

Abc.

4.0.2 RESULTADO OBTIDO COM A BASE DE DADOS TESTE

Abc.

5 CONCLUSÃO

O desenvolvimento do plugin proposto contou com relevante dedicação de tempo para a parte computacional, que se mostrou um desafio para realizar.

Os resultados do arquivo de saída em relação às bases de dados usadas como teste atenderam às expectativas, e de fato, conclui-se que na maioria das bases a serem convertidas, haverá classes da EDGV 3.0 sem correspondência na MGCP TRD4 v4.4. Isso se dá pelo fato de os interesses de mapeamentos dos padrões são diferentes, assim como suas escalas, e então certas representações contidas em um padrão, não estarão necessariamente disponíveis no outro.

5.1 CONTINUIDADE DA PESQUISA

As classes e atributos não mapeados são fornecidos ao usuário no documento de saída do plugin desenvolvido na presente pesquisa. Estas podem ser relacionadas manualmente pelo usuário após o recebimento das relações.

Como continuidade da pesquisa, pode-se propor o desenvolvimento de uma metodologia que ofereça essa relação de forma computacional e automática.

Outra continuidade proposta para a pesquisa, seria o desenvolvimento de um plugin que fornecesse a representação visual em carta da padronização convertida da EDGV 3.0 para o MGCP. Compreende-se que esta seria uma evolução vantajosa para o Exército Brasileiro, em questão de alocação de tempo e de recursos humanos.

6 REFERÊNCIAS BIBLIOGRÁFICAS

GALVÃO, W.P; LIMA, N.C.C (2015). **Avaliação do Impacto da Utilização da ET-EDGV Defesa Fter Na Linha de Produção Cartográfica.** Disponível em: <https://bdex.eb.mil.br/jspui/handle/1/847> [capturado em 24 ago 2020]

COMISSÃO NACIONAL DE CARTOGRAFIA - CONCAR (2010). **Especificação Técnica para a Estruturação de Dados Geoespaciais Vetoriais.** Disponível em: <http://www.geoportal.eb.mil.br/index.php/inde2?id=139> [capturado em 13 jul 2020]

JÓZSEF, C.; OLÍVIA, M. (2009). **Multinational geospatial co-production program.** Disponível em: <https://www.researchgate.net/publication/321700982> [capturado em 13 jul 2020]

DIRETORIA DE SERVIÇOS GEOGRÁFICOS - DSG. **Participação do 1 CGEO em evento Internacional de Defesa.** Disponível em: <http://www.dsg.eb.mil.br/index.php/ultimas-noticias/55-ultimas-noticias-da-dsg/124-participacao-do-1-cgeo-em-evento-internacional-de-defesa> [capturado em 25 ago 2020]

DATE, C.J.; **Introdução a Sistemas de Bancos de Dados.** 8 ed. São Paulo: GEN LTC, 2004.

OLIVEIRA, S.S. (2014). **Bancos de Dados Não-Relacionais: Um Novo Paradigma para Armazenamento de Dados em Sistemas de Ensino Colaborativo.** Disponível em: <https://www2.unifap.br/oliveira/files/2016/02/35-124-1-PB.pdf> [capturado em 25 ago 2020].

QGIS Project, c2020. Sobre o QGIS. Disponível em: <https://www.qgis.org/ptBR/site/about/index.html>. [Capturado em 25 ago 2020].

VITAL, Marcos. SIG em Código Aberto: A História do QGIS. [Entrevista Concedida a] QGIS Project. **QGIS Project**, 2019. Disponível em: <https://www.youtube.com/watch?v=fbIea0o3RWU> [capturado em 25 ago 2020].

CASTRO, B.P.S. (2019). **Mapeamento Entre as Especificações MGCP TRD4 v4.4 e ET-EDGV 3.0** [capturado em 26 ago 2020].

SILVA, D.M.. Python: História e Ascendência. **Revista Programar**, ed. 59, 2018, página 96. Disponível em: <https://www.revista-programar.info/edicoes/edicao-59/> [capturado em 26 ago 2020].

DEVMEDIA (2018). **Python no Mundo Real**. Disponível em: devmedia.com.br/python-no-mundo-real/40244 [capturado em 26 ago 2020].

