



UNIVERSIDADE FEDERAL DE PERNAMBUCO  
CENTRO DE TECNOLOGIA E GEOCIÊNCIAS  
DEPARTAMENTO DE ENGENHARIA CARTOGRÁFICA  
BACHARELADO EM ENGENHARIA CARTOGRÁFICA E DE AGRIMENSURA

GIVALDO CESAR DA PAIXÃO JUNIOR

**TEXTURE ANALYZER: um complemento QGIS3 para identificação de texturas em  
imagens**

Recife  
2019

GIVALDO CESAR DA PAIXÃO JUNIOR

**TEXTURE ANALYZER: um complemento QGIS3 para identificação de texturas em imagens**

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Cartográfica, do Centro de Tecnologia e Geociências da Universidade Federal de Pernambuco, como requisito para a obtenção do título de Bacharel em Engenharia Cartográfica e de Agrimensura.

**Orientador:** Prof. Dr. João Rodrigues Tavares Junior

Recife  
2019

Catálogo na fonte  
Bibliotecário Gabriel da Luz, CRB-4 / 2222

P149t Paixão Junior, Givaldo Cesar da.  
Texture analyzer: um complemento QGIS3 para identificação de  
texturas em imagens / Givaldo Cesar da Paixão Junior – Recife, 2019.  
60f., figs., quadros, siglas e abrev.

Orientador: Prof. Dr. João Rodrigues Tavares Junior.

TCC (Graduação) – Universidade Federal de Pernambuco. CTG.  
Departamento de Engenharia Cartográfica e de Agrimensura, 2019.  
Inclui Referências.

1. Engenharia Cartográfica. 2. Plugin. 3. Python. 4. QGIS. 5. Texture. 6.  
Texture analyzer. I. Tavares Junior, João Rodrigues (Orientador). II. Título.

UFPE

526.1 CDD (22. ed.) BCTG/2019 -284

GIVALDO CESAR DA PAIXÃO JUNIOR

**TEXTURE ANALYZER: um complemento QGIS3 para identificação de texturas em imagens**

Trabalho de Conclusão de Curso apresentado ao Departamento de Engenharia Cartográfica, do Centro de Tecnologia e Geociências da Universidade Federal de Pernambuco, como requisito para a obtenção do título de Bacharel em Engenharia Cartográfica e de Agrimensura.

Aprovada em: 18/06/2019.

**BANCA EXAMINADORA**

---

Prof. Dr. João Rodrigues Tavares Junior (Orientador)

Universidade Federal de Pernambuco

---

Prof. Dr. Ana Lúcia Bezerra Candeias (Examinador Interno)

Universidade Federal de Pernambuco

Dedico esse trabalho, com todo o meu orgulho, a minha mãe, Anaci, por ter me apoiado em todo esse tempo de curso, permitindo que eu o conclua sem outras preocupações.

## **AGRADECIMENTOS**

Agradeço a minha família por ter acreditado em mim e ter me apoiado por todos estes cinco longos anos, me permitindo focar neste curso e concluí-lo.

À minha companheira, Irislane, por ter sido minha base nos momentos difíceis, evitando, por muitas vezes, que eu desistisse.

Ao meu orientador Prof. Dr. João Rodrigues, por sempre estar me apresentando oportunidades, nunca esquecerei sua fé em mim desde o meu segundo período de curso, agradeço por todos os artigos e iniciações científicas que desenvolvemos.

A Prof. Dr. Ana Lúcia, por ter dedicado parte de seu tempo em correções e dicas para o engrandecimento deste trabalho.

Aos amigos adquiridos durante esta longa jornada, os quais eu sempre estarei disponível a ajudar e sei que eles sempre estarão dispostos a me ajudar.

## RESUMO

O presente trabalho traz uma adaptação para o ambiente QGIS do software *Texture Analyzer*, desenvolvido por Paixão Junior e Tavares Junior em 2018. Este software elaborado em linguagem Python têm a finalidade de classificar as texturas presentes em um conjunto de imagens de entrada utilizando um modelo matemático de fácil compreensão. Aqui estão descritos todo o procedimento para a produção de um complemento QGIS3, que servirá de base para todo aquele que desejar criar suas próprias ferramentas. Após a concepção do complemento foram feitos diversos testes para avaliar o desempenho deste. Inicialmente, foi executado testes em uma imagem sintética que verificou a necessidade de alteração do antigo modelo matemático, resultando em uma melhor classificação. Em seguida, foram feitos testes para analisar o tempo de processamento, a fim de perceber o impacto do número de pixels de uma imagem neste parâmetro, tendo como resultado o gráfico apresentado. Também foram executados testes em imagens obtidas de diversos sensores óticos e RADAR com o intuito de perceber o potencial de mapeamento de recursos hídricos do complemento, para tal foram construídos e comparados os gráficos obtidos a partir de dados de perfis sobre uma banda do sensor e a classe binária correspondente as massas d'água, apresentando bons resultados com erros em torno de 1,5 pixel. Além disto foi verificada o índice de correlação entre a imagem textura resultante do processamento e cada banda do sensor, observando valores maiores que 0,7, indicando boa correlação.

Palavras-chave: Complemento. Python. QGIS. Texturas, Texture Analyzer.

## **ABSTRACT**

The present work brings an adaptation for the QGIS environment of the Texture Analyzer software, developed by Paixão Junior and Tavares Junior in 2018. This software developed in Python language has the purpose of classifying the textures present in a set of input images using a mathematical model of easy understanding. Here is a description of the procedure for producing a QGIS3 plugin, which will be the basis for anyone who wants to create your own tools. After completing the design, several tests were done to evaluate the performance of this. Initially, tests were run on a synthetic image that verified the need to change the old mathematical model, resulting in a better classification. Then tests were performed to analyze the processing time in order to realize the impact of the number of pixels of an image in this parameter, resulting in the graph presented. We also performed tests on images obtained from several optical and RADAR sensors in order to understand the potential of mapping of water resources of the complement, for this were constructed and compared the graphs obtained from profile data on a sensor band and the binary class corresponding to the water masses, presenting good results with errors around 1.5 pixels. In addition, the correlation index between the texture image resulting from the processing and each sensor band was verified, observing values greater than 0.7, indicating a good correlation.

**Keywords:** Plugin. Python. QGIS. Textures, Texture Analyzer.

## LISTA DE FIGURAS

Figura 1 –	O espectro eletromagnético e suas principais regiões.....	21
Figura 2 –	Software Texture Analyzer.....	29
Figura 3 –	Resultados obtidos pelo software.....	30
Figura 4 –	Imagens binárias resultantes.....	31
Figura 5 –	Localização dos perfis utilizados.....	31
Figura 6 –	Perfis utilizados.....	31
Figura 7 –	Ícones do Plugin Builder e Plugin Reload.....	33
Figura 8 –	Janela Inicial do Plugin Builder.....	34
Figura 9 –	Janela resultante do Plugin Builder.....	35
Figura 10 –	Janela inicial Texture Analyzer.....	36
Figura 11 –	Novo ícone do Texture Analyzer.....	37
Figura 12 –	Tela Inicial Qt Designer.....	38
Figura 13 –	Tela editada do Texture Analyzer.....	39
Figura 14 –	Tela inicial PyCharm.....	40
Figura 15 –	Texture Analyzer no ambiente QGIS.....	41
Figura 16 –	Comparação Imagem-teste X Resultado.....	42
Figura 17 –	Imagem binária com confusão de classes.....	42
Figura 18 –	Comparação entre resultados sem a medida da média e com a medida.....	43
Figura 19 –	Evolução do tempo de execução.....	44
Figura 20 –	Área de Estudo.....	45
Figura 21 –	Recorte da Área de Estudo.....	46
Figura 22 –	Valores GVF.....	50
Figura 23 –	Comparação Perfil Banda 7 – TM x Imagem Binária.....	51
Figura 24 –	Comparação Perfil Banda 7 – ETM+ x Imagem Binária.....	52
Figura 25 –	Comparação Perfil Banda 7 – OLI x Imagem Binária.....	52
Figura 26 –	Comparação Perfil Banda 8 – MUX x Imagem Binária.....	53
Figura 27 –	Comparação Perfil Banda 11 – IRS x Imagem Binária.....	53
Figura 28 –	Imagens-Binárias e perfis de classes.....	54
Figura 29 –	Comparação Perfil Polarização HH – PALSAR FBS x Imagem Binária.....	54
Figura 30 –	Comparação Perfil Polarização HH – PALSAR FBD x Imagem Binária.....	55
Figura 31 –	Estrutura necessária.....	56

