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

BRYAN MAIA CORREA
TIAGO PRUDENCIO SILVANO

PLUGIN DE VISUALIZAÇÃO DE DADOS MULTIVARIADOS EM MAPAS COROPLÉTICOS

RIO DE JANEIRO
2020

BRYAN MAIA CORREA
TIAGO PRUDENCIO SILVANO

PLUGIN DE VISUALIZAÇÃO DE DADOS MULTIVARIADOS EM MAPAS COROPLÉTICOS

Projeto Final de Curso apresentado ao Curso de Graduação de Engenharia Cartográfica do Instituto Militar de Engenharia, para a obtenção de grau na Verificação Final (VF) de 2020.

Orientação: Prof. Dr. Ivanildo Barbosa

Rio de Janeiro
2020

c2020

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 autor e dos orientadores.

Silvano, Tiago Prudencio; Correa, Bryan Maia

Plugin de visualização de dados multivariados em mapas coropléticos / Bryan Maia Correa, Tiago Prudencio Silvano. — Rio de Janeiro, 2020.

56 f.

Orientador: Ivanildo Barbosa.

Projeto Final de Curso (graduação) — Instituto Militar de Engenharia, Bacharel em Engenharia Cartográfica, 2020.

1. *Plugin*. 2. Mapa Coroplético. 3. Clusterização. 4. QGIS. 5. *Python*. I. Barbosa, Ivanildo (orient). II. Título

INSTITUTO MILITAR DE ENGENHARIA

BRYAN MAIA CORREA
TIAGO PRUDENCIO SILVANO

PLUGIN DE VISUALIZAÇÃO DE DADOS MULTIVARIADOS EM MAPAS COROPLÉTICOS

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: Ivanildo Barbosa – D.Sc.

Aprovada em 30 de outubro de 2020 pela seguinte Banca Examinadora:

Ivanildo Barbosa – D.Sc. do IME, presidente da banca

Felipe Ferrari – M.Sc. do IME

Philipe Borba – UnB

Rio de Janeiro
2020

Aos familiares, companheiros e professores, fundamentais para nosso crescimento não somente profissional como também pessoal.

AGRADECIMENTOS

Agradeço às pessoas que estiveram presentes em minha vida neste período e me deram apoio e orientação para conseguirmos fazer este trabalho. Em especial agradeço ao nosso orientador, Prof. Dr. Ivanildo Barbosa, pela paciência e bom humor, sempre nos fazendo pensar mais além, e ao Philipe Borba por toda orientação no desenvolvimento do nosso projeto.

Agradeço aos que me apoiaram em todos os momentos de minha formação acadêmica e, principalmente, nesta etapa de conclusão. Agradeço ao meu pai, principal incentivador de ter escolhido este Instituto para iniciar minha carreira acadêmica; minha mãe, por me encorajar em todas as atividades da minha vida; e à minha esposa, Maiara, por me fornecer apoio incondicional ao longo de todo o período em que estive envolvido nas diversas tarefas acadêmicas ao longo de minha formação.

Por fim, agradeço ao amigo e colega de trabalho Tiago Prudencio, sem o qual não seria possível fazer um trabalho de tamanha qualidade como este. Sua amizade sempre foi muito importante ao longo de toda nossa formação e sua dedicação e capacidade técnica foram fonte de inspiração para que eu pudesse chegar ainda mais longe. Sem dúvidas, seria muito mais difícil sem a sua companhia.

Meus sinceros agradecimentos.

BRYAN MAIA CORREA

Agradeço à Deus, inicialmente, por ter me dado essa oportunidade realizar um sonho de vida e ao mesmo tempo me conceder saúde, sabedoria e capacidade para chegar até aqui.

A minha família pelo apoio incondicional, carinho, amor e paciência durante os momentos que estive ausente, nas noites de estudo e nos momentos mais necessários. Em especial, obrigado ao meu filho Henrique e esposa Anna, motivo de todo meu esforço e dedicação para sempre seguir em frente.

Aos professores do Instituto Militar de Engenharia, pelos conhecimentos que foram passados, disposição em ajudar e os conselhos de vida que de uma forma ou de outra contribuíram para minha formação. Ao Prof. Dr, Ivanildo Barbosa por ter idealizado esse projeto e acreditar em nós para desenvolvê-lo, pelas orientações e pelas críticas sempre construtivas. Ao Philipe Borba pelo comprometimento, pelo tempo nas reuniões EAD, seu conhecimento que contribuíram em todas as fases desse projeto.

Por último, mas não menos importante, meu amigo Bryan Maia pela companhia diária, pelas dúvidas tiradas, pelos trabalhos que realizamos juntos e pelos bons momentos de descontração e alegria. Amigo e profissional que eu espero encontrar outras vezes nessa longa carreira que temos pela frente.

A todos que contribuíram de alguma forma para essa realização, obrigado.

TIAGO PRUDENCIO SILVANO

RESUMO

Com a popularização dos softwares livres multiplataforma que integram os sistemas de informação geográfica (SIG), iniciou-se uma intensa demanda voltada para o gerenciamento, manipulação e análise de dados multivariados espaciais. A análise desses dados é uma tarefa complexa que exige métodos adequados para extrair informações e auxiliar na tomada de decisão. Neste projeto, foi implementado um *plugin* para o ambiente QGIS capaz de realizar a clusterização a partir de um conjunto de variáveis de tipo numérico associadas a objetos georreferenciados. O usuário tem a possibilidade de executar a clusterização modificando as variáveis utilizadas, a quantidade de *clusters* e o método de clusterização, obtendo diferentes mapas coropléticos, indicadores de qualidade de *clusters* e regras de decisão. O *plugin* ClusterMap permite a visualização da distribuição espacial dos *clusters* e análise, por meio de regras de árvore de decisão, das características de cada *cluster*. Foi disponibilizada uma funcionalidade com métodos de análise de número ótimo de *clusters: Elbow* (cotovelo) e Silhueta, retornando gráficos que permitem ao usuário a identificação de um número ótimo de *clusters.* O código foi implementado na linguagem *Python* com base no *Processing Framework*, utilizando as bibliotecas *Python*: *Sklearn,* para implementação dos métodos de agrupamento e classificação; *Numpy,* para o processamento de *array* e gerenciamento de matrizes; e *Matplotlib,* com o objetivo de gerar gráficos bidimensionais. São apresentados os requisitos do *plugin,* revisão conceitual que envolve os métodos de clusterização e classificação, etapas de implementação do *plugin,* fases de teste, publicação no repositório do QGIS e aplicações.

Palavras-chaves: *Plugin*; Mapa Coroplético; Clusterização; *QGIS; Python.*

ABSTRACT

Popularization of free multiplatform software integrates the Geographic Information Systems (GIS), an intense demand for management, manipulation and analysis of multivariate spatial data began. The analysis of this data is a complex task that requires adequate methods to extract information and assist in decision making. In this project, a plugin for the QGIS environment was implemented, capable of performing clustering from a set of numerical variables associated with georeferenced objects. The user has the possibility to perform the clustering by modifying the variables used, the number of clusters and the method of clustering, obtaining different choropleth maps, cluster quality indicators and decision rules. The ClusterMap plugin allows visualization of the spatial distribution of the clusters and analysis, through decision tree rules, of the characteristics of each cluster. A functionality with methods for analyzing the optimal number of clusters was made available: Elbow and Silhouette methods, returning graphs that allow the user to identify an optimal number of clusters. The code was implemented in the Python language based on the Processing Framework, using the Python libraries: Sklearn, to implement the grouping and classification methods; Numpy, for array processing and matrix management; and Matplotlib, with the objective of generating two-dimensional graphics. The requirements of the plugin are presented, a conceptual review of clustering and classification methods, stages of implementation of the plugin, testing phases, publication in the QGIS repository and applications.

Keywords: Plugin, Choropleth Map, Clustering, QGIS, Python.

LISTA DE FIGURAS

Figura 1.1 Mapa coroplético de variáveis étnicas clusterizadas dos municípios brasileiros (SILVANO et al., 2020) 12

Figura 3.1 Taxonomia de Técnicas de Clusterização do Projeto 19

Figura 3.2 Exemplo Elbow Method. 20

Figura 3.3 Exemplo Método Silhueta. 21

Figura 3.4 Exemplo de Árvore de Decisão 24

Figura 4.1 Interface K-means customizada. 26

Figura 4.2 Interface K-Means 28

Figura 4.3 Interface Método Hierárquico customizada 28

Figura 4.4 Interface Método Hierárquico. 30

Figura 4.5 Regras da classificação. 33

Figura 4.6 Gerenciador de complementos do QGIS com detalhes do plugin. 36

Figura 4.7 Página oficial do repositório do QGIS. 36

Figura 8.1 Fluxograma do Questionário. 49

Figura 8.2 Representação de UML simplificado para o método k-means. 50

Figura 8.3 Representação de UML simplificado para o Método Hierárquico. 51

Figura 8.4 Resultado da clusterização Método K-Means. 52

Figura 8.5 Resultado da clusterização Método Ward. 53

Figura 8.6 Resultado da Clusterização Método Average Linkage. 54

Figura 8.7 Resultado da Clusterização Método Complete Linkage. 55

Figura 8.8 Resultado da clusterização Método Single Linkage. 56

LISTA DE TABELAS

Tabela 4.1 Cores de cada cluster (categorização) 34

Tabela 5.1 Ordem de importância dos atributos segundo as regras da Árvore de Decisão 37

Tabela 8.1 Regras e Coeficiente de Silhueta de cada cluster obtidos do Método KMeans 52

Tabela 8.2 Regras e Coeficiente de Silhueta de cada cluster obtidos do Método Ward 53

Tabela 8.3 Regras e Coeficiente de Silhueta de cada cluster obtidos do Método Average Linkage 54

Tabela 8.4 Regras e Coeficiente de Silhueta de cada cluster obtidos do Método Complete Linkage ... 55

Tabela 8.5 Regras e Coeficiente de Silhueta de cada cluster obtidos do Método Single Linkage 56

SUMÁRIO

1 INTRODUÇÃO 11
1.1 OBJETIVOS 12
1.2 JUSTIFICATIVA 13
1.3 ORGANIZAÇÃO 13
2 REQUISITOS 15
2.1 REQUISITOS GERAIS DO *PLUGIN* 15
2.2 LIMITAÇÕES 16
3 REVISÃO DE LITERATURA 18
3.1 MÉTODO DE CLUSTERIZAÇÃO 18
3.1.1 ALGORITMO K-MEANS 19
3.1.2 ALGORITMO CLUSTERIZAÇÃO HIERÁRQUICA 21
3.2 MÉTODO DE CLASSIFICAÇÃO 23
4 DESENVOLVIMENTO 25
4.1 ESTRUTURA DO *PLUGIN* 25
4.2 INTERFACE 26
4.2.1 INTERFACE MÉTODO K-MEANS 26
4.2.2 INTERFACE MÉTODO HIERÁRQUICO 28
4.3 IMPLEMENTAÇÃO DOS MÉTODOS DE CLUSTERIZAÇÃO 30
4.4 IMPLEMENTAÇÃO DO MÉTODO COTOVELO E MÉTODO SILHUETA 31
4.5 IMPLEMENTAÇÃO DA CLASSIFICAÇÃO 32
4.6 SIMBOLOGIA E LEGENDA 33
4.7 FASE DE TESTES 34
4.8 PUBLICAÇÃO EM REPOSITÓRIO OFICIAL DO QGIS 35
5 APLICAÇÃO PRÁTICA 37
6 CONCLUSÃO 39
7 REFERÊNCIAS 40
8 ANEXOS 43
8.1 GUIA DE INSTALAÇÃO DO *PLUGIN* CLUSTERMAP PARA QGIS 43
8.2 FLUXOGRAMA DO QUESTIONÁRIO DA FASE DE TESTES 49
8.3 DIAGRAMAS UML 50
8.4 RESULTADOS 52

1 INTRODUÇÃO

De acordo com OLIVEIRA (2008), a clusterização é uma técnica de mineração de dados que oferece uma maneira de entender e extrair informações relevantes de grandes conjuntos de dados. A abordagem em relação a aspectos como a representação dos dados e medida de similaridade entre *clusters*, e a necessidade de ajuste de parâmetros iniciais são as principais diferenças entre os algoritmos de clusterização, influenciando na qualidade da divisão dos cluster*s*. Deste modo, dados podem ser agrupados de acordo com suas características, presentes em uma base de dados e identificados pelas informações que representam cada uma de suas variáveis.

Dentre outras aplicações, o QGIS, software profissional GIS de código aberto, é utilizado para atender as necessidades e potencializar o uso das infraestruturas de geoinformação da Diretoria de Serviço Geográfico. Possibilita a manipulação de base de dados geoespaciais matriciais e vetoriais em um ambiente de banco de dados geográficos e pode executar diversos algoritmos de processamento: sejam ferramentas nativas – os chamados *plugins* principais, mantidos pela equipe de desenvolvimento do QGIS – ou, ainda, *plugins* mantidos por autores individuais e armazenados no Repositório do QGIS ou em repositórios externos.

Diversas áreas de pesquisa como aprendizado de máquina (MITCHELL, 1997), mineração de dados (TAN et al, 2005) e reconhecimento de padrões (BISHOP, 2006) podem ter necessidade de implementação de diferentes métodos de análise de dados. Implementar um *plugin* capaz de realizar diferentes formas de clusterização a partir da escolha, por parte do usuário, das diferentes métricas disponíveis – seja o método de clusterização, a distância ou mesmo o número de *clusters* – torna-se algo relevante neste contexto.

Atualmente, o QGIS possui alguns *plugin*s que contemplam diferentes métodos e métricas para clusterização. Os principais, como *Attribute based clustering*, *ClusterPoints* e *QgisMarkerCluster Plugin* podem limitar a experiência do usuário, reduzindo suas escolhas ao fixar o tipo de método e/ou métricas, além de não apresentarem relatórios parciais durante sua execução.

A implementação de um *plugin* de clusterização, com o desenvolvimento de uma interface que permite ao usuário customizar o processo no ambiente QGIS, produz uma nova ferramenta que permite unir a exibição gráfica da distribuição espacial de *clusters* criados com base em variáveis não-espaciais, a escolha de quantidade de classes e atributos pelo usuário – de modo

a atender a algum tipo de especificação ou para testar melhores combinações – e a possibilidade do usuário interpretar, de forma objetiva, os resultados obtidos.

Esta representação visual pode ser dada através de um mapa temático do tipo coropleta, facilitando uma interpretação dos clusters gerados a partir da execução do algoritmo, como apresentado na Figura 1.1.

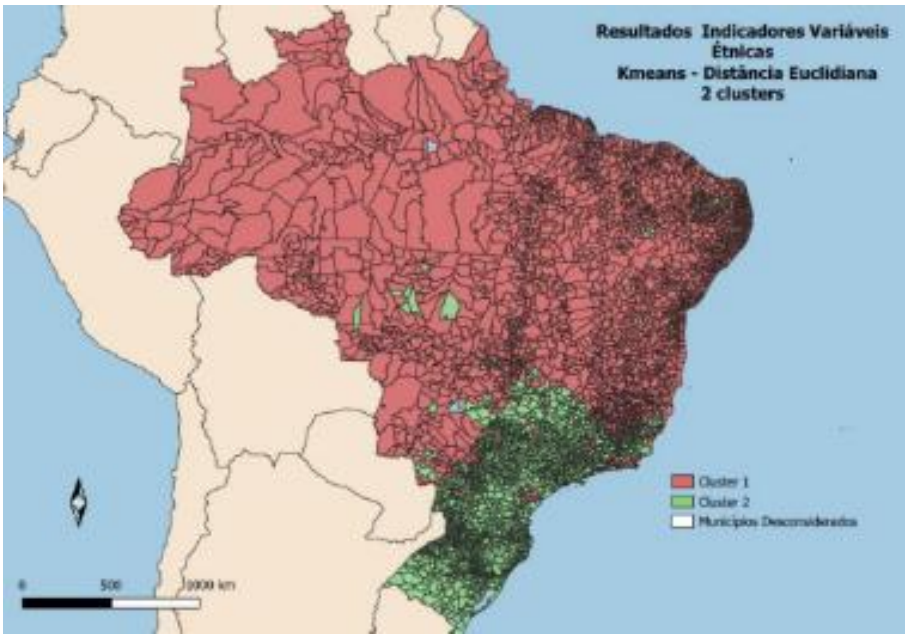


Figura 1.1 Mapa coroplético de variáveis étnicas clusterizadas dos municípios brasileiros (SILVANO et al., 2020)

Estes *clusters* foram gerados a partir de variáveis étnicas consideradas para os municípios do Brasil, sugestão proposta em SILVANO et. al. (2020).

1.1 OBJETIVOS

Com base na técnica apresentada, o objetivo deste Projeto Final de Curso (PFC) é implementar um *plugin*, compatível com o software QGIS, para realizar a clusterização por diferentes opções selecionadas pelo usuário, como método de clusterização, tipo de distância e número de *clusters*.

Além disso, o presente projeto tem como objetivo secundário realizar uma modelagem descritiva dos *clusters* obtidos no processo de clusterização utilizando um modelo de classificação supervisionado para descrever as regras de definição de cada classe (*cluster*).