BORGES, L.E.. **Python para desenvolvedores**. 2ª Edição, Rio de Janeiro. 2010. Disponível em: https://ark4n.files.wordpress.com/2010/01/python_para_desenvolvedores_2ed.pdf [capturado em 26 ago 2020].

COMISSÃO NACIONAL DE CARTOGRAFIA - CONCAR (2017). **Especificação Técnica para a Estruturação de Dados Geoespaciais Vetoriais (ET-EDGV 3.0)**. Disponível em: https://www.concar.gov.br/temp/365@ET-EDGV_versao_3.0_2018_05_20.pdf [capturado em 26 ago 2020].

Multinational Geospatial Co-production Program (2016). **TRD4 v4.4 MGCP Technical Reference Documentation** [capturado em 26 ago 2020].

DIRETORIA DE SERVIÇOS GEOGRÁFICOS - DSG (2014). **Certificado de Registro de Programa de Computador**. Disponível em: http://www.dsg.eb.mil.br/images/certificado_dsg_tools-tarjapreta.pdf [capturado em 27 ago 2020].

PRATES, I. (2015). **DSG Tools**. Disponível em: <https://mundogeo.com/2015/04/15/dsg-tools/> [capturado em 27 ago 2020].

GALVÃO, W.P.; FREITAS, F. L.; LIMA, N. C. C. (2017). **O Impacto do Versi-onamento da Estrutura de Dados Espaciais Vetoriais na Etapa de Reambu-lação, no Âmbito da Diretoria de Serviço Geográfico do Exército**. Disponível em: http://www.cartografia.org.br/cbc/2017/trabalhos/3/fullpaper/CT03-22_1506790445.pdf [capturado em 31 ago 2020].

COSTA, E.R. (2011). **Bancos de Dados Relacionais**. Disponível em: <http://www.fatecsp.br/dti/tcc/tcc0025.pdf> [capturado em 03 set 2020].

SIQUEIRA, F. (2016); **Modelo Entidade e Relacionamentos**. Disponível em: <https://sites.google.com/site/uniplibancodedados1/aulas/aula-4—modelo-entidade-e-relacionamentos> [capturado em 03 set 2020].

BALAGUER, A. (2016); **Modelo Entidade Relacionamento (MER) e Diagrama Entidade-Relacionamento (DER)**. Disponível em: <https://professor.adrianobalaguer.com/2016/10/modelo-entidade-relacionamento-mer-e.html> [capturado em 03 set 2020].

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. (2006). **Sistema de banco de dados**. 5. ed. Rio de Janeiro: Elsevier, Campus, 2006. 781 p. [capturado em 03 set 2020].

FIDELIX, M.C. (2019); **Modelo Entidade e Relacionamento (MER)**. Disponível em: <https://www.slideshare.net/CrisFidelix/3-modelo-entidade-relacionamento> [capturado em 03 set 2020].

MELO, A.C. (2004); **Desenvolvimento aplicações com UML 2.0: do conceitual à implementação**. 2. Ed. Rio de Janeiro: Brasport, 2004 [capturado em 03 set 2020].

DEVMEDIA (2018); **Orientações básicas na elaboração de um diagrama de classes**. Disponível em: <https://www.devmedia.com.br/orientacoes-basicas-na-elaboracao-de-um-diagrama-de-classes/37224> [capturado em 03 set 2020].

FONSECA, G. (2011); **Os principais diagramas da UML**. Disponível em: <https://www.profissionaisti.com.br/os-principais-diagramas-da-uml-resumo-rapido/> [capturado em 03 set 2020].

MACHADO, F.T.S. (2017); **Um Processo para Extração de Esquemas Conceituais em Fontes de Dados JSON Baseado em Técnicas de Similaridade de**

Texto [capturado em 03 set 2020].

INTERNATIONAL, E. (2013); **The JSON Data Interchange Format**. Disponível em: <http://www.ecma-international.org/publications/files/ECMA-ST/ECMA-404.pdf> [capturado em 03 set 2020].

SANTOS, B.A.; ALMEIDA, F.C.; WOLFGRAM, V.A. (2019); **Ferramenta de Conversão de Dados Geoespaciais Vetoriais** [capturado em 10 set 2020].

DINIZ, F.C. (2019); **Relatório Técnico N° 04/2019 – DGEO/1°CGEO, Padrão de Mapeamento Entre Modelagens**.

7 APÊNDICES

8 ANEXOS