## LISTA DE QUADROS

Quadro 1 –	Interpretação dos coeficientes de correlação.....	18
Quadro 2 –	Características do Sensor TM.....	23
Quadro 3 –	Características do Sensor ETM+.....	24
Quadro 4 –	Características do Sensor OLI.....	25
Quadro 5 –	Características do Sensor MUX.....	25
Quadro 6 –	Características do Sensor IRS.....	26
Quadro 7 –	Bandas de operação dos sistemas RADAR e aplicações.....	27
Quadro 8 –	Características do sensor PALSAR.....	28
Quadro 9 –	Campos requeridos pelo Plugin Builder.....	34
Quadro 10 –	Características dos materiais de estudo.....	46
Quadro 11 –	Resultado da classificação.....	47

## LISTA DE SIGLAS E ABREVIACÕES

B	Blue
CBERS	China-Brazil Earth Resources Satellite
DBV	Deep Blue-Violet
DPI	Divisão de Processamento de Imagens
G	Green
INPE	Instituto Nacional de Pesquisas Espaciais
NASA	National Aeronautics and Space Administration
ND	Número Digital
NIR	Near InfraRed
OS	Operational System
p. ex.	Por exemplo
Pan	Pancromática
PSF	Python Software Foundation
QGIS3	Quantum GIS versão 3
R	Red
SIG	Sistema de Informações Geográficas
SWIR	Short-Wave InfraRed
TIR	Thermal InfraRed
USGS	United States Geological Survey

## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO.....</b>	<b>13</b>
1.1	OBJETIVOS.....	13
<b>1.1.1</b>	<b>Objetivo geral.....</b>	<b>13</b>
<b>1.1.2</b>	<b>Objetivos Específicos.....</b>	<b>14</b>
<b>2</b>	<b>FUNDAMENTAÇÃO TEÓRICA.....</b>	<b>15</b>
2.1	TEXTURA.....	15
2.2	MEDIDAS RESUMO.....	15
<b>2.2.1</b>	<b>Média.....</b>	<b>15</b>
<b>2.2.2</b>	<b>Mediana.....</b>	<b>15</b>
<b>2.2.3</b>	<b>Variância e desvio padrão.....</b>	<b>15</b>
<b>2.2.4</b>	<b>Desvio Mediano.....</b>	<b>16</b>
2.3	FÓRMULA DE STURGES.....	16
2.4	OTIMIZAÇÃO DE JENKS.....	17
2.5	CORRELAÇÃO LINEAR DE DADOS.....	17
2.6	LINGUAGEM PYTHON.....	18
2.7	SOFTWARE QGIS.....	19
<b>2.7.1</b>	<b>Complementos QGIS.....</b>	<b>20</b>
2.8	SENSORIAMENTO REMOTO.....	20
<b>2.8.1</b>	<b>Sistemas imageadores eletro-óticos.....</b>	<b>22</b>
2.8.1.1	Sistema TM.....	23
2.8.1.2	Sistema ETM+.....	23
2.8.1.3	Sistema OLI.....	24
2.8.1.4	Sistema MUX.....	25
2.8.1.5	Sistema IRS.....	25
<b>2.8.2</b>	<b>Sistemas RADAR.....</b>	<b>26</b>
2.8.2.1	Sistema PALSAR.....	27
<b>2.8.3</b>	<b>Classificação de imagens.....</b>	<b>28</b>
2.9	TEXTURE ANALYZER (2018) .....	29
<b>3</b>	<b>METODOLOGIA.....</b>	<b>32</b>
3.1	DESENVOLVIMENTO DO COMPLEMENTO.....	32
<b>3.1.1</b>	<b>Materiais utilizados nesta etapa.....</b>	<b>32</b>
<b>3.1.2</b>	<b>Procedimentos.....</b>	<b>32</b>
3.1.2.1	Aquisição dos complementos necessários.....	33
3.1.2.2	Execução do Plugin Builder.....	33
3.1.2.3	Instalação do complemento gerado no QGIS.....	35
3.1.2.4	Customização do ícone.....	37
3.1.2.5	Customização da interface do usuário.....	37
3.1.2.6	Edição dos arquivos Python .....	39
<b>4</b>	<b>RESULTADOS E DISCUSSÕES.....</b>	<b>42</b>
4.1	TESTE INICIAL COM IMAGEM SINTÉTICA.....	42
4.2	ANÁLISE DO TEMPO DE EXECUÇÃO.....	43
4.3	TESTES EM IMAGENS GERADAS POR SENSORES ORBITAIS....	45
<b>4.3.1</b>	<b>Características da área de estudo.....</b>	<b>45</b>
<b>4.3.2</b>	<b>Materiais.....</b>	<b>46</b>
<b>4.3.3</b>	<b>Classificação das imagens.....</b>	<b>47</b>
<b>4.3.4</b>	<b>Valores GVF.....</b>	<b>49</b>
<b>4.3.5</b>	<b>Coefficientes de correlação.....</b>	<b>50</b>

<b>4.3.6</b>	<b>Análises de perfis.....</b>	<b>51</b>
4.4	PUBLICAÇÃO DO COMPLEMENTO.....	56
<b>5</b>	<b>CONCLUSÃO E RECOMENDAÇÕES.....</b>	<b>57</b>
	<b>REFERÊNCIAS.....</b>	<b>58</b>

## 1 INTRODUÇÃO

Existem muitos softwares capazes de mapear texturas através do uso de algoritmos complexos e de propriedades físicas dos alvos, no entanto, estes softwares necessitam de uma licença, às vezes com alto custo, além de não disponibilizar os algoritmos utilizados, sendo assim, o usuário opera uma ferramenta que não compreende, dificultando a análise de resultados.

Uma opção gratuita para estes softwares é o ambiente QGIS, software SIG de código aberto e ampla comunidade ativa, além de possuir uma grande quantidade de complementos disponíveis produzidos por seus próprios usuários.

No entanto, nem sempre existe uma ferramenta para um caso específico, sendo necessário o usuário desenvolver por si. Porém não existem muitos tutoriais em português que guiem o desenvolvimento, dificultando a produção.

Pensando nessa dificuldade, aqui é proposto a adaptação do software *Texture Analyzer* desenvolvido para o ambiente QGIS. Descrevendo todo o procedimento para a elaboração de um complemento funcional, que servirá de base para todos que desejarem desenvolver suas ferramentas.

O *Texture Analyzer* gera como resultado uma classificação da região utilizando um conjunto de imagens de entrada utilizando um algoritmo simples e não supervisionado. Além de fornecer imagens binárias para cada classe encontrada.