A proposta do *plugin* não é encontrar o melhor grupamento para um conjunto de objetos, uma vez que os parâmetros de entrada são selecionados pelo usuário, bem como pelos problemas inerentes aos próprios algoritmos de clusterização conforme descrito em ANKERST:

> Existem três razões interconectadas para explicar o motivo pelo qual a efetividade dos algoritmos de clusterização pode ser um problema. Inicialmente, quase todos os algoritmos de clusterização requerem valores para os parâmetros de entrada que demandam dificuldade na determinação, especialmente para conjuntos de dados reais contendo objetos com uma quantidade expressiva de atributos. Além disso, os algoritmos são sensíveis a estes valores de parâmetros, frequentemente produzindo partições muito diferentes do conjunto de dados mesmo para ajustes de parâmetros significativamente pouco diferentes. (1999)

Ainda, os conjuntos de dados multivariados têm uma distribuição muito ampla que não pode ser revelada por um algoritmo de clusterização usando somente um ajuste de parâmetro global.

1.2 JUSTIFICATIVA

A demanda por análise de dados multivariados traz a necessidade do desenvolvimento de um *plugin* que realize clusterização e visualização em uma mesma ferramenta e de forma customizada com relação a métodos e métricas diferenciadas. Linguagens como R, *Python* e *Weka,* embora realizem a tarefa de clusterização e visualização de dados, necessitam que o usuário tenha conhecimento em programação. A popularidade do QGIS pode tornar mais amigável a realização da clusterização e sua representação coroplética.

Alguns das funcionalidades não encontradas, atualmente, em outros *plugins* de clusterização são indicadores de qualidade da clusterização, auxílio para interpretação das classes criadas, escolha da quantidade de classes e atributos pelo usuário, relatórios para análise do método de classificação a partir das regras do modelo de Árvore de Decisão, bem como uma interface de fácil utilização para os usuários no ambiente QGIS.

A ferramenta pode ter aplicações em diversas áreas como, por exemplo, na área de segurança pública ao identificar regiões com maiores índices de ocorrências criminais; na área da saúde ao identificar regiões com maiores incidências de doenças infecciosas; entre outras aplicações que utilizam dados geográficos em feições pontuais de grande concentração, como visto em SILVANO et al (2019) em uma análise de distribuição de indicadores sociais, baseado em mineração de dados e em CARVALHO et. al (2009) ao detectar áreas de prosperidade ou comparar agrupamentos políticos de municípios brasileiros.

1.3 ORGANIZAÇÃO

No segundo capítulo, serão apresentados os requisitos. Será apresentado, também, os requisitos gerais do *plugin*, métodos de agrupamento, *inputs e outputs* e suas limitações.

No terceiro capítulo será apresentada a revisão teórica, abordando conceitos utilizados no desenvolvimento como: métodos de clusterização, detalhando os algoritmos *K-Means* e de

clusterização hierárquica, os conceitos de árvore de decisão e as regras para definição de classes.

O quarto capítulo fará referência ao processo de desenvolvimento do *plugin*, abordando suas fases de elaboração, buscando-se evidenciar os procedimentos adotados em sua implementação para se alcançar os devidos resultados, além de detalhar os parâmetros de entrada e saída de cada processo.

O quinto capítulo apresentará uma aplicação prática de uso para verificar as funcionalidades do *plugin* implementadas. A base de dados utilizada será relacionada aos componentes Renda, Longevidade e Educação para cálculo dos Índices de Desenvolvimento Humano Municipais (IDHM) obtidos do censo 2010 (IBGE, 2010) relativos aos municípios do estado do Rio de Janeiro.

2 REQUISITOS

Esse capítulo apresenta os requisitos gerais do *plugin* desenvolvido nesse projeto. Conforme SOMMERVILLE (2007), requisitos de um sistema são descrições dos serviços fornecidos pelo sistema e as suas restrições operacionais. Os requisitos demonstram as necessidades de um cliente de um sistema que ajuda a resolver um determinado problema.

O levantamento dos requisitos que devem ser alcançados ao final do projeto é a fase inicial no processo de desenvolvimento do *plugin*. Os requisitos foram definidos com base no ambiente de desenvolvimento e a finalidade do *plugin*, buscando maximizar sua funcionalidade e aplicação no que diz respeito ao método de *cluster*ização para criação de mapas coropléticos nos prazos definidos para o projeto.

2.1 REQUISITOS GERAIS DO *PLUGIN*

Como o *plugin* desenvolvido nesse projeto é um complemento ao programa QGIS, um dos principais requisitos é que o *plugin* seja compatível com o próprio QGIS e que permita que qualquer usuário que tenha o QGIS e o *plugin* instalado possa utilizar o mesmo. Sua interface deverá ser coerente com a interface do QGIS e 'amigável' de forma que o usuário não tenha dificuldades de acessar as funcionalidades do *plugin*.

O *plugin* deverá possuir dois Métodos de agrupamento implementados para escolha do usuário, sendo um não hierárquico (*K-Means)* e um hierárquico, que serão executados de acordo com os dados de entrada, a métrica, medidas de dissimilaridade e o número de *clusters* selecionados pelo usuário.

Após a escolha do método de clusterização, o *plugin* deverá possibilitar ao usuário selecionar dentre as camadas no formato vetorial do tipo polígono, linha e ponto, já carregadas na plataforma QGIS, aquela que será utilizada no processo de agrupamento. O *plugin* permitirá o usuário escolher uma, e somente uma, camada de entrada (*input).*

Após o usuário escolher a camada, o *plugin* deverá exibir automaticamente para o usuário somente os seus atributos numéricos – os atributos não-quantitativos estão fora do escopo desta versão do *plugin* –. Os atributos exibidos deverão ser atualizados à medida que o usuário altera a camada de entrada. O *plugin* deverá verificar se o usuário selecionou pelo menos um atributo antes do processamento, caso contrário, o mesmo deve emitir uma mensagem de erro ao executar o *plugin.*

O *plugin* deverá permitir ao usuário escolher o número de cluster*s* para o processamento. Esta escolha do número de cluster*s* não poderá ser irrestrita, devido às limitações da própria metodologia do processo de clusterização. O número de *clusters* deverá ter um valor como *default,* bem como um valor máximo e mínimo para escolha do usuário, sendo sugerido valores entre 2 e 10 para a primeira versão do *plugin.*

Caso o usuário opte pelo Método *K-Means,* uma ferramenta para análise do número ótimo de *clusters* deverá estar disponível, que pode ou não ser utilizada pelo usuário, empregando os Métodos de *Elbow* e silhuetas (THE SCIKIT-VC DEVELOPERS, 2019). Essa ferramenta é sugere ao usuário valores desse parâmetro capazes de fornecer resultados mais promissores, não influenciando na execução do código a nível do processo de agrupamento.

No Método hierárquico, além do número de *clusters*, o *plugin* possibilitará ao usuário escolher o Método de dissimilaridade entre cluster*s* (*Ward, Single Linkage, Complete Linkage e Average Linkage*), bem como a métrica (*Euclidean* e *Manhattan*). O *plugin* permitirá ao usuário escolher apenas uma opção para o número de cluster*s*, dissimilaridade e métrica para cada processamento.

Após o processamento, o usuário terá como *output* um arquivo de formato vetorial, cópia do arquivo selecionado na camada de entrada com um atributo adicionado chamado *'cluster'*, resultado do cluster*ing*, indicando a qual *cluster* cada feição pertence. O *output* poderá ser um arquivo temporário ou salvo em um diretório escolhido pelo usuário.

O arquivo de saída deverá ser uma réplica do arquivo original, acrescida dos resultados da clusterização, categorizado e com legenda. A categorização terá como base o número de *clusters,* sendo atribuído diferentes cores para cada *cluster,* caracterizando assim, um mapa coroplético. A legenda deverá exibir as regras de decisão relacionadas a cada categoria.

Com o objetivo da publicação do *plugin* no repositório do QGIS, o *plugin* deve satisfazer os requisitos de documentação, metadados e licenças exigidos pela política de publicação da plataforma QGIS.

2.2 LIMITAÇÕES

As limitações do *plugin* são as restrições operacionais impostas pelo QGIS, visto que esse é o ambiente de funcionamento da ferramenta desenvolvida.

Um fator importante no processo de agrupamento é o custo computacional, de forma que o hardware utilizado pelo usuário pode se tornar um limitador nessa fase. Desta forma, a quantidade de dados selecionados pelo usuário pode implicar diretamente em algum tipo de erro na execução do *plugin.*

3 REVISÃO DE LITERATURA

3.1 MÉTODO DE CLUSTERIZAÇÃO

De uma maneira mais formal, OCHI et al., (2004), definem o problema de clusterização da seguinte forma: dado um conjunto com n elementos $X = \{X_1, X_2, \dots, X_n\}$ o problema de clusterização consiste na obtenção de um conjunto de k *clusters*, $C = \{C_1, C_2, \dots, C_K\}$. O conjunto C é considerado uma clusterização com k *clusters* caso as seguintes condições sejam satisfeitas:

$$\bigcup_{i=1}^{k} C_i = X$$

$$C_i \neq \emptyset, para\ 1 \leq i \leq k$$

$$\boldsymbol{C_i \cap C_j \neq \emptyset, para\ 1 \leq i, j \leq k\ e\ i \neq j}$$

Em outras palavras, clusterização é a divisão de dados, com base na similaridade entre eles, em grupos disjuntos chamados *clusters*. Isso significa que dados em um mesmo *cluster* são mais similares do que dados pertencentes a *clusters* diferentes (OLIVEIRA, 2008). Cada objeto é semelhante aos demais do grupo, maximizando a homogeneidade dentro do grupo e maximizando a heterogeneidade entre grupos (MORAES, 2016)

A definição de similaridade ou dissimilaridade entre objetos depende do tipo de dado considerado e é geralmente expressa em termo de uma função distância $d(i, j)$, cuja métrica deve ser satisfeita as seguintes condições (HAN, 2001):

a) $d(i, j) \geq 0$: $a\ dist\hat{a}ncia$ é $um\ numero\ n\tilde{a}o\ negativo.$
b) $d(i, i) = 0$: $a\ dist\hat{a}ncia\ de\ um\ objeto\ para\ ele$ é $0.$
c) $d(i, j) = d(j, i)$: $a\ dist\hat{a}ncia$ é $uma\ fun\tilde{c}\tilde{a}o\ sim\acute{e}trica$
d) $d(i, j) \leq d(i, h) + d(h, j)$: *a distância de um objeto i para um objeto j no espaço não é maior do que o caminho entre eles passando por qualquer outro objeto (Desigualdade Triangular)*

Os métodos de clusterização implementados nesse projeto são k-médias (*k-means)* e o hierárquico, como visto na figura 3.1. Estes algoritmos dependem da definição de parâmetros como o número de *clusters*, a métrica de similaridade entre os dados e a medida de dissimilaridade entre cluster*s*.

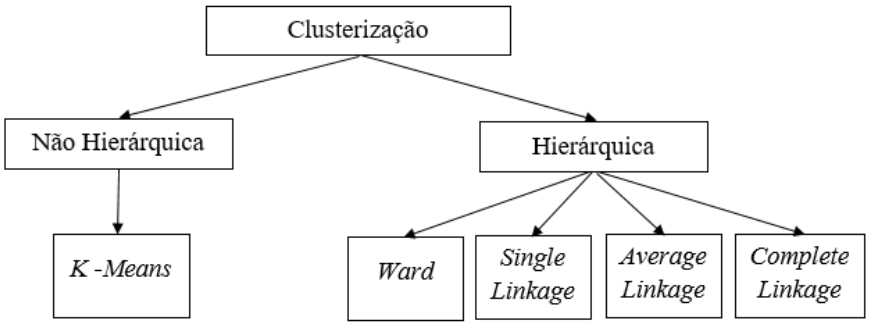


Figura 3.1 Taxonomia de Técnicas de Clusterização do Projeto

3.1.1 ALGORITMO K-MEANS

O algoritmo *k-means* necessita que um número de *cluster* k seja escolhido antecipadamente, a função objetivo a ser minimizada pelo algoritmo é definida pela equação de erro quadrático (CASSIANO, 2015):

$$\sum_{j=1}^{K} \sum_{x \in C_i} \left|\left|x_i - \mu_j\right|\right|^2$$

$$x_i \; elementos \; do \; cluster \; k;$$

$$\mu_j \; centro \; do \; cluster.$$

Os passos do algoritmo são descritos por OLIVEIRA, 2008 (adaptado):

1) Escolha da partição inicial formada por *k cluster*.

A escolha da partição inicial envolve a definição de dois parâmetros: números de *clusters*, e os centros iniciais. Os centros podem ser escolhidos de forma aleatória dentre os elementos dos conjuntos de dados inicial.

2) Atribuindo cada dado ao *cluster* com o qual possui a maior semelhança.

A semelhança entre um dado e um *cluster* é calculada através da distância entre os dois: $||x_i - \mu_j||$, onde $i = \{1, \dots, n\}$ e $j = \{1, \dots, k\}$. O dado é adicionado ao *cluster* com qual tiver a menor distância, em seguida, recalcula-se o centro de cada *cluster*. Ao fim desse passo, os n objetos estão distribuídos entre os k *clusters*.

3) Atualize os centros dos K *clusters*.
4) Volte ao Passo 2 até a convergência.

O critério de término do algoritmo seria quando a função objetivo não pudesse mais ser otimizada. Entretanto, não há garantias que o resultado alcançado seja ótimo.

Uma etapa importante no processo de clusterização, como mencionado anteriormente, é a escolha correta do número de *clusters*, essa escolha não é uma questão simples e exata, necessitando na maioria das vezes a utilização de métodos de análise que tentam encontrar o número ótimo de *clusters*.

Nesse trabalho foram implementados dois métodos de análise do número ótimo de clusters*: Elbow* (cotovelo) e Silhueta. Ambos retornam gráficos que permitem que o usuário identifique um número ótimo de *clusters*.

O Método cotovelo (*Elbow)* permite uma boa aproximação de *k* por meio da variação total intra-*cluster* ou de forma que a soma total quadrada dentro do *cluster* seja minimizada, definido pelo ponto de máxima curvatura para a trajetória da dissimilaridade interna (OLIVEIRA et al., 2017). A determinação do número apropriado de *clusters* é então definido no ponto onde o gráfico apresenta uma curva acentuada, à semelhança de um cotovelo (figura 3.2), ou seja, ponto onde o decaimento reduzir-se-ia bruscamente.

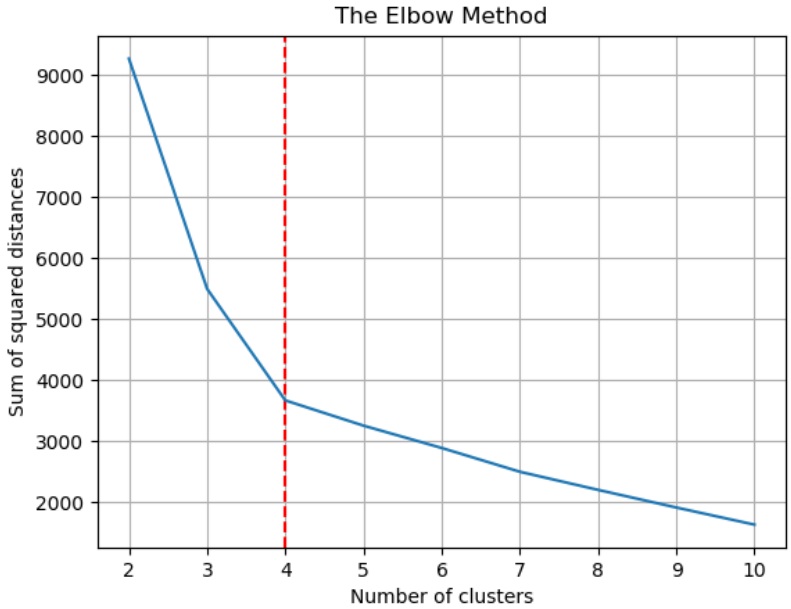


Figura 3.2 Exemplo *Elbow Method.*

No método de Silhueta, de acordo com MACIEL et al. (2015), a silhueta é um gráfico do *cluster* C composto por um valor de silhueta de $s(i)$, $i = 1, \ldots., n$, que reflete a qualidade da alocação dos objetos nos grupos. Cada objeto (indivíduo) do *cluster* é representado por *i* e para cada objeto i o valor $s(i)$, é calculado:

$$s(i) = \frac{b(i) - \alpha(i)}{\max\left(\alpha(i), b(i)\right)}$$

Onde

$\alpha(i)$ é $a\ dissimilaridade\ média\ do\ objeto\ i\ em\ realação\ a\ todos\ os\ objetos\ do\ mesmo\ grupo\ C.$

$b(i)$ é $a\ dissimilaridade\ média\ do\ objeto\ i\ em\ realação\ a\ todos\ os\ objetos\ do\ grupo\ vizinho\ mais\ próximo\ a\ ele, grupo\ X.$

De acordo com MACIEL et al.:

> O valor de $s(i)$ varia no intervalo entre -1 e 1, sendo adimensional. Quando um valor de $s(i) \approx 1$, significa que o objeto i foi bem classificado no grupo C, pois $\alpha(i) < b(i)$. Se o valor de $s(i) \approx -1$, significa que o objeto foi mal classificado, pois $\alpha(i) > b(i)$, ou seja, o objeto i, em média, está mais distante dos objetos do seu próprio grupo, isto é, o objeto do grupo C está mais próximo dos objetos do grupo X. Por sua vez, se $s(i) \approx 0$, o objeto i está entre os grupos C e X, isso ocorre quando $\alpha(i) = b(i)$, indicando que o objeto está num ponto intermediário a dois grupos. Logo, quanto mais próximo a 1, melhor será a qualidade do agrupamento (2015).