Além disso, foram executados experimentos com o *Texture Analyzer*, já funcionando dentro do ambiente QGIS, com o intuito de verificar o potencial de mapeamento de recursos hídricos nos sensores TM, ETM+, OLI, MUX, IRS e ALOS PALSAR, além de verificar o impacto da quantidade de pixels em seu tempo de execução.

### 1.1 OBJETIVOS

Este Trabalho de Conclusão de Curso tem seus objetivos dispostos de acordo com os itens 1.1.1 e 1.1.2.

#### 1.1.1 Objetivo Geral

Adaptar o software *Texture Analyzer*, desenvolvido como um complemento para o software QGIS3.

### 1.1.2 Objetivos Específicos

- a) Desenvolver um complemento para o software QGIS3 capaz de ler os valores ND de cada pixel, calcular a textura utilizando o modelo matemático do *Texture Analyzer* e então classificá-las pelo método de quebras naturais.
- b) Testar e validar os resultados obtidos pelo complemento em diversos tipos de imagens.
- c) Encontrar meios para a publicação do complemento.

## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção serão expostos os conceitos e definições que fornecem embasamento teórico para este trabalho.

### 2.1 TEXTURA

No contexto de imagens espectrais de sensoriamento remoto, textura é uma propriedade importante na percepção de regiões e de superfícies, contendo informações sobre a distribuição espacial das variações de tonalidade dos objetos. Na imagem, a textura se manifesta pela variação quase periódica de valores digitais (tons), que tem origem na reflectância dos objetos, sombras e sombreamentos devidos à topografia do relevo. O esforço é tentar reconhecer variações de padrões de formas por meio de uma análise das variações de padrões de tons de cinza. Não há uma definição geral para textura tonal e tal dificuldade é refletida pelo grande número de métodos de análise de texturas encontrados na literatura. (MENESES e ROSA, 2012)

### 2.2 MEDIDAS RESUMO

Segundo Magalhães e Lima (2004), estas medidas buscam sumarizar as informações disponíveis sobre o comportamento de uma variável, caracterizando assim um conjunto de dados, por exemplo, apresentando a tendência central ou a forma como estão dispersos.

#### 2.2.1 Média

A média ( $\bar{x}$ ) de um conjunto de dados é a soma de todos os valores dividida pelo número total de observações ( $n$ ).

$$\bar{x} = \frac{\sum_{i=1}^n x_i}{n} \quad (1)$$

#### 2.2.2 Mediana

A mediana ( $Md$ ) é o valor que ocupa o valor central do conjunto de dados ordenados.

#### 2.2.3 Variância e Desvio Padrão

A variância ( $var$ ) é uma medida de variabilidade obtida por:

$$var = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 \quad (2)$$

Para manter a mesma unidade dos dados de originais, convenientemente definiu-se o desvio padrão ( $\sigma$ ) como:

$$\sigma = \sqrt{var} \quad (3)$$

#### 2.2.4 Desvio Mediano

Uma maneira alternativa de obter a variabilidade de um conjunto de dados é através do desvio mediano ( $DMd$ ), este é definido por:

$$DMd = \frac{1}{n} \sum_{i=1}^n |x_i - Md| \quad (4)$$

### 2.3 FÓRMULA DE STURGES

Segundo Scott (2009), a construção de um histograma começa com a escolha de um intervalo de suporte,  $I = (a, b)$ . Os valores utilizados, geralmente, são o menor e maiores valores da amostra ou alguns valores convenientes que cobrem todos os dados. Dado o intervalo  $I = (a, b)$ , um histograma igualmente espaçado ou diagrama de frequência é construído a partir da divisão de  $I$  em  $k$  sub-intervalos de espaçamento igual. A largura de cada sub-intervalo é denotada por  $h$ , onde:

$$h = \frac{b - a}{k} \quad (5)$$

Claramente, quanto maior o tamanho da amostra  $n$ , mais sub-intervalos podem ser formados. Sturges propôs em 1926, a seguinte estimativa para o número  $k$  de subintervalos:

$$k = 1 + \log_2 n = 1 + 3,322 * \log n \quad (6)$$

Esta simples regra é facilmente implementada e é amplamente utilizada em softwares de estatísticas hoje.

## 2.4 OTIMIZAÇÃO DE JENKS

Na otimização de Jenks ou quebras naturais é definido um fator GVF (*Goodness of Variance Fit*) que tende a um, para os melhores métodos de classificação, enquanto a soma da variação dentro de cada uma das classes é minimizada. Assim, o GFV é usado para determinar os limites de classe no método de otimização, assim como é um indicador da qualidade da classificação. O GVF é calculado da seguinte forma: (GALANT, 2006)

$$GVF = \frac{SDAM - SDCM}{SDAM} \quad (7)$$

Onde SDAM é a soma dos desvios quadrados da média da matriz de dados e SDCM é a soma dos desvios quadrados da média das classes. Essas medidas são mais bem definidas por Armstrong et al. (2003):

$$SDAM = \sum_{i=1}^N (z_i - \bar{z})^2 \quad (8)$$

$$SDCM = \sum_{j=1}^k \sum_{i=1}^{N_j} (z_{ij} - \bar{z}_j)^2 \quad (9)$$

Onde  $z_i (i = 1, \dots, N)$  são os valores observados,  $\bar{z}$  é a média dos valores observados,  $k$  é o número de classes existentes,  $z_{ij}$  é o valor observado na classe  $j$  e  $\bar{z}_j$  é a média da classe  $j$ .

A ESRI (2016) descreve o algoritmo da seguinte forma:

1. Especifica-se inicialmente um agrupamento arbitrário dos dados numéricos.
2. Calcula-se o SDAM. Este é constante não muda a menos que os dados sejam alterados.
3. Calcula-se a média de cada classe e então o SDCM.
4. Calcula-se o GVF.
5. Movem-se as observações de uma classe para outra, num esforço para aumentar o GVF.
6. O processo continua até o GVF chegar ao máximo.

## 2.5 CORRELAÇÃO LINEAR DE DADOS

Magalhães (2004) diz que para um conjunto de dados com  $n$  pares de valores as variáveis  $X$  e  $Y$ , representados por  $(x_i, y_i)$ ,  $i = 1, 2, \dots, n$ . O coeficiente de correlação mede a dependência linear entre as variáveis e é calculado da seguinte forma:

$$\rho_{X,Y} = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{[\sum_{i=1}^n (x_i - \bar{x})]^2 [\sum_{i=1}^n (y_i - \bar{y})]^2}} \quad (10)$$

É possível verificar que o coeficiente de correlação é sempre um número entre -1 e 1. Um ponto a ser ressaltado, que muitas vezes causa confusão, é correlação igual a zero não indica independência. Se a correlação é zero, o que se pode dizer é que não existe dependência linear entre as variáveis.

Shimakura (2006) usa o termo correlação positiva quando o valor de  $\rho$  é maior que 0, o que indica que a medida que o valor de X cresce o de Y também cresce, e correlação negativa quando  $\rho$  é menor que 0, neste caso a medida que o valor de X cresce, o de Y decresce, em média. Quanto maior o valor do módulo de  $\rho$ , mais forte a associação. No extremo, se  $\rho$  igual a 1 ou a -1, então todos os pontos no gráfico de dispersão caem exatamente numa linha reta.

O quadro 1 fornece um guia de como é possível descrever uma correlação em palavras dado um valor numérico.

**Quadro 1** Interpretação dos coeficientes de correlação

Valor de $\rho$ (+ ou -)	Interpretação
<b>0.00 a 0.19</b>	Correlação bem fraca
<b>0.20 a 0.39</b>	Correlação fraca
<b>0.40 a 0.69</b>	Correlação moderada
<b>0.70 a 0.89</b>	Correlação forte
<b>0.90 a 1.00</b>	Correlação muito forte

Fonte: Shimakura, 2006

Considerando uma imagem como um conjunto de dados, o conceito é similar e pode-se avaliar a correlação entre imagens e a equação 10 pode ser adaptada para duas dimensões:

$$\rho_{X,Y} = \frac{\sum_{i=1}^n \sum_{j=1}^m (x_{ij} - \bar{x})(y_{ij} - \bar{y})}{\sqrt{[\sum_{i=1}^n \sum_{j=1}^m (x_{ij} - \bar{x})]^2 [\sum_{i=1}^n \sum_{j=1}^m (y_{ij} - \bar{y})]^2}} \quad (11)$$

Onde  $i,j$  é a posição do pixel na imagem X e Y, e  $n,m$  as dimensões destas.

## 2.6 LINGUAGEM PYTHON

De acordo com a PSF (2019), Python é uma linguagem de programação interpretada, interativa e orientada a objetos. Ele incorpora módulos, exceções, tipagem dinâmica, tipos de dados dinâmicos de nível muito alto e classes. O Python combina um poder notável com uma

sintaxe muito clara. Possui interfaces para muitas chamadas de sistema e bibliotecas, bem como para vários sistemas de janelas, e é extensível em C ou C ++. Também é utilizável como uma linguagem de extensão para aplicativos que precisam de uma interface programável. Por fim, o Python é portátil: ele é executado em muitas variantes do Unix, no Mac e no Windows 2000 e posterior.

Ainda de acordo com a PSF(2019), Python é uma linguagem de programação de alto nível com propósito geral podendo ser aplicada a muitas classes diferentes de problemas. A linguagem vem com uma imensa biblioteca padrão que cobre áreas como processamento de *strings* (expressões regulares, Unicode, cálculo de diferenças entre arquivos), protocolos de Internet (HTTP, FTP, SMTP, XML-RPC, POP, IMAP, programação CGI), engenharia de software (teste unitário, criação de log, criação de perfil, análise de código Python) e interfaces do sistema operacional (chamadas do sistema, sistemas de arquivos, soquetes TCP / IP). Para se ter uma ideia do que está disponível é necessário acessar o índice da biblioteca padrão, disponibilizado no link <https://docs.python.org/3/library/index.html#library-index>. Uma grande variedade de extensões criadas por terceiros também está disponível no link <https://pypi.org/>.

Como um projeto popular de desenvolvimento de software livre, o Python tem uma comunidade de suporte ativa de colaboradores e usuários que também disponibilizam seus softwares para outros desenvolvedores de Python usarem sob termos de licença de código aberto. Isso permite que os usuários compartilhem e colaborem de forma eficaz, aproveitando as soluções que outros já criaram para problemas comuns e raros.

## 2.7 SOFTWARE QGIS

Segundo o QGIS Development Team (2019), o QGIS é um Sistema de Informações Geográficas de Código Aberto com interface amigável ao usuário, além de fornecer funções e recursos de uso comum. O projeto nasceu em maio de 2002 e foi estabelecido como um projeto no SourceForge em junho do mesmo ano com o intuito de criar um software SIG (os quais possuem licenças tradicionalmente caras) com uma perspectiva viável para qualquer pessoa com acesso básico a um computador pessoal. Atualmente, o QGIS esta na versão 3.6 e é executado na maioria das plataformas Unix, Windows e macOS, sendo desenvolvido usando o

kit de ferramentas Qt<sup>1</sup> e C++. O QGIS suporta a um vasto número de formatos de dados raster e vetor, com novos formatos sendo facilmente utilizados usando a arquitetura de complementos.

### 2.7.1 Complementos QGIS

O QGIS foi desenvolvido com uma arquitetura de complementos, o que permite que muitos recursos e funções sejam facilmente adicionados. Os complementos são implementados como *Core Plugins* ou *External Plugins*. Core Plugins são mantidos pelo grupo de desenvolvimento e são distribuídos junto com o próprio QGIS.

External Plugins geralmente são escritos em Python e são armazenados no repositório “oficial” no link <https://plugins.qgis.org/plugins/> ou em repositórios externos mantidos por autores individuais. No repositório oficial é disponibilizada toda a documentação sobre uso dos complementos, além versão mínima requerida do QGIS, *home page*, autores e outras informações importantes. Já nos outros repositórios as informações devem ser obtidas juntas com o próprio complemento.

Para instalar ou ativar um complemento é necessário ir ao menu “Complementos” e selecionar “ Gerenciar e Instalar Complementos...”. Complementos externos em Python são postos na pasta “*python/plugins*” do diretório do perfil de usuário ativo.

## 2.8 SENSORIAMENTO REMOTO

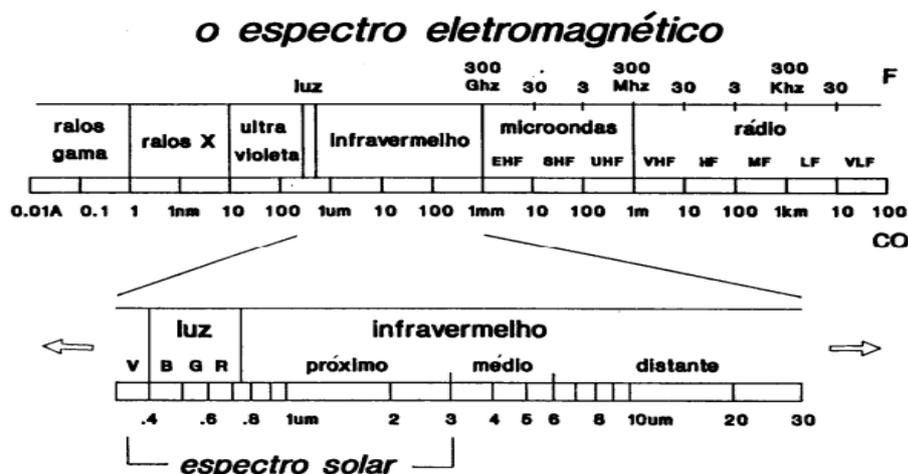
Novo e Ponzoni (2001) dizem que a definição clássica do termo sensoriamento remoto se refere a um conjunto de técnicas destinado à obtenção de informação sobre objetos, sem que haja contato físico com eles. Tendo como elementos fundamentais: a radiação eletromagnética, uma fonte emissora de radiação, um sensor e um alvo.

A radiação eletromagnética pode ser explicada por dois modelos: o corpuscular, ou quântico, e o ondulatório. Sob o ponto de vista quântico, a radiação é resultado da emissão de pequenos pulsos energéticos, enquanto sob a perspectiva ondulatória, se propaga no vácuo, na velocidade da luz e na forma de ondas geradas pela oscilação dos campos elétrico e magnético. No modelo ondulatório a radiação é caracterizada em comprimentos de onda ( $\lambda$ ), que representam a distância entre dois pontos de igual intensidade destes campos. O conjunto destes comprimentos é conhecido com espectro eletromagnético, o qual foi dividido em certo número de regiões espectrais, conforme figura 1.

---

<sup>1</sup> Mais informações em <https://www.qt.io>

Figura 1 O espectro eletromagnético e suas principais regiões.