Na figura 3.3, é apresentado um exemplo de valores de coeficiente de silhueta para valores entre 2 e 10, podendo ser identificado no gráfico o maior valor de 0,75 para o número $k = 2$ *clusters.*

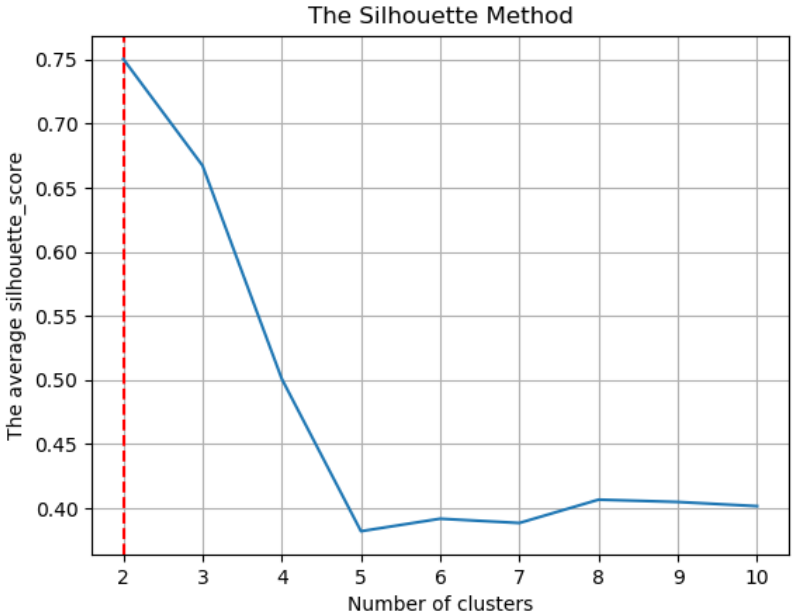


Figura 3.3 Exemplo Método Silhueta.

3.1.2 ALGORITMO CLUSTERIZAÇÃO HIERÁRQUICA

Segundo GAYARRE (2015), a clusterização hierárquica supõe a classificação em partições sequenciais segundo um critério de proximidade entre amostras. O resultado é uma representação em forma de árvore (dendograma). Os métodos hierárquicos são divididos em dois grupos: aglomerativos e divisivos, nesse trabalho será implementado o método aglomerativo.

Algoritmos de métodos aglomerativos (*buttom-up*) iniciam-se com cada objeto ou observação como sendo um respectivo *cluster*. Em cada etapa subsequente, os dois *clusters* mais semelhantes são combinados para criar um *cluster* agregado. O processo é repetido até que todos os objetos sejam finalmente combinados em um único *cluster*. (HAIR JR. et al., 2014)

As distâncias entre cada par de objetos implementadas nesse projeto são:

- Distância de Manhattan,

$$d(x_i, x_j) = \sum_{k=1}^{d} (|x_{ik} - x_{jk}|)$$

- Distância Euclidiana,

$$d(x_i, x_j) = \sqrt{\sum_{k=1}^{d} (|x_{ik} - x_{jk}|)^2}$$

Serão listadas as medidas similaridade ou dissimilaridade implementadas no *plugin* que podem ser selecionadas pelo usuário, bem como uma descrição do comportamento observado em aplicações para clusterização hierárquica tradicional conforme CARVALHO et al. (2009):

- *Ward*

O método parte da soma dos quadrados dos desvios de cada objeto para o centroide dos respectivos cluster*s*, o método consiste em analisar todos os possíveis paras de cluster*s* unidos, detectando qual união produz o menor aumento da soma dos quadrados dos desvios. Tende a unir *clusters* com número pequeno de observações, sendo fortemente enviesado no sentido de produzir *clusters* com mesmo formato e número de observações. É também muito sensível a outliers.

$$D_{K,L} = \frac{d(\overline{x_K}, \overline{x_L})^2}{(\frac{1}{N_k} + \frac{1}{N_L})}$$

$sendo$:

$D_{K,L} \rightarrow medida\ de\ disimilaridade\ entre\ o\ cluster\ L\ e\ o\ Cluster\ K;$

$\overline{x_K}\ e\ \overline{x_L} \rightarrow$ são $vetores\ correspondentes$ às $médias\ dos\ vetores\ de\ caracteristicas\ de\ todos\ elementos\ dentro\ dos\ clusters\ L\ e\ K\ respectivamente;$

$N_L\ e\ N_K \rightarrow número\ de\ lementos\ dentro\ dos\ clusters\ L\ e\ K\ respectivamente.$

- *Single Linkage*

Medida de similaridade entre dois *clusters* é definida pela menor distância de qualquer objeto do *cluster K* para qualquer objeto do *cluster L*.

$$D_{K,L} = \min_{i \in C_K, j \in C_L} d(x_i, x_j)$$

- *Complete Linkage*

Medida de similaridade entre dois *clusters* é definida pela maior distância de qualquer objeto do *cluster K* para qualquer objeto do *cluster L*. É fortemente viesado no sentido de produzir cluster*s* compactos com diâmetros semelhantes, e pode ser severamente distorcido por outliers moderados.

$$D_{K,L} = \min_{i \in C_K, j \in C_L} d(x_i, x_j)$$

- *Average Linkage*

Medida de similaridade entre dois *clusters* é definida pela média das distâncias de todos os objetos do *cluster K* em relação aos objetos do *cluster L*. É um método que tende a juntar cluster*s* com baixa variância, sendo ligeiramente viesado a produzir cluster*s* com igual variância.

$$D_{K,L} = \frac{1}{N_K N_L} \sum_{i \in C_K} \sum_{j \in C_L} d(x_i, x_j)$$

3.2 MÉTODO DE CLASSIFICAÇÃO

Nesse trabalho foi utilizado o método de classificação baseado no modelo de Árvore de Decisão para realizar uma modelagem descritiva dos *clusters* obtidos no processo de clusterização, em outras palavras, foi utilizado um modelo de classificação supervisionado para identificar os principais atributos que distinguem uma classe (*cluster*) de outra. Dessa forma, o usuário poderá verificar, segundo o critério da classificação, por que determinado elemento (vetor) foi atribuído a um determinado *cluster*, bem como a ordem de importância dos atributos, uma vez que o atributo mais importante é colocado no primeiro nó e os atributos subsequentes são alocados em ordem de importância exemplificado na figura 3.4.

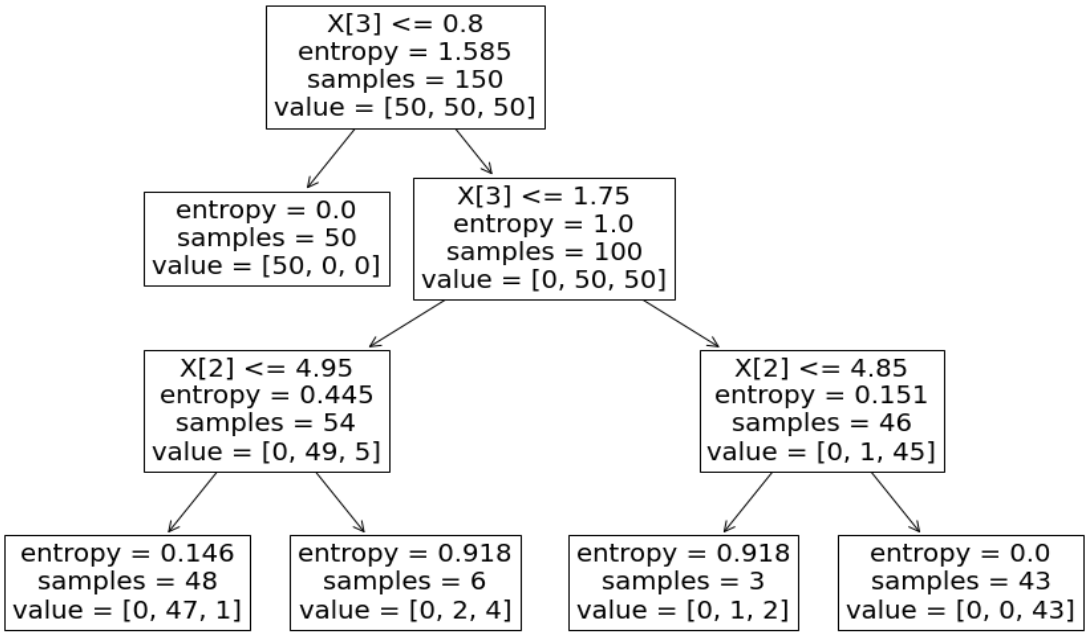


Figura 3.4 Exemplo de Árvore de Decisão

De acordo com DA SILVA et al.:

> O problema de construir uma árvore de decisão pode ser expresso recursivamente: primeiro, deve-se selecionar um atributo para criar um nó-raiz e fazer um ramo para cada possível valor. Isso divide o problema em subconjuntos, um para cada valor do atributo. Depois o processo pode ser repetido recursivamente para cada ramo. Se a qualquer instante todos os exemplos em um nó tiverem a mesma classificação, interrompe-se o desenvolvimento daquela parte da árvore (2008).

Neste trabalho, para o treinamento do modelo de Árvore de Decisão é utilizado como entrada o conjunto de atributos utilizados pelo usuário no método de clusterização e para o conjunto de saída (classes), os *cluster* obtidos no processamento do mesmo. Como critério de seleção de atributos é utilizado o ganho de informação (Gain) que tem como base a medida de entropia.

A entropia caracteriza a impureza dos dados, num conjunto de dados, ela caracteriza a falta de homogeneidade dos dados de entrada em relação a sua classificação. Por exemplo, a entropia é máxima (igual a 1) quando o conjunto de dados é heterogêneo (BARBOSA et al., 2001).

Uma abordagem completa dos conceitos que envolvem Árvore de decisão e critérios de escolha de atributos, além de aplicações do modelo podem ser encontrados em GARCIA 2003, BARBOSA et al. (2001), DA SILVA et al. (2008).

4 DESENVOLVIMENTO

O primeiro passo do *plugin* especificado é a instalação do programa QGIS. Este, pode ser instalado em qualquer sistema operacional como, por exemplo, Linux, OSX, Windows e Android. Neste projeto, será utilizado o sistema operacional Windows 10.

Quanto à atualização de suas versões, o QGIS possui atualizações periódicas, sendo a versão 3.14 a mais recente no momento deste desenvolvimento. Entretanto, para efeito de estabilidade na execução do QGIS, foi utilizada nesse projeto a versão 3.10.8, *long-term-release* (*LTR*), versão estável mais atualizada no momento do desenvolvimento do *plugin*.

Foram instalados, no ambiente QGIS, dois *plugins* para auxiliar no desenvolvimento, sendo eles o *Plugin Builder*, que cria os arquivos necessários para o desenvolvimento de novos *plugins,* e o *Plugin Reloader* que atualiza as alterações realizadas no código do novo *plugin* sem a necessidade de reiniciar o QGIS para carregar as atualizações. Informações adicionais sobre instalação, funcionamento e configuração desses *plugins* podem ser encontrados em Gandhi (2017).

O código do *plugin* foi implementado na linguagem *Python* fazendo uso do *Processing Framework* e com auxílio de um editor de texto. Existem algumas vantagens no uso *Framework,* tais como: diminuição do número de erros; interface padronizada; execução do *plugin* como uma tarefa em lote; e disponibilidade de padrões que facilitam a construção do código, bem como permitem que o *plugin* seja combinado a outras ferramentas do QGIS, criando fluxos de trabalhos automatizados (*Model Builder)*.

4.1 ESTRUTURA DO *PLUGIN*

De acordo com BOOCH (2005), a Linguagem Unificada de Modelagem (UML) é uma linguagem gráfica para visualização, especificação, construção e documentação de artefatos de sistemas complexos de software. Ainda, esta proporciona uma forma-padrão para a preparação de planos de arquitetura de projetos de sistemas, incluindo aspectos conceituais, classes escritas e esquemas de banco de dados.

Para melhor visualização e entendimento da estrutura do *plugin* foram criados dois diagramas de classes simplificados–referentes à clusterização pelo Método Hierárquico e Método *K-Means* (anexo 8.3).

Nas próximas seções serão explicados detalhadamente as heranças, atributos e relações de cada classe apresentada no diagrama UML.

4.2 INTERFACE

A interface do *plugin* foi desenvolvida com base nos requisitos apresentados no capítulo 2, sendo necessária a implementação de duas interfaces, uma para o método *k-means* e outra para o método Hierárquico. O processo de implementação da interface em ambos os casos pode ser dividido em duas partes: a primeira foi desenvolvida com objetivo de criar uma parte da interface customizada utilizando o auxílio do programa *Qt Designer* que possui ferramentas para implementar a interface gráfica; a segunda foi desenvolvida utilizando os métodos da classe *QgsProcessingAlgorithm* do *Processing Framework*, buscando criar uma interface com os mesmos padrões e funcionalidades dos algoritmos de processamento do QGIS.

4.2.1 INTERFACE MÉTODO K-MEANS

Na primeira parte da implementação da interface foi criado o arquivo *kmeansWidget.ui* no ambiente do *Qt Designer* conforme figura 4.1. Foram adicionados à interface os seguintes elementos:

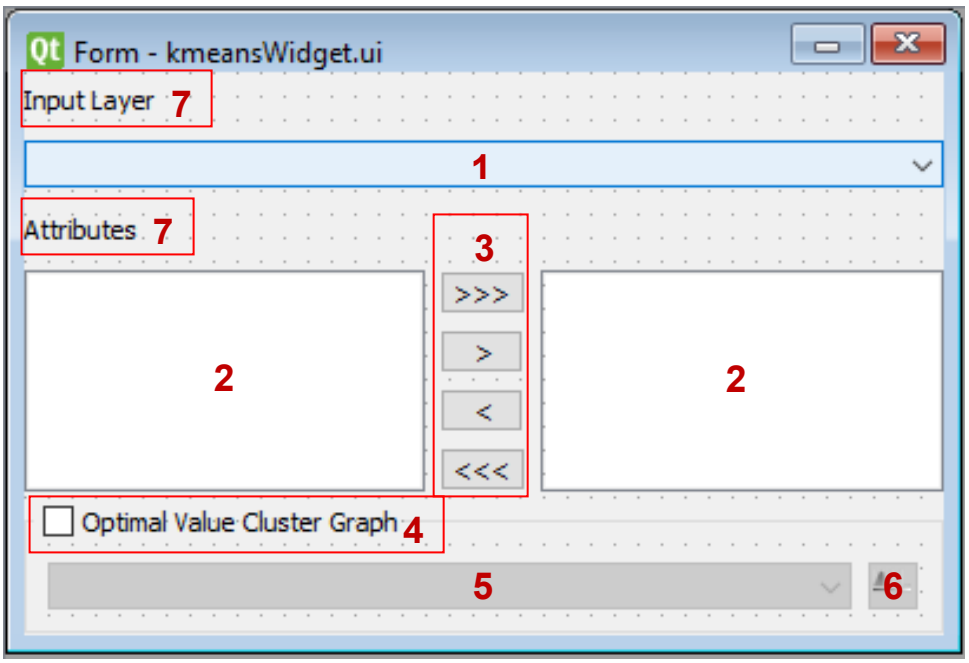


Figura 4.1 Interface *K-means* customizada.

- 1: Uma *Combo box* da categoria *Input Widgets* que receberá as camadas vetoriais carregadas no QGIS para seleção do usuário;
- 2: Duas *List Widget* da categoria *Item Widgets* que exibirá os atributos numéricos da camada selecionada pelo usuário, o atributo passará da *listWidget1* para *listWidget2* caso o usuário selecione o atributo para o processamento;

- 3: Quatro *tool Button* da categoria *Buttons* que permitirá o usuário passar os atributos da *listWidget1* para *listWidget2*;
- 4: Um *Group Box* da categoria *Containers* que poderá ser habilitado caso o usuário deseje gerar a ferramenta gráfica de análise de *clusters*;
- 5: Uma *Combo box* da categoria *Input Widgets* que inserido no *Group Box* com opção de escolha do método de Elbow e Silhueta;
- 6: Um *Tool Button* da categoria *Buttons* que exibirá o gráfico.
- 7: Dois *label* da categoria *Display Widgets* para identificação dos elementos.