Fonte: Novo e Ponzoni, 2010

A aquisição de informação sobre os alvos é viabilizada pelos sensores, que são instrumentos capazes de coletar e registrar a radiação refletida ou emitida pelos objetos. Os sensores se diferenciam entre si pela forma de funcionamento e suas características (resoluções espacial, espectral e radiométrica).

A resolução espacial está relacionada ao tamanho do pixel do terreno, a qual o sensor é capaz de registrar a radiação, geralmente com formato quadrangular é mensurada por seu comprimento, quanto menor esta área, maior a resolução espacial. A resolução espectral refere-se à largura da faixa do espectro eletromagnético a qual o sensor é sensível, faixas mais largas conferem uma resolução espectral menor. E a resolução radiométrica é a capacidade do sensor de distinguir pequenas variações na intensidade da radiação observada.

Quanto ao funcionamento, o DPI/INPE ([2006]) classifica os sensores em função da fonte de radiação e em função do tipo de produto que produz.

Em função da fonte de radiação podem ser:

- *Passivos*: não possuem fonte própria de radiação. Medem radiação solar refletida ou radiação emitida pelos alvos. Ex.: Sistemas fotográficos e eletro-óticos;
- *Ativos*: possuem sua própria fonte de radiação eletromagnética, trabalhando em faixas restritas do espectro. Ex.: Radares;

Em função do tipo de produto são classificados como:

- *Não-imageadores*: não geram imagem da superfície sensorizada.  
Ex.: Espectrorradiômetros (assinatura espectral) e radiômetros (saída em dígitos ou gráficos). Essenciais para aquisição de informações precisas sobre o comportamento espectral dos objetos;

- *Imageadores*: obtém-se uma imagem da superfície observada como resultado. Fornecem informações sobre a variação espacial da resposta espectral da superfície observada;

Além disso, conforme Novo e Ponzoni (2001), os sensores imageadores podem grosseiramente classificados em três tipos:

- *Imageadores de quadro*: são os mais antigos e se desenvolveram a partir dos sistemas de televisão. Tais sensores possuem um sistema ótico grande angular que focaliza toda a energia proveniente da cena sobre um tubo fotossensível. A imagem é formada instantaneamente sobre o tubo fotossensível e então “varrida” por um feixe de elétrons que a converte em sinal.
- *Imageadores de matriz linear de sensores (pushbroom)*: Estes utilizam sistemas óticos de grande campo de visada, o que permite que toda a faixa perpendicular ao deslocamento da plataforma seja imageada instantaneamente. À medida que a plataforma se move ao longo da órbita, linhas sucessivas são imageadas pela matriz linear e amostradas por um multiplexador responsável pela transmissão dos sinais.
- *Varredores mecânicos*: compõem-se basicamente de um espelho giratório que “varre” a superfície imageada e focaliza a energia proveniente do solo sobre um sensor pontual. Assim sendo, a imagem é construída ponto a ponto a cada variação instantânea da posição do espelho.

Os dados do sensoriamento remoto normalmente possuem o formato de imagens digitais, que podem ser definidas, segundo o DPI/INPE ([2006]), como uma função bidimensional da radiação refletida ou emitida por uma cena, na forma  $I(x, y)$ ; onde os valores  $I$  representam a intensidade da imagem a cada coordenada espacial  $(x, y)$ . Essa intensidade, geralmente, é representada por um valor inteiro, não-negativo e finito.

Estas imagens podem ser adquiridas em diferentes regiões do espectro eletromagnético, p. ex. a região do visível, termal, micro-ondas etc.; com canais de diferentes resoluções espectrais que vão de poucos nanômetros, como é o caso dos sensores hiperespectrais, até sistemas pancromáticos que integram radiação em todo espectro visível.

### 2.8.1 Sistemas imageadores eletro-óticos

Segundo Novo e Ponzoni (2001), estes sistemas possuem sensores capazes de transformar a radiação eletromagnética em sinais elétricos, que podem ser transmitidos a uma estação

remota. São constituídos por um sistema de lentes e espelhos com a função de concentrar energia sobre seus sensores.

Sua resolução espectral depende da sensibilidade espectral de seus sensores, geralmente constituídos por ligas metálicas que têm a propriedade de “traduzir” determinados valores de intensidade de radiação em pulsos elétricos; enquanto que a resolução espacial é função do tamanho do elemento de resolução da cena, corriqueiramente denominado de pixel (Picture Element).

#### 2.8.1.1 Sistema TM

O *Thematic Mapper* (TM) foi um avançado sensor multiespectral a bordo do satélite Landsat 5, projetado para alcançar maior resolução de imagem, separação espectral mais precisa, maior fidelidade geométrica e maior precisão e resolução radiométrica do que o sensor MSS (*Multispectral Scanner*). (NASA, 2019)

Suas principais características são apresentadas no quadro 2.

**Quadro 2** Características do Sensor TM

<b>Características</b>	<b>Dado</b>	<b>Características</b>	<b>Dado</b>
<b>Bandas Espectrais</b>	0,45-0,52 $\mu$ m (B)	<b>Resolução Espacial</b>	30 m/120m (TIR)
	0,52-0,60 $\mu$ m (G)		
	0,63-0,69 $\mu$ m (R)	<b>Largura da Faixa</b>	185 km
	0,76-0,90 $\mu$ m (NIR)		
	1,55-1,75 $\mu$ m (SWIR)	<b>Revisita</b>	16 dias
	10,41-12,50 $\mu$ m (TIR)	<b>Quantização</b>	8 bits
	2,08-2,35 $\mu$ m (SWIR)		

Fonte: NASA, 2019

#### 2.8.1.2 Sistema ETM+

O *Enhanced Thematic Mapper Plus* (ETM+) é o instrumento a bordo do LANDSAT 7, ele replica as capacidades do seu antecessor, o TM, além de incluir recursos adicionais que o fizeram mais versátil e eficiente para estudos de mudanças globais, monitoramento de cobertura terrestre e mapeamento de grandes áreas. Estes recursos são: uma banda pancromática com 15m de resolução espacial; um canal infravermelho termal com 60m de resolução espacial e um gravador de dados a bordo. (NASA, 2019)

Suas principais características são apresentadas no quadro 3.

**Quadro 3** Características do Sensor ETM+

Características	Dado	Características	Dado
<b>Bandas Espectrais</b>	0,45-0,52 $\mu$ m (B)	<b>Resolução Espacial</b>	15 m (Pan)/30m/60m (TIR)
	0,52-0,60 $\mu$ m (G)		
	0,63-0,69 $\mu$ m (R)	<b>Largura da Faixa</b>	185 km
	0,77-0,90 $\mu$ m (NIR)		
	1,55-1,75 $\mu$ m (SWIR)	<b>Revisita</b>	16 dias
	10,40-12,50 $\mu$ m (TIR)		
	2,08-2,35 $\mu$ m (SWIR)	<b>Quantização</b>	8 bits
0,52-0,90 $\mu$ m (Pan)			

Fonte: USGS, 2017

### 2.8.1.3 Sistema OLI

O *Operational Land Imager* (OLI), construído pela *Ball Aerospace & Technologies Corporation*, está a bordo do Landsat 8 e mede a radiação nas porções visível, infravermelho próximo e infravermelho de ondas curtas do espectro. Suas imagens têm resoluções espaciais de 15 m na banda pancromática e de 30 m nas bandas multiespectrais ao longo de uma faixa de 185 km de largura, cobrindo amplas áreas da paisagem da Terra, fornecendo resolução suficiente para distinguir recursos como centros urbanos, fazendas, florestas e outros usos da terra. Toda a Terra será vista uma vez a cada 16 dias devido à órbita quase polar do LANDSAT 8. O OLI usa longas matrizes de detectores, com mais de 7.000 detectores por banda espectral, alinhados em todo o seu plano focal para ver através da faixa. Este design de "pushbroom" resulta em um instrumento mais sensível, fornecendo informações aprimoradas da superfície da terra com menos partes móveis. Com uma relação sinal-ruído melhorada em comparação com os instrumentos Landsat anteriores, o OLI é mais confiável e proporciona um melhor desempenho. (NASA, 2019)

O OLI captura dados com precisão radiométrica aprimorada em uma faixa dinâmica de 12 bits, o que melhora a relação sinal-ruído geral. Isso se traduz em 4096 níveis de cinza em potencial, em comparação com apenas 256 níveis de cinza nos instrumentos LANDSAT 1-7 de 8 bits. O melhor desempenho do sinal-ruído permite uma melhor caracterização do estado e condição da cobertura da terra. Os dados de 12 bits são dimensionados para inteiros de 16 bits e entregues nos produtos de dados Nível 1. Os produtos são dimensionados para 55000 níveis de cinza e podem ser redimensionados para a refletância e/ou radiância do TOA (*Top of Atmosphere*) usando coeficientes de redimensionamento radiométricos fornecidos no arquivo de metadados do produto. (USGS, 2017)

Suas principais características são apresentadas no quadro 4.

**Quadro 4** - Características do Sensor OLI.

Características	Dado	Características	Dado
<b>Bandas Espectrais</b>	0,43-0,45 $\mu$ m (DBV)	<b>Resolução Espacial</b>	15 m (Pan)/30m
	0,45-0,51 $\mu$ m (B)		
	0,53-0,59 $\mu$ m (G)	<b>Largura da Faixa</b>	185 km
	0,64-0,67 $\mu$ m (R)		
	0,85-0,88 $\mu$ m (NIR)	<b>Revisita</b>	16 dias
	1,57-1,65 $\mu$ m (SWIR)		
	2,11-2,29 $\mu$ m (SWIR)	<b>Quantização</b>	16 bits
	0,50-0,68 $\mu$ m (Pan)		
1,36-1,38 $\mu$ m (Cirrus)			

Fonte: USGS, 2017.

#### 2.8.1.4 Sistema MUX

O sistema MUX (Câmera Multiespectral Regular) está a bordo dos satélites CBERS-3 e 4. Foi construído no Brasil com o intuito de manter a continuidade dos imageamentos feitos pelos três CBERS anteriores (1, 2 e 2B). É o sensor que assegura o recobrimento global pelo CBERS numa resolução espacial padrão a cada 26 dias. Com um campo de visada estreito, há pouca alteração no tamanho do pixel nas bordas da imagem. As aplicações desse sistema são as já desenvolvidas para os satélites que possuem câmeras nessa classe de resolução (20 m) (INPE, 2019). Suas principais características são apresentadas no Quadro 5.

**Quadro 5** - Características do Sensor MUX.

Características	Dado	Características	Dado
<b>Bandas Espectrais</b>	0,45-0,52 $\mu$ m (B)	<b>Resolução Espacial</b>	20 m
	0,52-0,59 $\mu$ m (G)	<b>Largura da Faixa</b>	120 km
	0,63-0,69 $\mu$ m (R)	<b>Revisita</b>	26 dias
	0,77-0,89 $\mu$ m (NIR)	<b>Quantização</b>	8 bits

Fonte: INPE, 2019

#### 2.8.1.5 Sistema IRS

O IRS (Imageador Multiespectral e Termal), também a bordo do CBERS-3 e 4, é um sensor de varredura mecânica e teve uma significativa melhoria em relação ao seu predecessor presente nos CBERS-1 e 2. Sua resolução espacial passa para 40 metros na banda pancromática e do SWIR (*Short Wave Infra Red*) e para 80 m na banda termal. Há uma expectativa de que a disponibilidade de bandas na região do SWIR venha trazer um grande interesse para uso deste sensor. Agora, que está com uma resolução espacial muito melhorada e próxima daquela do TM, é esperado que as aplicações típicas do LANDSAT passem a utilizar também imagens do IRS. Apesar de estarem em sensores diferentes, é plausível que se busquem soluções para a fusão de imagens do IRS com as da MUX ou mesmo da PAN, com a intenção de aproveitar a

alta capacidade discriminativa das bandas do SWIR presentes no IRS. Com seu campo de visada de 120 km, acompanha o ciclo de 26 dias para fazer o recobrimento global (INPE, 2019). Suas principais características são apresentadas no Quadro 6.

**Quadro 6 - Características do Sensor IRS.**

<b>Características</b>	<b>Dado</b>	<b>Características</b>	<b>Dado</b>
<b>Bandas Espectrais</b>	0,50-0,90 $\mu$ m (Pan)	<b>Resolução Espacial</b>	40 m/80m (TIR)
	1,55-1,75 $\mu$ m (SWIR)	<b>Largura da Faixa</b>	120 km
	2,08-2,35 $\mu$ m (SWIR)	<b>Revisita</b>	26 dias
	10,40-12,50 $\mu$ m (TIR)	<b>Quantização</b>	8 bits

Fonte: INPE, 2019.

## 2.8.2 Sistemas RADAR

Segundo Novo e Ponzoni (2001), a palavra RADAR é um acrônimo da expressão “*Radio Detection ad Ranging*”. Um RADAR executa três funções básicas: transmite um pulso de micro-ondas em direção a um alvo, que o classifica como um sensor ativo, recebe a porção refletida do pulso transmitido após este haver interagido com o alvo (a porção refletida recebe o nome de energia retro-espalhada), registra a potência, a variação temporal e o tempo de retorno do pulso retro-espalhado.

Os sistemas RADAR medem a distância e a potência recebida de pulsos emitidos por uma antena. Esses pulsos se caracterizam por ondas eletromagnéticas com comprimentos que variam de 1 m a 1 mm (ou frequências entre 0,3 GHz e 300 GHz). Do ponto de vista das atividades de sensoriamento remoto, o fato de um sistema RADAR operar nesses comprimentos de onda, traz algumas vantagens: as micro-ondas são passíveis de penetrarem nuvens, chuva, fumaça e neblina. Em comprimentos de ondas menores (6 cm) chuvas e nuvens espessas podem afetar o sinal de radar; as micro-ondas são sensíveis a propriedades distintas daquelas que afetam a radiação na região do visível e infravermelho. As imagens de radar apresentam informações que são qualitativamente e quantitativamente diferentes daquelas registradas em imagens ópticas, sendo, portanto, complementares àquelas.

Ao contrário dos sistemas ópticos, os sensores de micro-ondas operam em bandas específicas para as quais foram associadas letras. As bandas em uso atualmente encontram-se resumidas no Quadro 7.