Para interagir com cada objeto da interface foi implementado o código *Python kmeansWidget.py* que cria a classe *kmeansWidget* que tem como herança a classe pai *QtWidgets.QWidget*. Todas as funcionalidades dos objetos da interface são criadas através dos métodos implementados na classe *kmeansWidget.*

Para importar a interface customizada para *Processing Framework,* foi implementado o código *kmeansWrapper.py* que cria a classe *kmeansWrapper* que tem como herança a classe pai *WidgetWrapper.* A classe filha possui o método *createPanel que* cria um objeto que é uma instância da classe *kmeansWidget*, método *createWidget* que retorna o objeto instanciado e o método *value* que retorna os parâmetros selecionados pelo usuário na interface customizada. Esses métodos são iniciados no código *kmeasn*cluster*ingalgorithm.py.*

É no arquivo *kmeasn*cluster*ingalgorithm.py* que a interface customizada é mesclada com a interface padronizada gerada pelo *Processing Framework*, finalizando, assim, a interface do método *k-means* conforme figura 4.2.

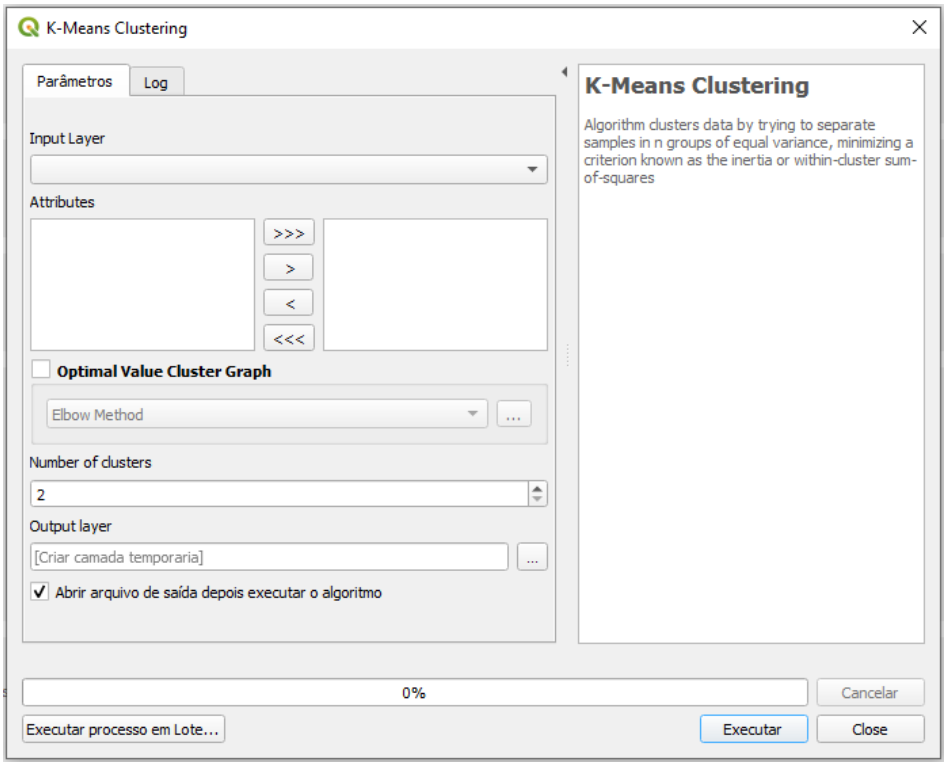


Figura 4.2 Interface *K-Means.*

4.2.2 INTERFACE MÉTODO HIERÁRQUICO

A implementação da interface do Método hierárquico segue a mesma metodologia que o Método *k-means*, porém utiliza elementos diferentes para compor a interface conforme figura 4.3. Foram utilizados os seguintes elementos:

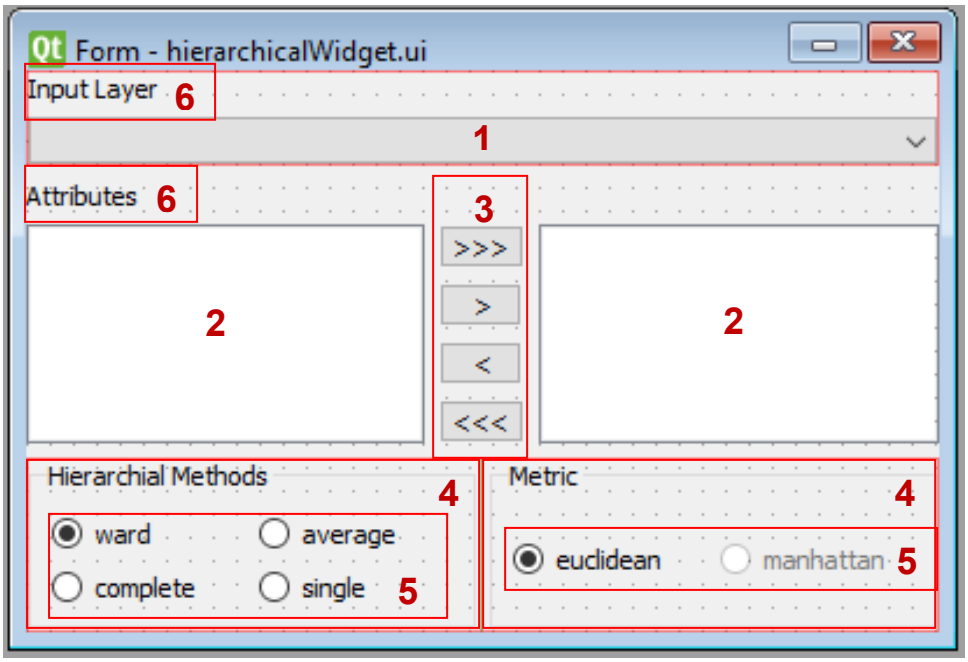


Figura 4.3 Interface Método Hierárquico customizada.

- 1: Uma *Combo box* da categoria *Input Widgets* que receberá as camadas vetoriais carregadas no QGIS para seleção do usuário;
- 2: Duas *List Widget* da categoria *Item Widgets* que exibirá os atributos numéricos da camada selecionada pelo usuário, o atributo passará da *listWidget1* para *listWidget2* caso o usuário selecione o atributo para o processamento;
- 3: Quatro *tool Button* da categoria *Buttons* que permitirá o usuário passar os atributos da *listWidget1* para *listWidget2;*
- 4: Dois *group Box* da categoria *Containers* que exibirão os métodos de dissimilaridade e métrica;
- 5: Seis *Radio Button* da categoria *Buttons* para o usuário selecionar os métodos de dissimilaridade e métrica.
- 6: Dois *label* da categoria *Display Widgets* para identificação dos elementos.

Assim como no método anterior, para importar a interface customizada para *Processing Framework,* foi implementado o código *hierarchicalWrapper.py* que cria a classe *hierarchicalWrapper* com os mesmos métodos da classe *kmeansWrapper*. Esses métodos são iniciados no código *hierarchicalclusteringalgorithm.py.*

É no arquivo *hierarchicalclusteringalgorithm.py* que a interface customizada é combinada com a interface padronizada do *Processing Framework,* finalizando, assim, a interface do método Hierárquico conforme figura 4.4.

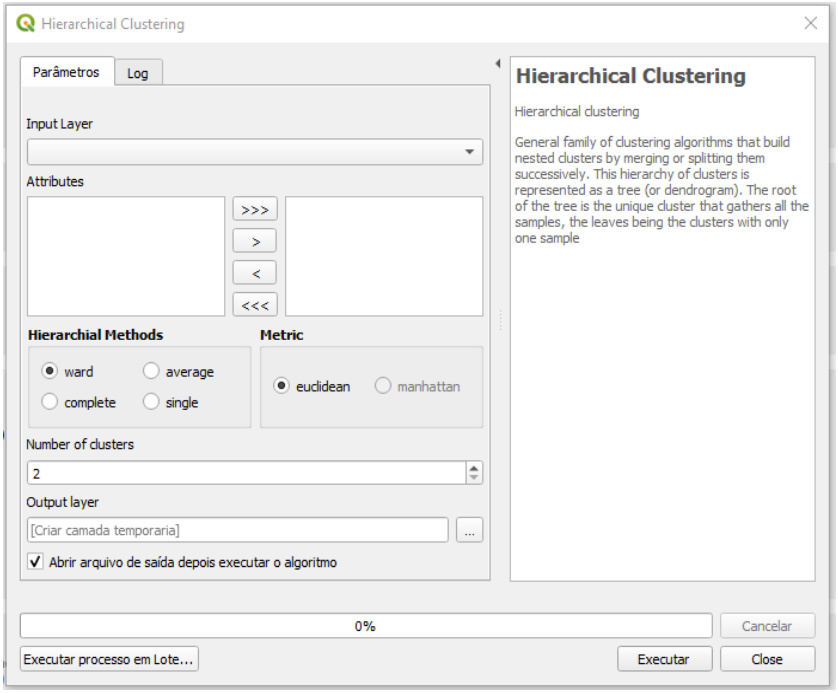


Figura 4.4 Interface Método Hierárquico.

4.3 IMPLEMENTAÇÃO DOS MÉTODOS DE CLUSTERIZAÇÃO

Os Métodos *k-Means* e hierárquico foram implementados nos arquivos *kmeansclusteringalgorithm.py* e *hierarchicalclusteringalgorithm.py*, respectivamente. No caso do Método *kmeans* e Método Hierárquico, os códigos foram adicionados no método *processAlgorithm* da classe *KMeansClusteringAlgorithm* e da classe *HierarchicalClusteringAlgorithm,* respectivamente. As duas classes têm como herança a classe pai *QgsProcessingAlgorithm* seguindo o modelo do *Processing Provider.*

Para implementar ambos os Métodos de clusterização foi utilizada a biblioteca de código aberto *Scikit-Learn*, que possui diferentes algoritmos implementados para a aplicação de modelos de mineração de dados na linguagem *Python.* A biblioteca foi adotada nesse projeto pois proporciona uma simplicidade na aplicação dos modelos e possui uma documentação completa disponível na página oficial (*USER GUIDE*, 2011).

O Método *k-means* utiliza a classe *KMeans* da biblioteca *sklearn*. A classe possui parâmetros como *default* e outros que devem ser passados. Nesse método, a classe recebe como parâmetros:

- O número de cluster*s* escolhido pelo usuário;
- O *randon_state* igual a 0 (zero), definido no projeto para tornar a aleatoriedade da escolha dos centroides de inicialização determinística.

O método *KMeans.fit()* é responsável realizar o treinamento do Método e determinar o *cluster* de cada objeto, ele recebe como parâmetro um *array* com os atributos das feições selecionados pelo usuário (dados), enquanto o atributo *labels_* retorna um *array* com o número do *cluster* de cada feição que é adicionado a tabela de atributos da camada de saída.

O Método Hierárquico utiliza a classe *Agglomerative*Cluster*ing* da biblioteca *sklearn*, a classe possui parâmetros como *default* e outros que devem ser passados. Nesse método a classe recebe como parâmetros:

- O número de cluster*s*;
- *Affinity*, que defini a métrica utilizada (Euclidiana ou *Manhattan*); e
- *linkage*, que define a medida de dissimilaridade entre cluster*s* (*Ward, Single Linkage, Complete Linkage e Average Linkage*).

Todos esses parâmetros são definidos pelo usuário.

Assim como a classe *KMeans,* a classe *Agglomerative*Cluster*ing* possui o método *fit()* e o atributo *label_*. Os resultados são adicionados a tabela de atributo da camada de saída.

Mais informações sobre as classes apresentadas, bem como os parâmetros podem ser pesquisados na documentação oficial (USER GUIDE, 2011)

4.4 IMPLEMENTAÇÃO DO MÉTODO COTOVELO E MÉTODO SILHUETA

A implementação do código foi realizada no arquivo *graph.py,* que cria a classe *createGraph* e utiliza como principais bibliotecas a *matplotlib* (HUNTER, 2007) e *sklearn* (PEDREGOSA, 2011). O construtor da classe recebe como parâmetros os atributos das feições (dados) e o método selecionado pelo usuário, passados para os métodos da classe *elbowMethod*() e *silhouetteMethod*(), que constroem os respectivos gráficos.

O Método *Elbow* cria uma instância da classe *KMeans* e realiza clusterização com o método *fit()* para cada número de *cluster* entre 2 a 10, para cada treinamento é calculada a soma das distâncias quadradas das amostras até o centro do *cluster* mais próximo utilizando um atributo *inerttia_* do objeto instanciado.

O Método Silhueta possui uma implementação similar ao Método *Elbow,* porém não realiza o cálculo da soma das distâncias quadradas das amostras até o centro do *cluster* mais próximo. Ao invés disso, calcula o valor médio de silhueta utilizando os métodos da classe *silhouette_score* da biblioteca *sklearn*.

É importante destacar que o gráfico de análise do número ótimo de *cluster* nesse projeto só pode ser realizado no método *k-means* de clusterização não sendo disponível para o método hierárquico.

4.5 IMPLEMENTAÇÃO DA CLASSIFICAÇÃO

A implementação do Método de classificação foi realizada no arquivo *classification.py* que cria a classe *classification* e utiliza como principais bibliotecas *sklearn, collections* e *statistics.* O construtor da classe recebe como parâmetros:

- Um *array* com os dados utilizados no treinamento do modelo de clusterização;
- Nome dos atributos selecionado pelo usuário;
- O resultado da clusterização, número do *cluster* de cada feição.

O método *decisionTree* instância um objeto da classe *DecisionTreeClassifier* com parâmetro *criterion = 'entropy',* esse mesmo objeto chama o método *fit()* que realiza o treinamento do método de classificação. O resultado da classificação é passado para os métodos *find_path* e *get_rules* que descrevem o caminho e o critério de classificação de cada classe (*cluster).*

A classe *classification* é importada nos arquivos *KMeans*Cluster*ingAlgorithm.py* e *Hierarchical*Cluster*ingAlgorithm.py,* a classe passa os critérios de cada *cluster* determinados no modelo de classificação que são descritos na saída do processamento do *plugin* para o usuário conforme figura 4.6.

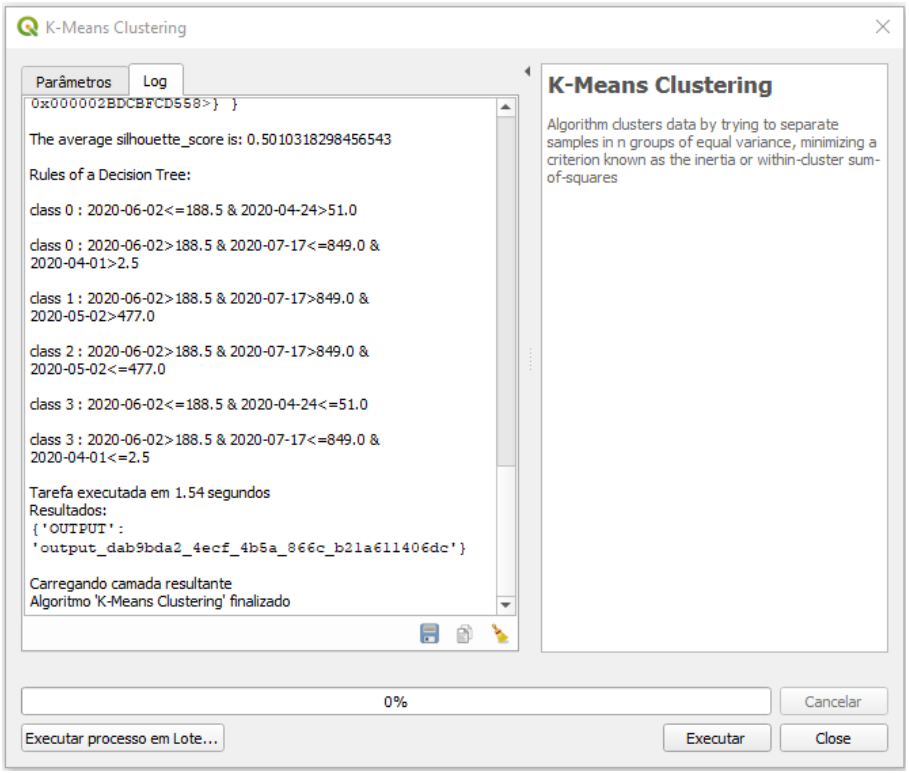


Figura 4.5 Regras da classificação.

4.6 SIMBOLOGIA E LEGENDA

Com objetivo de criar o mapa coroplético com uma legenda, as feições criadas foram categorizadas com base no atributo *cluster* obtido com o resultado da clusterização.

Para implementar a categorização criou-se uma instância da classe *QgsRendererCategory()*, que recebe como parâmetros os rótulos do atributo *cluster*, a legenda obtida dos critérios gerados pelo Método de Classificação e a cores a serem atribuídas a cada *cluster* conforme tabela 4.1. A instância é passada a classe *QgsCategorizedSymbolRenderer()* que, por sua vez, cria uma instância que categoriza à camada de saída através do método *setRenderer()*.