**Quadro 7** - Bandas de operação dos sistemas RADAR e aplicações.

Banda	Comprimento de onda (cm)	Frequência (GHz)	Principal Aplicação
X	2,4-3,8	8,0-12,5	Reconhecimento militar, reconhecimento de terreno.
C	3,8-7,5	4,0-8,0	Monitoramento de gelo e aplicações oceanográficas.
S	7,5-15,0	2,0-4,0	Reconhecimento de terreno
L	15,0-30,0	1,0-2,0	Mapeamento de cobertura vegetal
P	75,0-133,0	0,225-0,400	Mapeamento de cobertura vegetal

Fonte: Novo e Ponzoni (2001)

Existem basicamente dois tipos de sistemas RADAR. O mais antigo, e já fora de funcionamento, Radar de Abertura Real (RAR) e os SAR (*Synthetic Aperture Radar*)

O SAR representa um modo engenhoso de superar o problema do tamanho físico da antena a partir da “síntese” de uma antena virtual a partir do registro acumulado dos sinais de retorno de cada objeto da antena durante o período em que ela se desloca sobre uma dada região do terreno. Uma vez que o pulso tenha passado sobre um ponto do terreno, toda a informação de fase sobre aquele ponto é armazenada em uma matriz bidimensional (alcance e azimute). Todas as “histórias” de fase de todos os pontos da imagem são combinadas numa série temporal que forma o dado sintético. Através de um processamento complexo, esta “assinatura de fase” de cada ponto é transformada em informação de azimute e alcance.

Atualmente, os sistemas de RADAR possuem a chamada antena de abertura sintética o que permite que estes sistemas possam ser colocados em plataformas orbitais produzindo resoluções espaciais da ordem de poucos metros.

#### 2.8.2.1 Sistema PALSAR

O PALSAR (*Phased Array type L-band Synthetic Aperture Radar*) do ALOS (*Advanced Land Observing Satellite*) foi um sensor de micro-ondas ativo que usava a frequência da banda L para obter dados da superfície, livre de nuvens, tanto de dia, quanto a noite. O desenvolvimento do PALSAR foi um projeto conjunto entre a *Japan Aerospace Exploration Agency* (JAXA) e a *Japan Resources Observation System* (JAROS). (JAXA, [1997?])

O PALSAR foi um dos três instrumentos do *Advanced Land Observing Satellite-1* (ALOS), ativo durante o período de 2006 a 2011 com dois modos de feixe fino: polarização única (FBS) e polarização dupla (FBD), além de polarização quádrupla, também conhecida como modo polarimétrico (PLR). E ainda possuía o feixe largo ScanSAR (WB1, WB2), que operava com uma perda considerável de resolução. (*Alaska Satellite Facility*, [2019])

Suas principais características são apresentadas no quadro 8.

**Quadro 8 - Características do sensor PALSAR.**

	Resolução Fina		ScanSAR	Polarimétrico
<b>Modo de Feixe</b>	FBS	FBD	WB1/WB2	PLR
<b>Frequência Central</b>	Banda-L (1.27 GHz)			
<b>Polarização</b>	HH ou VV	HH+HV ou VV+VH	HH ou VV	HH+HV+VV+VH
<b>Resolução Espacial</b>	10 m	20m	100m	30m
<b>Largura da Faixa</b>	70km	70km	250-350km	30km
<b>Quantização</b>	5 bits	5 bits	5 bits	3 ou 5 bits

Fonte: Alaska Satellite Facility [2019] e JAXA, [1997]

### 2.8.3 Classificação de imagens

Segundo Meneses e Sano (2011), classificação de imagens é o processo de associar os pixels da imagem a um número finito de classes individuais que representam os objetos do mundo real, com base nos seus valores digitais, ou seja, se um pixel satisfaz um determinado conjunto de critérios, normalmente avaliados por medidas estatísticas, o pixel é atribuído à classe que melhor satisfaz os critérios estabelecidos. O resultado final de um processo de classificação é uma imagem digital que constitui um mapa de "pixels" classificados, representados por símbolos, gráficos ou cores.

A classificação de imagens deve ser vista como um processo estatístico e probabilístico que tenta ao máximo aproximar o mapa digital à realidade. O resultado da classificação deve ser avaliado com base no desempenho do classificador proposto e validado por critérios numéricos para estimar a precisão e a sua acuracidade.

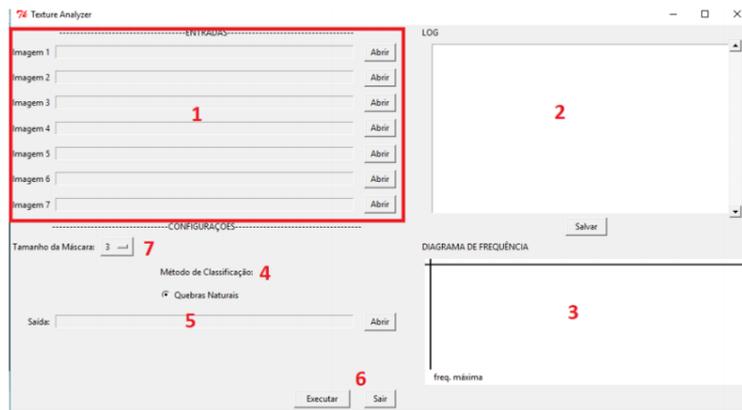
Os diversos métodos de classificação podem ser divididos segundo vários critérios, no entanto a forma mais comum e difundida, que acabou por ser adotada por softwares de processamento de imagens separa os classificadores em supervisionados e não-supervisionados.

São classificadores não-supervisionados, aqueles que requerem pouca ou nenhuma participação do analista no processo de classificação da imagem, sendo de responsabilidade do analista, depois da classificação, associar um significado às classes resultantes. O resultado da classificação é útil somente se as classes podem ser interpretadas apropriadamente. Estes classificadores apenas permitem ao analista especificar alguns parâmetros que o computador irá usar para descobrir padrões que são inerentes aos dados. Esses padrões não necessariamente correspondem às características reais da cena em termos de classes de alvos que reconhecemos. Eles são simplesmente agrupamentos de pixels com características espectrais similares (*clusters*). Mas, em alguns casos, pode ser mais importante identificar grupos de pixels com características espectrais similares, do que separar os pixels em classes reconhecidas.

## 2.9 TEXTURE ANALYZER (2018)

Em 2018 foi desenvolvido o software Texture Analyzer (Figura 2) com a finalidade de analisar as diferentes texturas presentes no reservatório de Itaparica, situado no estado de Pernambuco. Este foi desenvolvido na linguagem Python em sua versão 2.7.10 fora do ambiente QGIS, o que dificultou seu uso e divulgação.

**Figura 2** Software Texture Analyzer



Fonte: Paixão Junior e Tavares Junior, 2018.

Onde: 1) Campo de entrada de imagens; 2) Tela de Log; 3) Diagrama de frequência; 4) Método de classificação; 5) Diretório de saída; 6) Botões Executar e Sair; 7) Tamanho da máscara.

Segundo Paixão Junior e Tavares Junior (2018), para a definição de textura buscou-se meios de caracterizar uma amostra de ND's (um pixel e seus vizinhos<sup>2</sup>) de forma única. Para tal foram escolhidas três medidas que caracterizavam a amostra: a mediana, o desvio mediano e o desvio-padrão. E então o valor de textura na imagem  $i$  foi definido como:

$$T_i(x, y) = \sqrt{Md^2 + DMd^2 + \sigma^2} \quad (12)$$

Onde  $(x, y)$  é a posição do pixel,  $Md$  a mediana,  $DMd$  o desvio mediano e  $\sigma$  o desvio padrão da amostra de ND's.