Tabela 4.1 Cores de cada *cluster* (categorização)

Cluster	***(Red, Green, Blue, Transparency)***	**Hexadecimal**	**Cor**
1	(43, 131, 186, 255)	#2b83ba	
2	(241, 96, 93, 255)	#f1605d	
3	(157, 211, 167, 255)	#9dd3a7	
4	(237, 248, 185, 255)	#edf8b9	
5	(232, 221, 58, 255)	#e8dd3a	
6	(249, 158, 89, 255)	#f99e59	
7	(108, 206, 89, 255)	#6cce59	
8	(137, 107, 178, 255)	#896bb2	
9	(205, 63, 113, 255)	#cd3f71	
10	(215, 25, 28, 255)	#d7191c	

4.7 FASE DE TESTES

Como forma de validar as implementações realizadas no projeto, a fase de testes foi constituída a partir da confecção de um Guia de Instalação do clusterMap com vídeo de demonstração de uma visão geral do uso do *plugin*[1] e de um questionário *online* sobre a experiência de uso e instalação do clusterMap, presentes nos anexos.

Desta forma, foram enviados e-mails para integrantes dos Centros de Geoinformação, alunos de graduação do 3°ano do curso de Engenharia Cartográfica do Instituto Militar de Engenharia (IME), alunos de doutorado da PPGG-UFRJ e funcionários do IPP – Instituto Pereira Passos contendo uma breve explicação do objetivo do *plugin* e de sua fase de testes, além do anexo contendo um conjunto de arquivos, sendo: um arquivo compactado com a versão mais atualizada do *plugin*; o Guia de Instalação do ClusterMap; uma arquivo compactado com

[1] https://www.youtube.com/watch?v=XU8C9e9WNd8&feature=youtu.be

um *shapefile* para teste de uso; e um questionário sobre a experiência de uso e instalação do *plugin.*

Esta etapa visou coletar oportunidades de melhoria, bem como verificar a eficiência do Guia de Instalação, do vídeo de demonstração de uso e da interpretação dos resultados obtidos por parte do usuário. Foram recebidas respostas ao questionário e alguns dos resultados coletados na etapa de fase de testes indicam oportunidades de melhoria no detalhamento da descrição das etapas e na implementação de métodos de avaliação como o Akaike information criterion (AIC), o que pode ser implementado em versões futuras do *plugin.* Durante o período de desenvolvimento, não foram reportados *bugs* pelos usuários.

4.8 PUBLICAÇÃO EM REPOSITÓRIO OFICIAL DO QGIS

Para publicação do *plugin* em repositório oficial, foram atendidos os requisitos exigidos, conforme orienta o repositório do QGIS:

- *Plugins* precisam ter pelo menos documentação mínima;
- Os metadados do *plugin* devem conter um link para o código-fonte, um rastreador de problemas e uma licença;
- A licença do *plugin* deve ser compatível com GPLv2 ou posterior;
- Respeitar licenças por bibliotecas e outros recursos que seu *plugin* usa;
- Se o *plugin* tiver uma dependência externa, isso precisa ser claramente declarado no campo em metadados: você pode incluir um pequeno guia para instalar bibliotecas Python conforme necessário.

Assim, o código foi implementado e disponibilizado em um repositório público da plataforma *Github*[1], contendo documentação própria disponibilizada em formato PDF, Guia de instalação do *plugin* e de suas bibliotecas, vídeo para uma visão geral de uso do *plugin,* gerenciador de erros (*bugtracker)* e licença GPLv2, atendendo todos os requisitos exigidos para publicação do *plugin* em repositório oficial.

Desta forma, foi realizada a solicitação de publicação e, uma vez aceita, o *plugin* foi publicado e pode ser utilizado a partir do *download* direto do repositório do QGIS, conforme Figura 4.6 ou na página oficial do repositório do QGIS, conforme Figura 4.7.

[1] https://github.com/tiagoPrudencio/ClusterMap

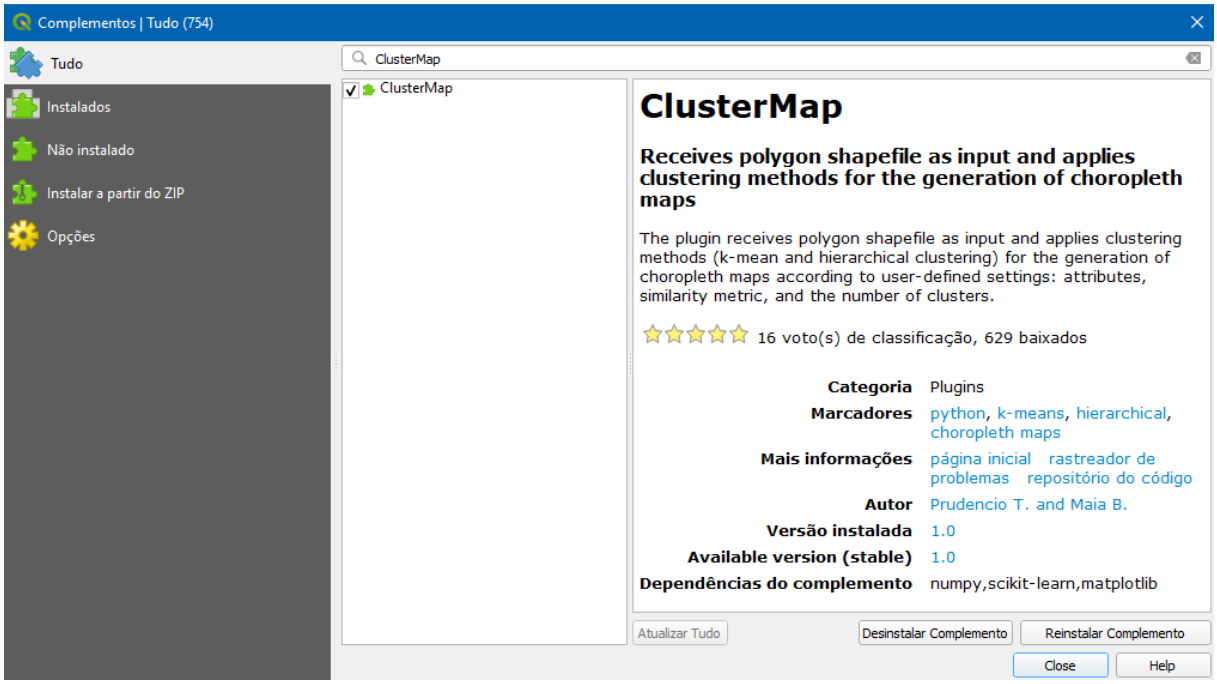


Figura 4.6 Gerenciador de complementos do QGIS com detalhes do plugin.

Figura 4.7 Página oficial do repositório do QGIS.

5 APLICAÇÃO PRÁTICA

Esta seção apresenta uma aplicação para demonstrar as funcionalidades do *plugin* implementadas. A base de dados utilizada refere-se a informações de IDHM Renda (*Renda*), IDHM Longevidade (*Longev*) e IDHM Educação (*Educação)* extraídos do Sistema IBGE de Recuperação Automática – SIDRA (SIDRA, 2019) com base na malha de municípios mais recente do Rio de Janeiro – Brasil (IBGE, 2010). A escolha desta base busca, tão somente, apresentar e comparar de maneira didática os mapas coropléticos e resultados gerados pelos Métodos de clusterização, não fazendo parte do escopo do projeto a análise desses resultados. Caso interesse ao leitor, consulte IPEA (2013) para esclarecer eventuais dúvidas relativas ao índice.

Nesta aplicação foi utilizado o Método não-hierárquico *K-means,* e os Métodos hierárquicos aglomerativos *Ward, Average Linkage, Complete Linkage e Single Linkage*, com número de *cluster* obtido com a ferramenta de análise do número ótimo de cluster*s* disponível no *plugin* segundo o Método do *Elbow.* A distância Euclidiana foi utilizada para os Métodos *K-Means, Ward e Average linkage,* enquanto a distância *Manhattan* foi utilizada para os Métodos *Complete Linkage* e *Single Linkage*.

Foram apresentados nos anexos 8.4 os mapas coropléticos resultantes do processo de clusterização, bem como o Coeficiente de Silhueta e regras de decisão referentes a cada *cluster*.

Observando-se os mapas, nota-se que, em relação à formação dos cluster*s*, os Métodos *K-Means, Ward* e *Complete Linkage* obtiveram resultados semelhantes, enquanto os Métodos *Average Linkage* e *Single Linkage* obtiveram resultados bastante distintos. Esta conclusão é corroborada pelas regras da Árvore de Decisão que definiram a ordem de importância dos atributos para classificação dos objetos em seus respectivos cluster*s,* conforme Tabela 5.1.

Tabela 5.1 Ordem de importância dos atributos segundo as regras da Árvore de Decisão

	Renda	Longevidade	Educação
K-Means	3	1	2
Ward	3	1	2
Average Linkage	1	2	3
Complete Linkage	3	1	2
Single Linkage	1	-	2

Em todos os Métodos os cluster*s* apresentaram um Coeficiente de Silhueta abaixo de 0,6, indicando que os objetos estão num ponto intermediário a dois grupos. É importante notar que alguns cluster*s* obtiveram valores de Coeficiente de Silhueta iguais a zero (0.0), um caso particular do resultado da clusterização, onde o *cluster* possui apenas um elemento.

Os resultados indicam que, para o Método *K-Means,* foi obtida uma maior homogeneidade quanto ao número de munícipios em cada *cluster.* Enquanto, para o Método hierárquico, foram verificados *clusters* com apenas um elemento, como, por exemplo, os municípios de Niterói e Rio de Janeiro. Evidenciando assim, que os valores de atributos destes municípios se diferenciam quando comparados aos demais municípios do Estado.

6 CONCLUSÃO

Este projeto teve como objetivo a implementação e disponibilização de um novo *plugin* para a plataforma QGIS com intuito de realizar um agrupamento de dados multivariados espaciais e a visualização dos cluster*s* formados por meio de um mapa coroplético.

A primeira etapa de desenvolvimento deste projeto foi o levantamento dos requisitos, relacionados à interface, os dados de entrada e a saída, os métodos de clusterização, medidas de dissimilaridade, os parâmetros para execução de cada Método, e a ferramenta de análise do número ótimo de cluster*s*. Ainda, foram especificados os parâmetros do *plugin*, formato dos arquivos de entrada e saída, tipo de atributo dos dados, limitações de número de *cluster*.

Um dos requisitos finais do projeto foi a publicação do *plugin* no repositório oficial do QGIS. Para isso, foram atendidos todos os requisitos exigidos conforme orienta a seção de publicação no site oficial do QGIS. Esta publicação foi realizada em 24 de setembro de 2020 e pode ser verificada na página oficial do repositório do QGIS, contendo informações adicionais como número de *downloads* e avaliação dos usuários.

Finalmente, o projeto foi completamente desenvolvido atendendo todos os requisitos estabelecidos nesse projeto com o objetivo de desenvolver um novo *plugin* de clusterização na plataforma QGIS, e ao final, ter a publicação do *plugin* no repositório oficial da plataforma. Além disso, há possibilidade de melhoria para trabalhos futuros com implementação de outras funcionalidades como análise de variância e a inclusão de atributos não-numéricos (categóricos) na clusterização.

7 REFERÊNCIAS

ANKERST, M.; BREUNING, M. M.; KRIEGEL, H.; SANFER, J (1999). **OPTICS: Ordering Points To Identify The** Cluster**ing Structure.** Institute for Computer Science, University of Munich. Disponível: https://www.researchgate.net/publication/221214752_OPTICS_Ordering_Points_to_Identify_the_Clustering_Structure. [Acesso em 17 mai 2020].

BARBOSA, J. M; CARNEIRO, T. G. S.; TAVARES, A. I.; (2001). **Método de Classificação por Árvore de Decisão.** Programa de Pós-Graduação em Ciência da Computação. Universidade Federal de Ouro Preto, MG, Brasil. Disponível em: http://www.decom.ufop.br/menotti/paa111/files/PCC104-111-ars-11.1-JulianaMoreiraBarbosa.pdf. [Acesso em 10 ago 2020].

BISHOP, C.M. **Pattern Recognition and Machine Learning (information science and statistics).** Springer. 2006. Disponível em : http://users.isr.ist.utl.pt/~wurmd/Livros/school/Bishop%20-%20Pattern%20Recognition%20And%20Machine%20Learning%20-%20Springer%20%202006.pdf [Acesso em 19 mai 2020].

BOOCH, G., RUMBAUGH, J., JACOBSON, I. **UML: guia do usuário.** Elsevier, 2005. Disponível em: https://books.google.com.br/books?hl=pt-BR&lr=&id=ddWqxcDKGF8C&oi=fnd&pg=PR13&dq=UML&ots=ffvIkc9LQN&sig=5ZJvW-N8XObbDFL9a3HkRicY8xA#v=onepage&q=UML&f=false [Acesso em 07 out 2020].

CARVALHO, A. X. Y., Albuquerque, P. H. M., Almeida Junior, G. R., Guimarães, R. D. (2009). Cluster**ização Hierárquica Espacial.** Instituto de Pesquisa Econômica Aplicada. Brasília, 2009. Disponível : http://www.ipea.gov.br/portal/index.php?option=com_content&view=article&id=4737 [Acesso em 13 mai 2020].

CASSIANO, K.M. (2015).**Análise de Séries Temporais Usando Análise Espectral Singular (SSA) e** Cluster**ização de Suas Componentes Baseada em Densidade.** Tese (Doutorado) – Engenharia Elétrica, Pontifícia Universidade Católica, RJ, Brasil. Disponível: https://www.maxwell.vrac.puc-rio.br/24787/24787_5.PDF [Acesso em 16 mai 2020].

DA SILVA, W. V.; DEL CORSO, J. M.; WELGACS, H. T.; PEIXE, J. B.; (2008). **Avaliação da Escolha de um Fornecedor Sob condição de Risco a Partir do Método de Árvore de Decisão.** Revista de Gestão USP, São Paulo, v.15, n.3,p.77-94. Disponível em: https://www.academia.edu/12953191/Avalia%C3%A7%C3%A3o_da_escolha_de_um_fornecedor_sob_condi%C3%A7%C3%A3o_de_riscos_a_partir_do_m%C3%A9todo_de_%C3%81rvore_de_Decis%C3%A3o. [Acesso em 10 ago 2020].

GARCIA, S. C., (2003). **O Uso de Árvore de Decisão na Descoberta de Conhecimento na Área da Saúde.** Dissertação (Mestrado) - Ciências da Computação, Universidade Federal do Rio Grande do Sul, Brasil. Disponível em: https://lume.ufrgs.br/bitstream/handle/10183/4703/000503532.pdf?sequence=1&isAllowed=y. [Acesso em 10 ago 2020].

GANDHI, Ujaval. ***QGIS tutorial and Tips.*** 2017. Disponível em: https://www.qgistutorials.com/en/docs/building_a_python_*plugin*.html. [Acesso em: 07 abr 2020].

GAYARRE, L. P.,(2015). **Um Algoritmo de** Cluster**ização de Dados para Auxílio e Análise de Comportamentos dos Sistemas.** Tese (Doutorado) – Engenharia e Tecnologia Espacial/Mecânica Espacial e Controle. Instituto Nacional de Pesquisa Espacial, São José dos Campos -SP, Brasil. Disponível**:** http://mtc-m21b.sid.inpe.br/col/sid.inpe.br/mtc-m21b/2015/04.23.19.03/doc/publicacao.pdf?metadatarepository=&mirror=iconet.com.br/banon/2006/11.26.21.31. [Acesso 20 mai 2020].

HAIR JR, J. F.; BLACK, W. C.; BABIN, B. J.; ANDERSON, R. E., (2014); ***Multivariate Data Analysis.*** Pearson Education Limited, 2014. Disponível: https://is.muni.cz/el/1423/podzim2017/PSY028/um/_Hair_-_Multivariate_data_analysis_7th_revised.pdf. [Acesso em 19 mai 2020]

HAN, JIAWEI, KAMBER, MICHELINE (2001). **Data mining: concepts and techniques.** São Francisco: Morgan Kaufmann, 2001. Disponível: http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/The-Morgan-Kaufmann-Series-in-Data-Management-Systems-Jiawei-Han-Micheline-Kamber-Jian-Pei-Data-Mining.-Concepts-and-Techniques-3rd-Edition-Morgan-Kaufmann-2011.pdf [Acesso em 13 mai 2020].

HUNTER, J.D., 2007. **Matplotlib: A 2D graphics environment. Computing in science & engineering**, 9(3), pp.90–95.

IBGE, **Instituto Brasileiro de Geografia e Estatística. Malhas Municipais – Brasil,** 2010. Disponível em: ftp://geoftp.ibge.gov.br/organizacao_do_territorio/malhas_territoriais/malhas_municipais/municipio_2015/Brasil/BR/br_municipios.zip [Acesso em 08 set 2019]

IPEA, **Índice de Desenvolvimento Humano Municipal Brasileiro**, 2013. Disponível: https://www.ipea.gov.br/portal/images/stories/PDFs/130729_AtlasPNUD_2013.pdf [Acesso em 21 out 2020]

MACIEL. A. M.; VINHAS, L.; CÂMARA, G., (2015). **Algoritmos de** Cluster***ing*** **para Separação de culturas Ágrícolas e tipos de Uso e Cobertura da Terra Utilizando de Sensoriamento Remoto.** Simpósio Brasileiro de Sensoriamento Remoto – SBSR, João Pessoa – PB, Brasil. Instituto Nacional de Pesquisa Espaciais – INPE. Disponível em: http://www.dsr.inpe.br/sbsr2015/files/p0903.pdf . [Acesso em 10 ago 2020].

MITCHELL, T.M., **Machine learning**. McGraw-Hill Science/Engineering/Math, 1997. Disponível em: http://profsite.um.ac.ir/~monsefi/machine-learning/pdf/Machine-Learning-Tom-Mitchell.pdf. [Acesso em 20 mai 2020].

MORAES, M. B. da Costa. **Análise Multivariada Aplicada à Contabilidade.** Universidade de São Paulo, 2016. Disponível em: https://edisciplinas.usp.br/pluginfile.php/2232110/mod_resource/content/1/An%C3%A1liseMultivariada-Aula12.pdf. [Acesso em 06 set 2020].

OCHI, L. S., Dias C. R., Soares S. S. F. (2004). Cluster**ização em Mineração de Dados**. Instituto de Computação - Universidade Federal Fluminense (IC - UFF), Niterói, Rio de Janeiro, Brasil. Disponível : http://www2.ic.uff.br/~satoru/conteudo/artigos/ERI-Minicurso-SATORU.pdf [Acesso em 15 mai 2020].

OLIVEIRA, S. R. de M.; ABREU, U. G. P.; FASIABEN, M. C. R.; BARIONI, L. G.; DE LIMA, H. P.; ALMEIDA, M. M. T. B.; DE OLIVEIRA; O. C. (2017). **Identificação de padrões tecnológicos do sistema de pecuária de corte desenvolvido no Cerrado.** SBIAgro. Disponível em: https://www.alice.cnptia.embrapa.br/bitstream/doc/1076719/1/SBIAGRO2017Oliveira.pdf. Acesso em: 3 jun 2020.

OLIVEIRA, Tatyana B. Soares, (2008). **Clusterização de dados utilizando técnicas de redes complexas e computação bioinspirada.** Dissertação (Mestrado) - Ciências de Computação e Matemática Computacional, Universidade de São Paulo, SP, Brasil. Disponível: https://teses.usp.br/teses/disponiveis/55/55134/tde-01042008-142253/publico/dissertacao.pdf [Acesso em 14 mai 2020].

PEDREGOSA et al., JMLR. **Scikit-learn: Machine Learning in Python**, 12, pp. 2825-2830, 2011.

USER GUIDE. **Scikit – Learn, 2011.** *User Guide.* Disponível em: https://scikit-learn.org/stable/user_guide.html.[Acesso em: 20 jun 2020].

SIDRA, **Sistema IBGE de Recuperação Automática**. Disponível em: <https://sidra.ibge.gov.br/home/pnadct/brasil> [Acesso em 08 set 2019]

SILVANO, T.P.; CORREA, B.M.; BARBOSA, I. (2020). **Análise da distribuição espacial de indicadores sociais e demográficos: uma abordagem baseada em mineração de dados**. Revista Brasileira de Cartografia, vol 72, n.1, p.67-80, 2020. Disponível em: https://doi.org/10.14393/rbcv72n1-50970. [Acesso em 05 abr 2020].

SOMMERVILLE, Ian. **Engenharia de Software**. Tradução: Selma Shin Shimizu Melnikoff, Reginaldo Arakaki, Edilson de Andrade Barbosa. 8. ed. São Paulo: Person Addison-Wesley, 2007.

TAN P.N., STEINBACH M., KUMAR V., **Introduction to Data Mining.** 1ed. Addison Wesley, 2005. Disponível em: https://www-users.cs.umn.edu/~kumar001/dmbook/ch7_clustering.pdf. [Acesso em 20 mai 2020].

THE SCIKIT-YB DEVELOPERS. **Clustering Visualizers**, 2019. Disponível em: https://www.scikit-yb.org/en/latest/api/cluster/index.html [Acesso em 15 set 2020].

8 ANEXOS

8.1 GUIA DE INSTALAÇÃO DO *PLUGIN* CLUSTERMAP PARA QGIS

Guia de Instalação do *Plugin* cluster*Map* para o QGIS

Versão Atual do Plugin: 1.0 *Versão do QGIS suportada: 3.1*

Equipe de edição:

Tiago Prudencio Silvano

Bryan Maia Correa

Agosto de 2020

Rio de Janeiro – RJ

CAPÍTULO 1

Visão Geral do *plugin* clusterMap

O *plugin* cluster*Map* foi desenvolvido por alunos do 5° Ano do Instituto Militar de Engenharia, do Exército Brasileiro, a fim de automatizar etapas de clusterização com o uso do QGIS.

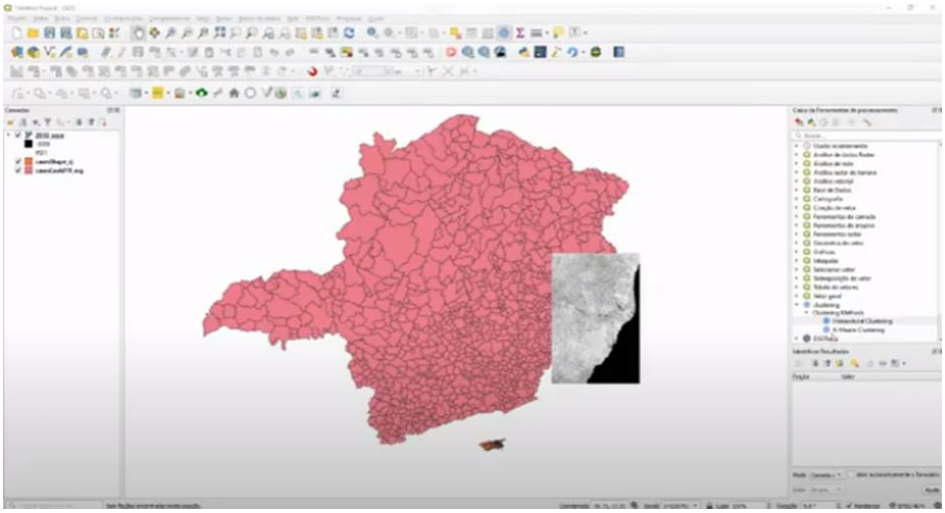
Este guia está de acordo com as funcionalidades do *plugin* e em concordância com o Caderno de Instrução de Geoinformação que pode ser encontrado na página do Geoportal da DSG.

Na sua forma nativa, o QGIS já possui um bom número de ferramentas úteis para trabalhos de Geoprocessamento. Entretanto, há certas tarefas que apenas poderão ser executadas de forma satisfatória após a instalação de *plugin*s (complementos) escritos na linguagem Python.

Entenda que quando falamos em *plugin* neste contexto estamos nos referindo a um pequeno aplicativo, ferramenta ou rotina que acrescenta funcionalidades ao QGIS.

Veremos neste guia como fazer a instalação de uma destas extensões (clusterMap), que tem como objetivo a clusterização de arquivos vetoriais utilizando diferentes métodos e métricas para tal.

No vídeo abaixo mostramos a visão geral do *plugin* numa versão inicial.



CAPÍTULO 2

Instalação e Requisitos do ClusterMap

2.1 Requisitos Mínimos

Para o funcionamento adequado do *plugin*, são necessários:

- QGIS 3.1 (ou superior) – Disponível para Windows, macOS e Linux
- Pacotes Python instalados: Numpy, Matplotlib e Scikit-learn
- *Plugin* ClusterMap: pasta contendo o *plugin* para download

2.2 Instalando o QGIS

O primeiro passo para usar o clusterMap é instalar o QGIS.

É altamente recomendável fazer o download da última versão estável e de longo prazo (LTR) do QGIS, que atualmente é a versão 3.10 Coruña ou posterior.

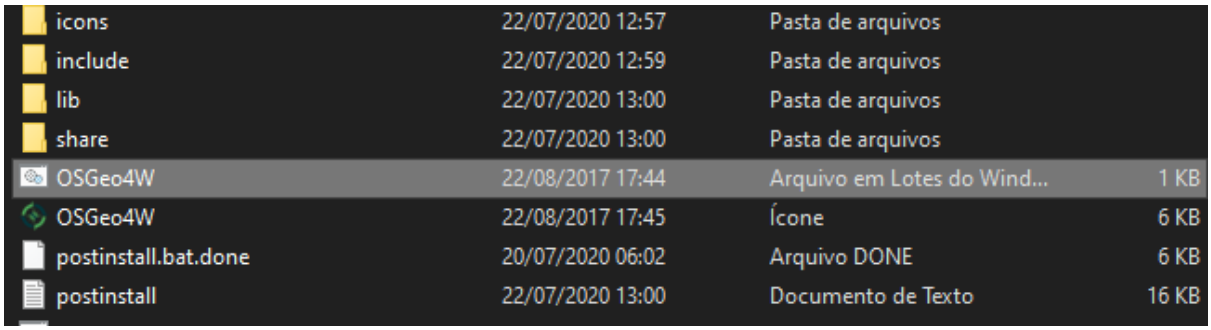
Para instalar o QGIS, consulte a documentação oficial disponível no site do QGIS, disponível em https://www.qgis.org/pt_BR/site/forusers/download.html

Nota: Para evitar conflitos durante a instalação do clusterMap, é recomendável ter **apenas uma versão do QGIS** instalada. Além disso, ter o QGIS e o ArcGIS instalados conjuntamente no mesmo sistema não é recomendado pelo mesmo motivo.

2.3 Instalando os pacotes Python: Numpy, Matplotlib e Scikit-learn

Inicialmente, para instalação dos pacotes, temos:

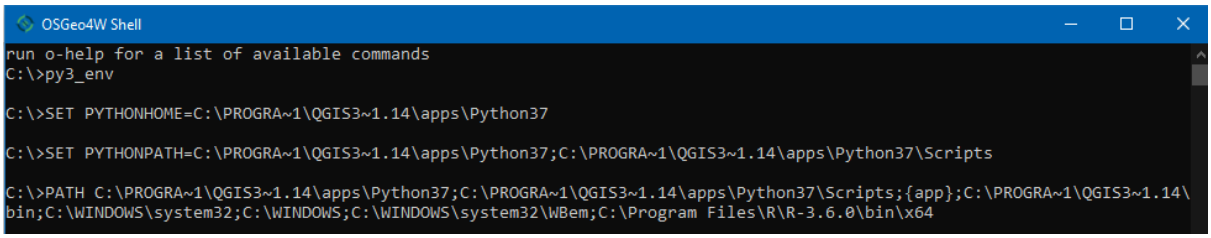
1°Passo) Abra o arquivo OSGeo4W (Arquivo em Lotes do Windows), encontrado no diretório onde o QGIS foi instalado (C:\Program Files\QGIS 3.14 ou equivalente) ou, de maneira alternativa, digite OSGeo4W na busca, canto esquerdo da barra de ferramentas.



icons	22/07/2020 12:57	Pasta de arquivos	
include	22/07/2020 12:59	Pasta de arquivos	
lib	22/07/2020 13:00	Pasta de arquivos	
share	22/07/2020 13:00	Pasta de arquivos	
OSGeo4W	22/08/2017 17:44	Arquivo em Lotes do Wind...	1 KB
OSGeo4W	22/08/2017 17:45	Ícone	6 KB
postinstall.bat.done	20/07/2020 06:02	Arquivo DONE	6 KB
postinstall	22/07/2020 13:00	Documento de Texto	16 KB

2°Passo) Aberto no OSGeo4W, é necessário entrar no console com:

```
py3_env
```



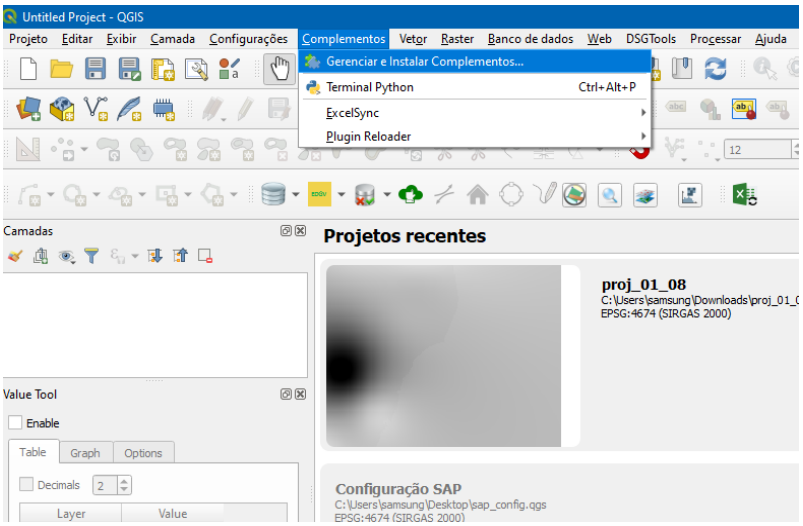
3°Passo) Em seguida é necessário entrar no console com:

```
python3 -m pip install numpy
python3 -m pip install sklearn
python3 -m pip install matplotlib
```

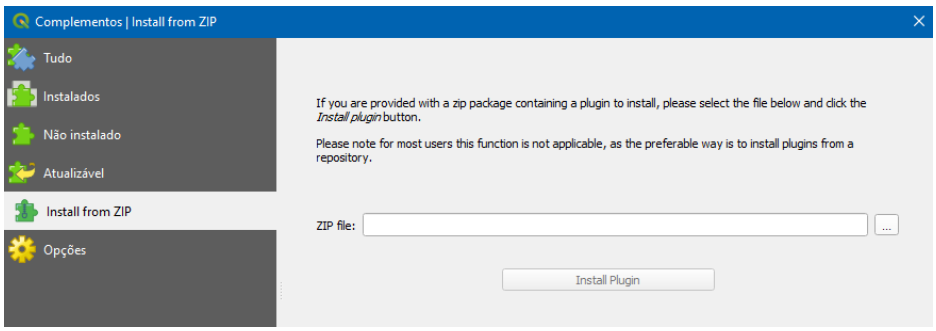
E pressionar ENTER após digitar cada uma das linhas acima.

1.4. Instalando o *plugin* clusterMap

1° Passo) Com o QGIS aberto, acesse o Menu Complementos > Gerenciar e Instalar Complementos



2° Passo) Clique em Install from ZIP e localize o arquivo zipado com o *plugin* que se deseja instalar.

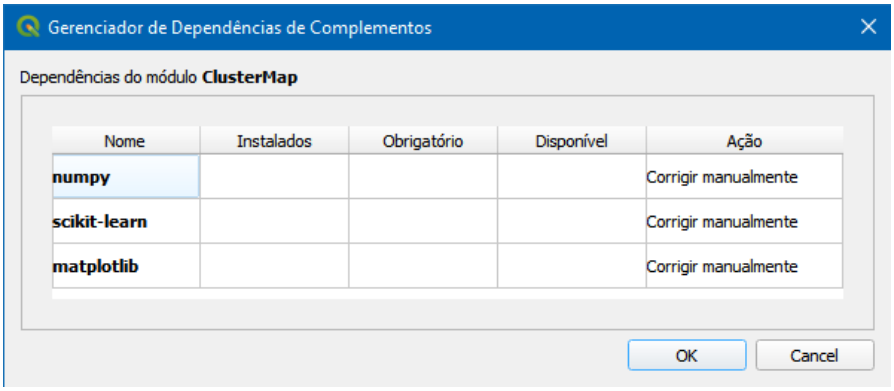


Não havendo erro durante o rápido processo de instalação, deverá aparecer uma mensagem informando sobre as dependências do módulo ClusterMap.

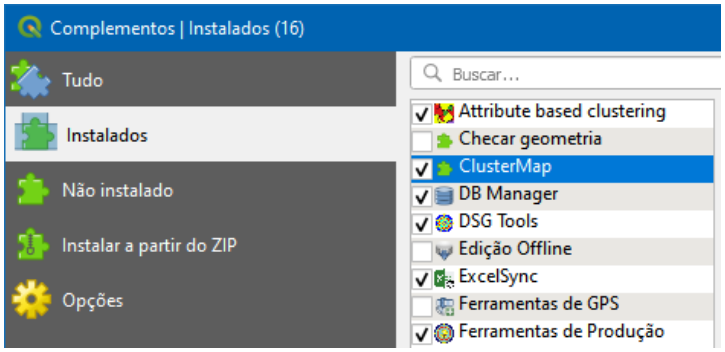
Nota: Observe que o arquivo deve ser de extensão *.zip, não podendo ser extensão *.rar

---

3° Passo) Ao verificar a caixa abaixo, basta clicar em OK.

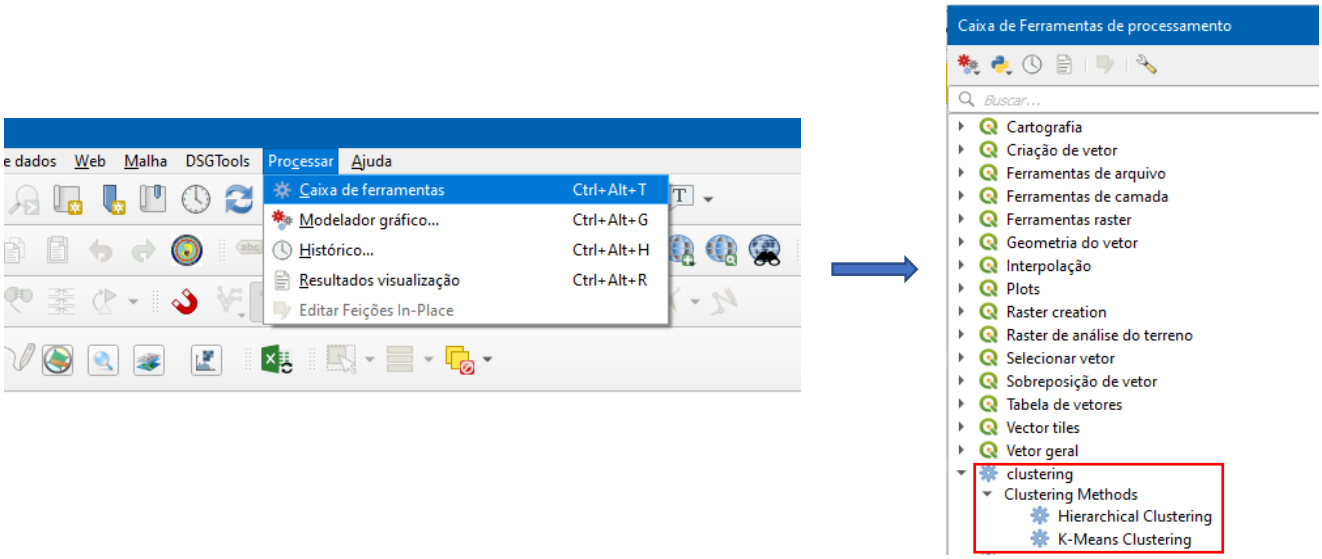


4° Passo) Com o *plugin* instalado, basta ir em Instalados e marcar a caixa de seleção de seu *plugin*.



2.5 Execução do *Plugin* clusterMap

Para executar o *plugin*, acesse Processar -> Caixa de ferramentas, ou abra utilizando o atalho *Crtl* + *Alt* + *T*.



8.2 FLUXOGRAMA DO QUESTIONÁRIO DA FASE DE TESTES

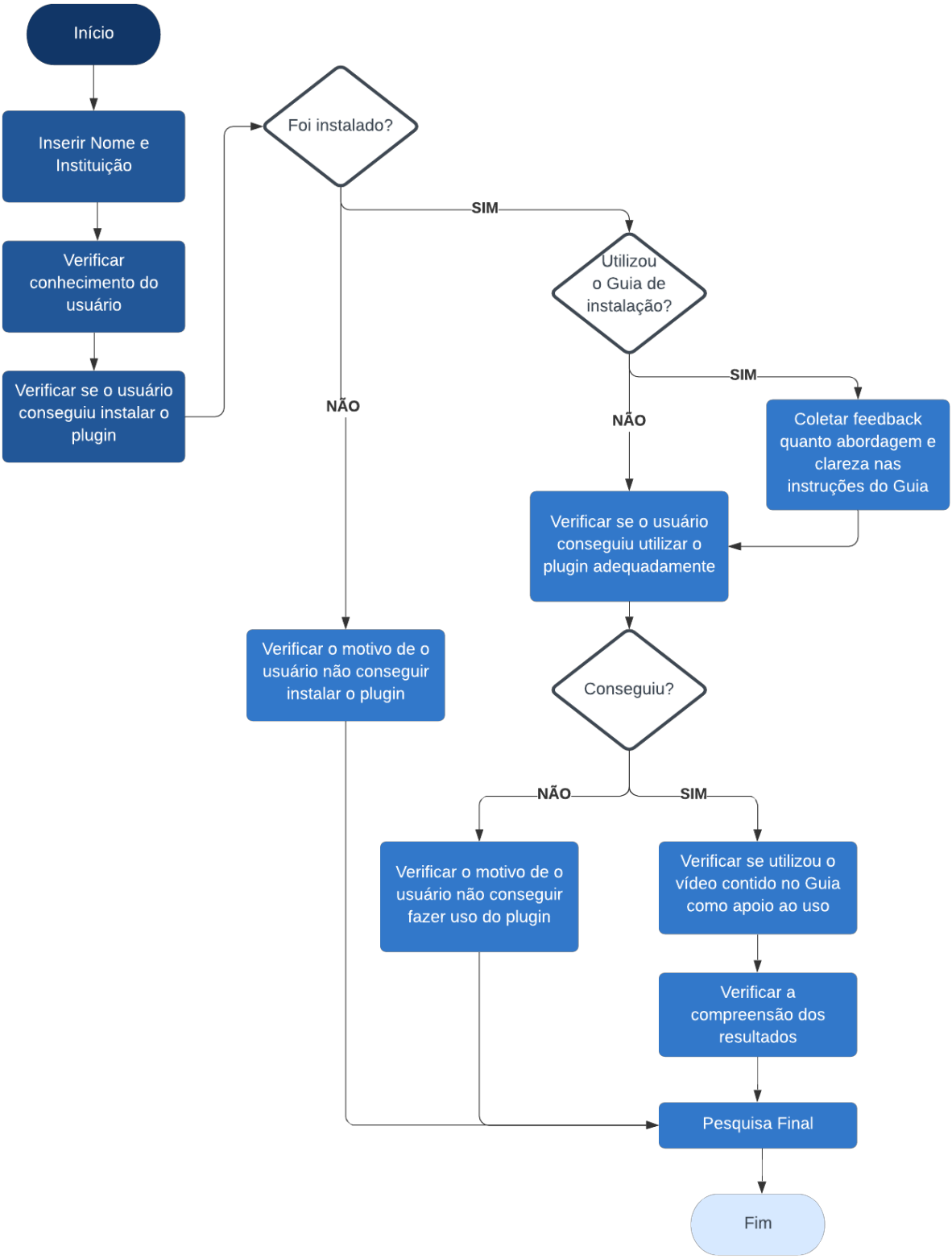


Figura 8.1 Fluxograma do Questionário.

8.3 DIAGRAMAS UML

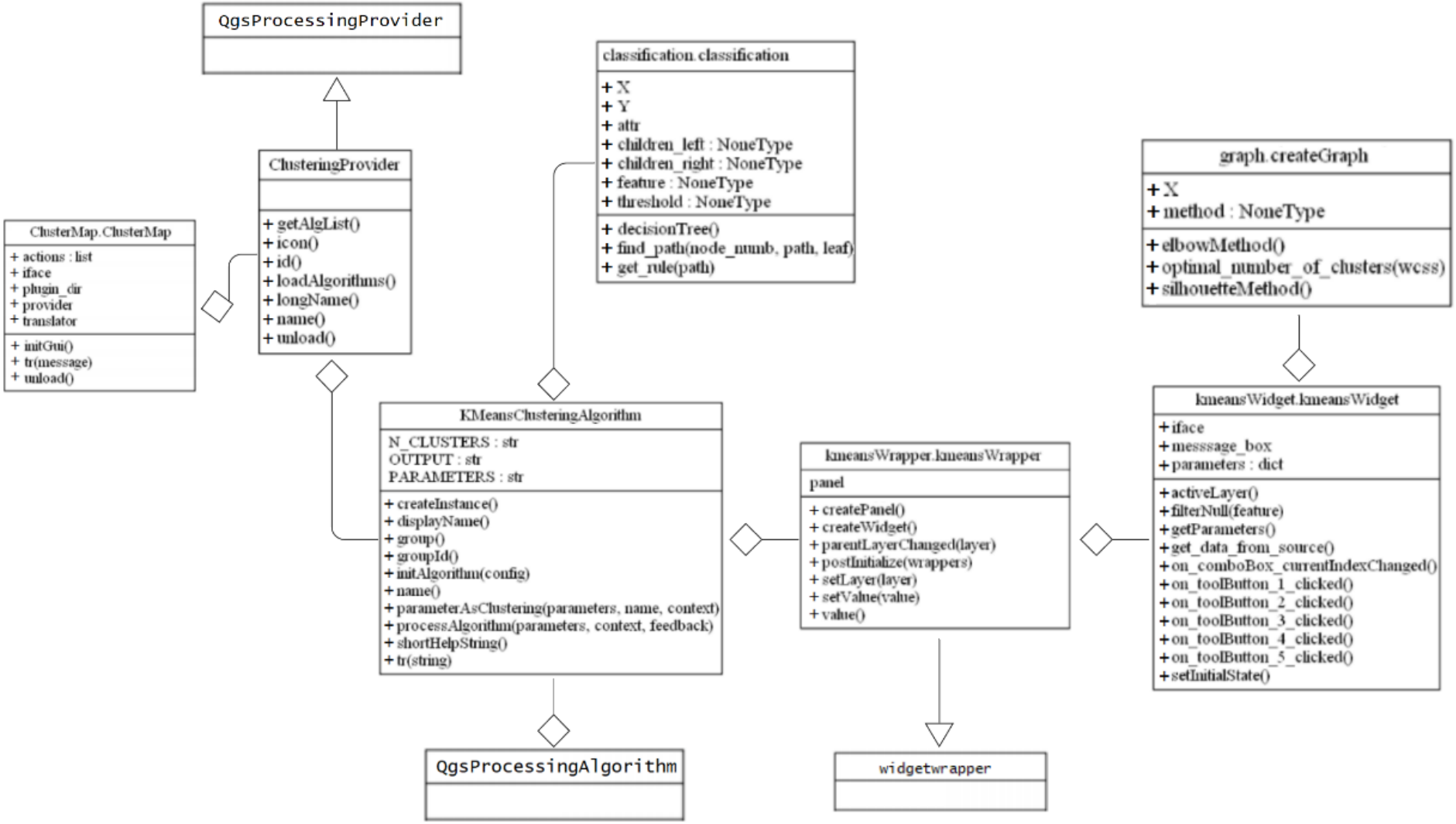


Figura 8.2 Representação de UML simplificado para o método *k-means.*

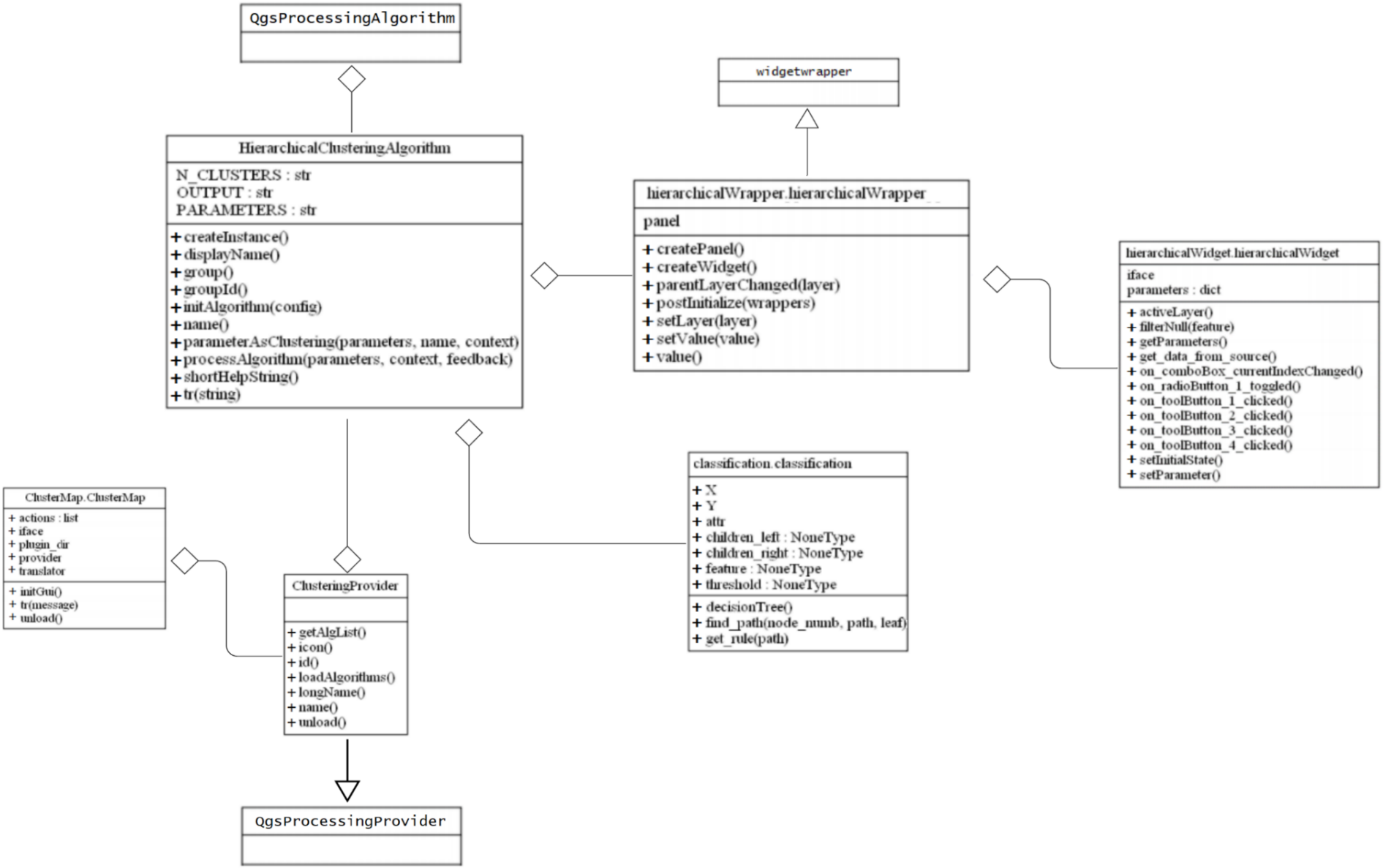


Figura 8.3 Representação de UML simplificado para o Método Hierárquico.

8.4 RESULTADOS

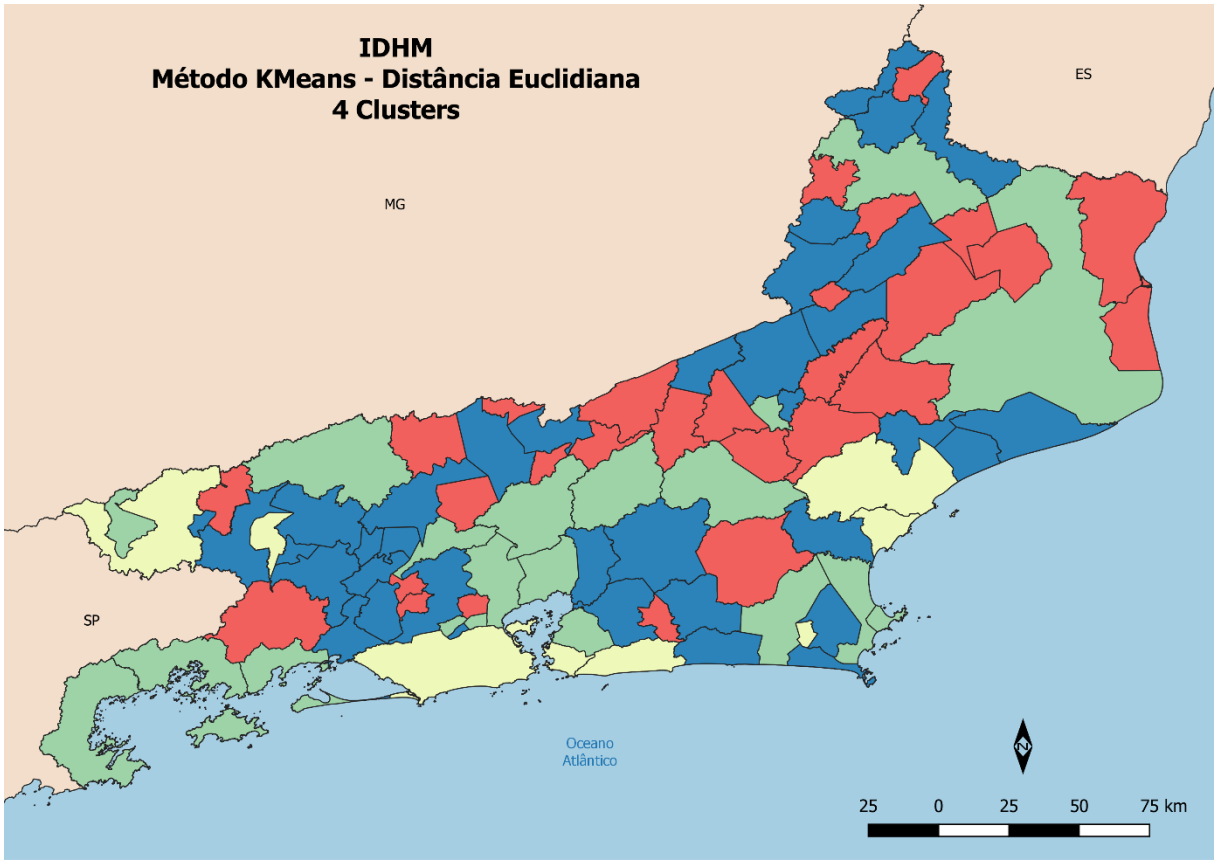


Figura 8.4 Resultado da clusterização Método K-Means.

Tabela 8.1 Regras e Coeficiente de Silhueta de cada *cluster obtidos* do Método *KMeans*

Cluster	**Regras (Árvore de Decisão)**	**Coeficiente Médio de Silhueta**
	• Longev<=0.823 Educação<=0.600 Educação>0.590 Educação<=0.594; • Longev<=0.823 Educação>0.600 Longev.>0.793 Educação<=0.603 Longev.>0.811; • Longev<=0.823 Educação>0.600 Longev.>0.793 Educação>0.603.	0.422
	• Longev<=0.823 Educação<=0.600 Educação<=0.590; • Longev<=0.823 Educação<=0.600 Educação>0.590 Educação>0.594; • Longev<=0.823 Educação>0.600 Longev<=0.793; • Longev<=0.823 Educação>0.600 Longev>0.793 Educação<=0.603 Longev<=0.811.	0.309
	• Longev>0.823 Educação<=0.679; • Longev>0.823 & Educação>0.679 Renda<=0.727.	0.281
	• Longev>0.823 Educação>0.679 Renda>0.727	0.233

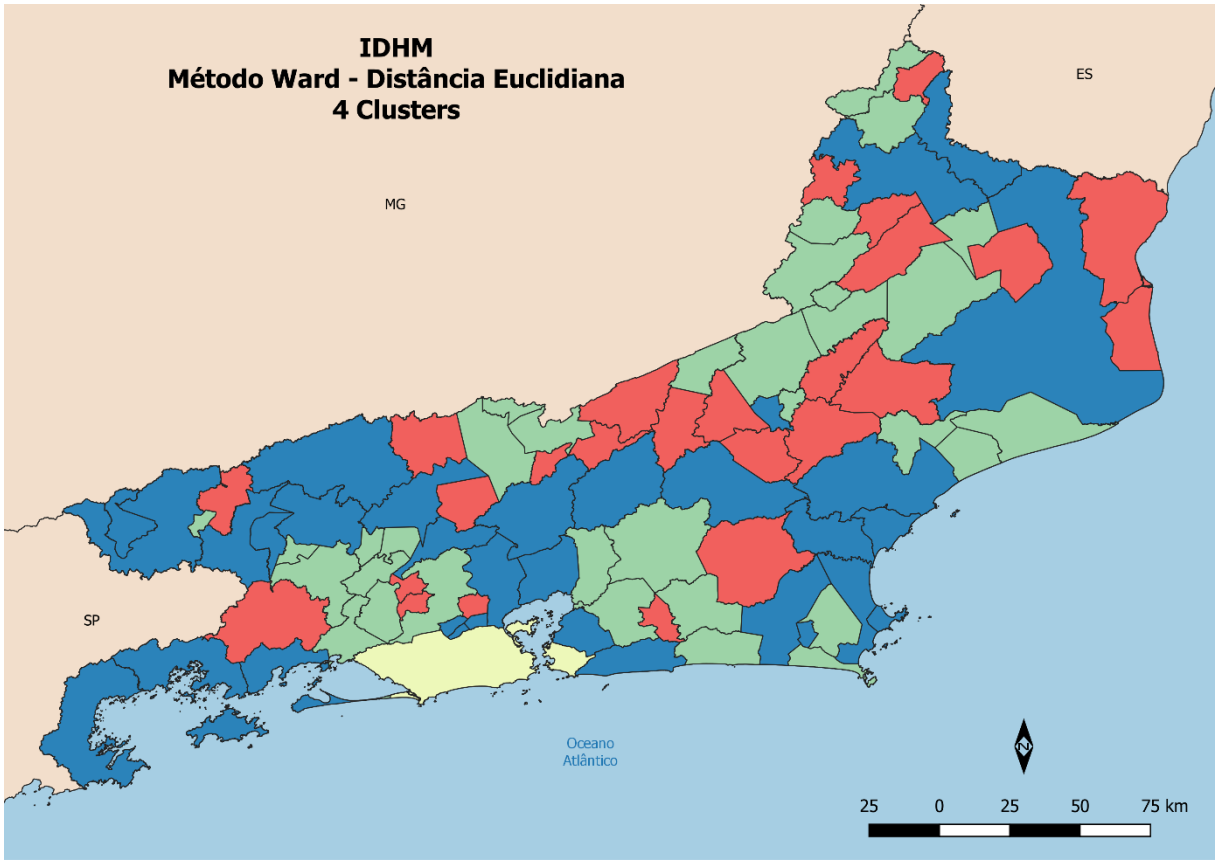


Figura 8.5 Resultado da clusterização Método *Ward.*

Tabela 8.2 Regras e Coeficiente de Silhueta de cada *cluster* obtidos do Método *Ward*

Cluster	**Regras (Árvore de Decisão)**	**Coeficiente Médio de Silhueta**
	• Longev<=0.823 Educação>0.590 Renda>0.716 Longev>0.807; • Longev>0.823 Renda<=0.815.	0.225
	• Longev<=0.823 Educação<=0.590; • Longev<=0.823 Educação>0.590 Renda<=0.716 Renda<=0.679 Longev>0.805	0.319
	• Longev<=0.823 Educação>0.590 Renda<=0.716 Renda<=0.679 Longev<=0.805; • Longev<=0.823 Educação>0.590 Renda<=0.716 Renda>0.679; • Longev<=0.823 Educação>0.590 Renda>0.716 Longev<=0.807	0.406
	• Longev>0.823 Renda>0.815	0.563

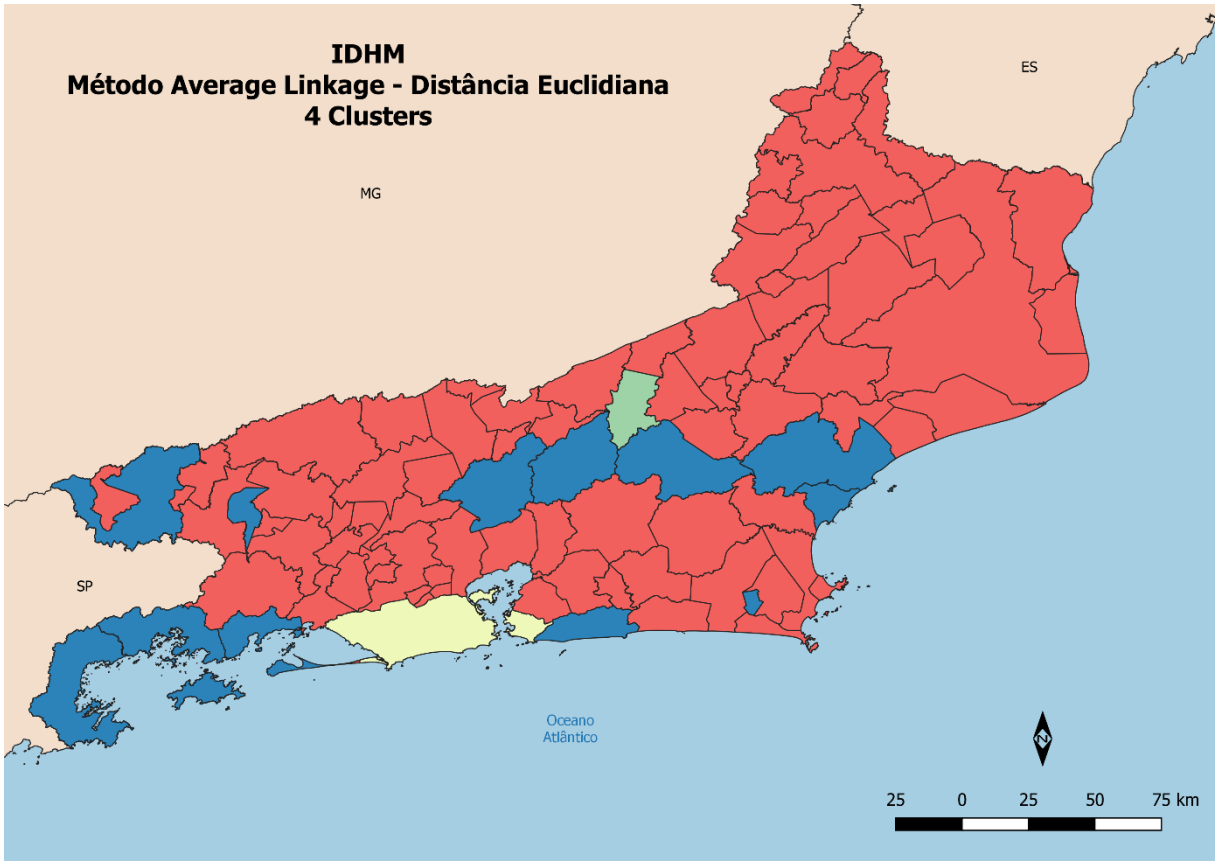


Figura 8.6 Resultado da Clusterização Método *Average Linkage*.

Tabela 8.3 Regras e Coeficiente de Silhueta de cada *cluster* obtidos do Método *Average Linkage*

Cluster	**Regras (Árvore de Decisão)**	**Coeficiente Médio de Silhueta**
	• Renda>0.725 Longev<=0.837 Renda>0.756; • Renda>0.725 Longev>0.837 Educação<=0.713.	0.451
	• Renda<=0.725 Educação>0.469; • Renda>0.725 Longev<=0.837 Renda<=0.756	0.236
	• Renda<=0.725 Educação<=0.469	0.0
	• Renda>0.725 Longev>0.837 Educação>0.713	0.469

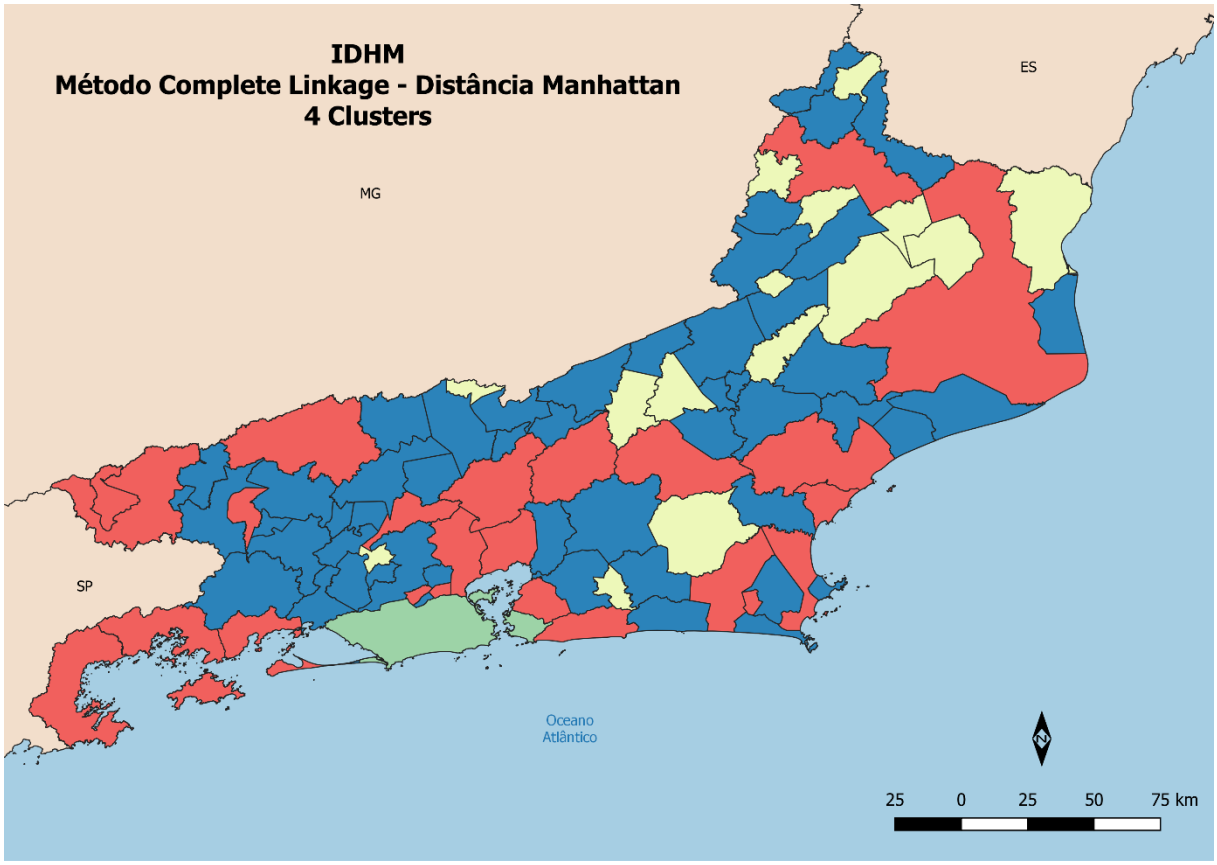


Figura 8.7 Resultado da Clusterização Método *Complete Linkage.*

Tabela 8.4 Regras e Coeficiente de Silhueta de cada *cluster* obtidos do Método *Complete Linkage*

Cluster	**Regras (Árvore de Decisão)**	**Coeficiente Médio de Silhueta**
	• Longev<=0.823 Educação>0.590 Renda>0.716 Longev>0.807; • Longev>0.823 Renda<=0.815.	0.271
	• Longev<=0.823 Educação<=0.590; • Longev<=0.823 Educação>0.590 Renda<=0.716 Renda<=0.679 Longev>0.805	0.332
	• Longev<=0.823 Educação>0.590 Renda<=0.716 Renda<=0.679500013589859 Longev<=0.805; • Longev<=0.823 Educação>0.590 Renda<=0.716 Renda>0.679; • Longev<=0.823 Educação>0.590 Renda>0.716 Longev<=0.807	0.489
	• Longev>0.823 Renda>0.815	0.373

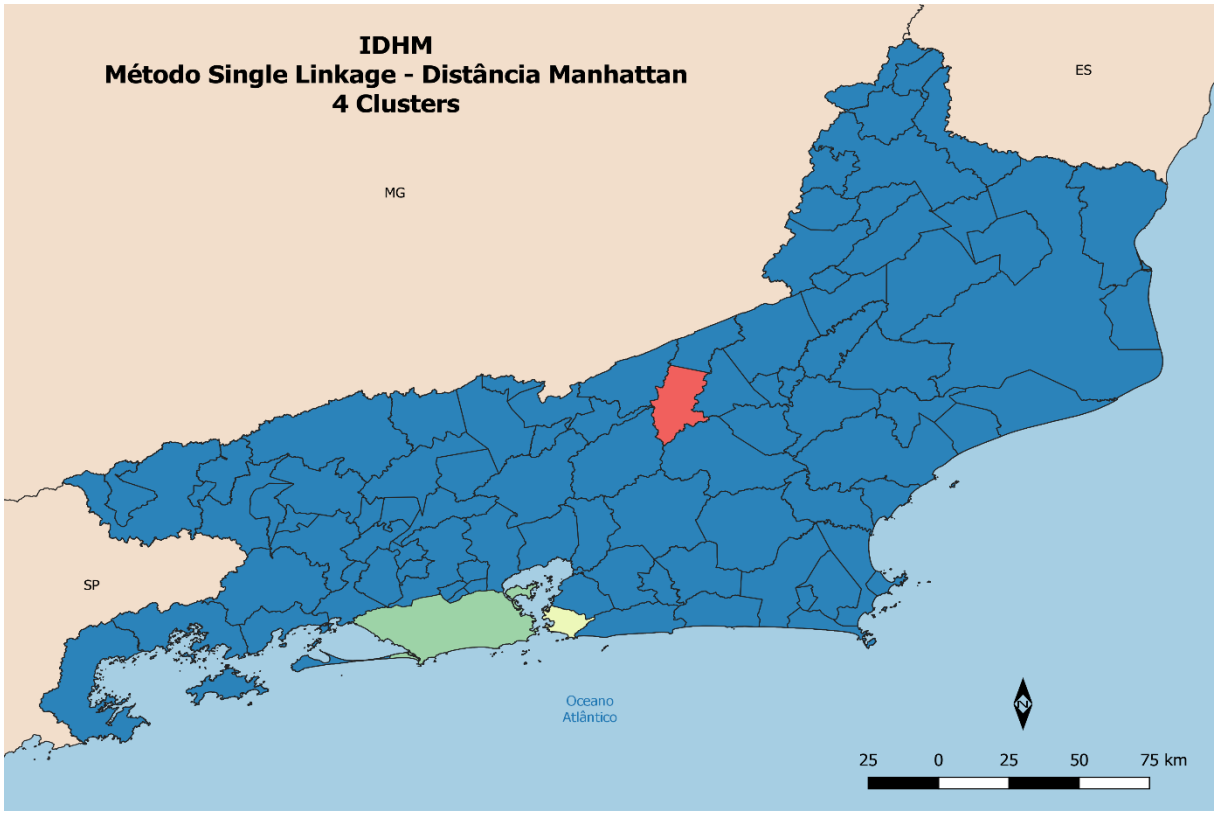


Figura 8.8 Resultado da clusterização Método *Single Linkage.*

Tabela 8.5 Regras e Coeficiente de Silhueta de cada *cluster* obtidos do Método *Single Linkage*

Cluster	**Regras (Árvore de Decisão)**	**Coeficiente Médio de Silhueta**
	• Renda<=0.81 Educação>0.469	0.236
	• Renda<=0.815 Educação<=0.469	0.0
	• Renda>0.815 Educação<=0.745	0.0
	• Renda>0.815 Educação>0.745	0.0