E a partir disso foi definida a Textura Espectral como:

$$T_s(x, y) = \sqrt{\sum_{i=1}^n (T_i(x, y))^2} \quad (13)$$

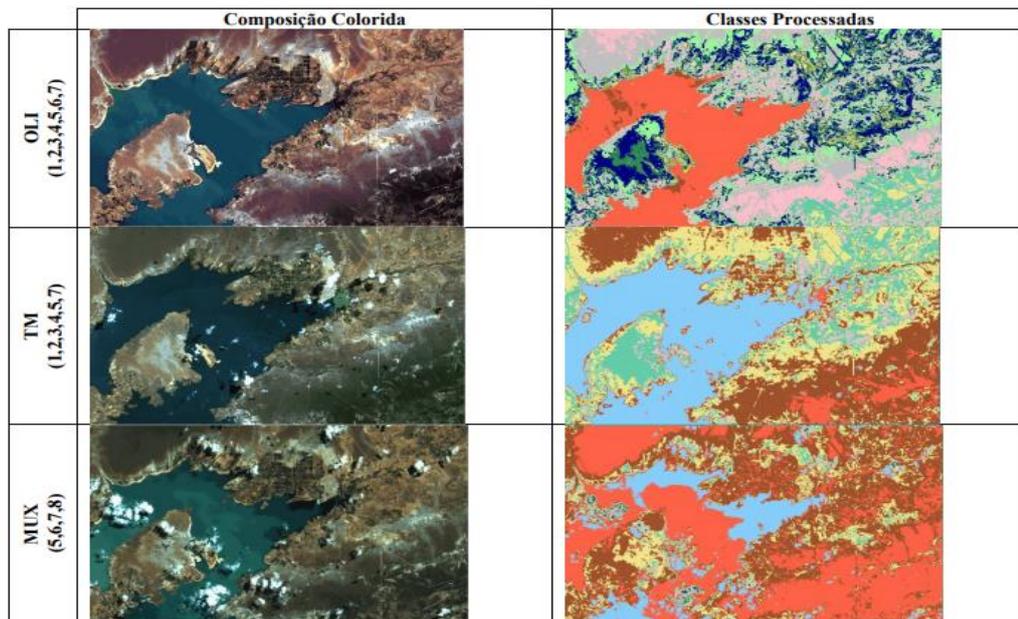
<sup>2</sup>Os vizinhos são definidos pelo tamanho da máscara inserida no software.

Onde n é o número de imagens inseridas no software.

Após o cálculo da textura espectral de cada pixel, os valores são ordenados, o número de classes definido pelo método de Sturges e classificados pelo método de quebras naturais, e então os resultados são salvos em uma pasta pré-definida.

Como resultado é apresentado uma imagem RGB onde cada cor representa uma classe de textura, além de uma imagem binária para cada classe discriminada (Figura 3).

**Figura 3** - Resultados obtidos pelo software.



Fonte: Paixão Junior e Tavares Junior, 2018.

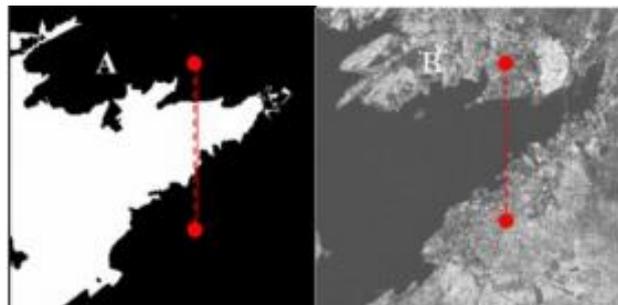
Na Figura 3 estão apresentados os resultados gerados para região em diversos sensores (OLI, TM e MUX). Os números ao lado do nome do sensor são as bandas utilizadas no cálculo da textura espectral. As imagens parecem confusas, pois há um grande número de texturas existentes, além de não ser possível escolher as cores atribuídas a cada classe no software, uma das deficiências deste. Outra deficiência é o resultado não ser georreferenciado.

Na Figura 4 são dispostas imagens binárias da classe correspondente a corpos d'água resultantes do processamento do software, observa-se que o resultado difere de sensor pra sensor, devido à data de obtenção da imagem e as características próprias de cada um. No caso do MUX, a imagem apresenta significante diferença, pois há outros materiais suspensos na água, sendo este tipo de massa d'água classificado como outro tipo de textura, na Figura 3 apresentado em vermelho.

**Figura 4 -** Imagens binárias resultantes.

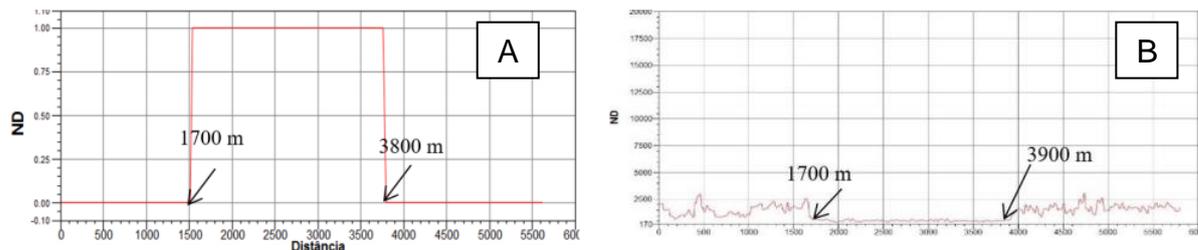
Fonte: Paixão Junior e Tavares Junior, 2018

Em 2019, Paixão Junior e Tavares Junior realizaram experimentos no *Texture Analyzer* a fim de avaliar o seu potencial de mapeamento. Para tal utilizou-se os resultados obtidos para o sensor OLI e comparou a uma imagem PALSAR através de um perfil (Figura 4).

**Figura 5 -** Localização dos perfis utilizados: A) Imagem Binária; B) Imagem PALSAR.

Fonte: Paixão Junior e Tavares Junior, 2019

Os resultados de mapeamento de texturas usando o *Texture Analyzer* demonstraram que na transição água e solo, há pontos de forte variação no ND, entre as distâncias de 1700 m e 3800 m, como mostra o perfil da imagem binária (figura 6.A) e 1700 m e 3900 m na banda da imagem PALSAR (Figura 6.B).

**Figura 6 -** Perfis utilizados. A) Imagem Binária; B) Imagem PALSAR.

Fonte: Paixão Junior e Tavares Junior, 2019.

A borda VV (Figura 6.B) coincidiu com a borda da imagem binarizada (Figura 6.A), revelando assim o potencial para mapeamento de borda de reservatórios hídricos de grandes extensões.

### **3 METODOLOGIA**

A metodologia foi dividida em duas etapas. Na primeira etapa foi executada uma adaptação do *Texture Analyzer* como um complemento QGIS3.

#### **3.1 DESENVOLVIMENTO DO COMPLEMENTO**

Aqui são descritos os materiais utilizados e os procedimentos efetuados para desenvolvimento e operação do complemento no ambiente QGIS3.

##### **3.1.1 Materiais utilizados nesta etapa**

- PC Desktop com AMD Quad-Core 3.80 GHz, 4 GB de memória RAM e Windows 10 Pro 64 bits.
- Software Paint
- Software QGIS 3.6.3
- Complementos Plugin Builder 3 e Plugin Reload
- OSGeo4W Shell
- Qt Designer
- PyCharm Community

##### **3.1.2 Procedimentos**

Os procedimentos desenvolvidos podem ser divididos em duas etapas: a primeira refere-se à preparação do ambiente QGIS3, obtenção da estrutura de arquivos e instalação do complemento em seu estado inicial (seção 3.1.2.1 a seção 3.1.2.3). A segunda consiste na customização da interface gráfica do complemento (seções 3.1.2.4 e 3.1.2.5). E a terceira é a etapa de programação e inserção dos algoritmos de processamento nos arquivos do complemento (seção 3.1.2.6).

### 3.1.2.1 Aquisição dos complementos necessários

Inicialmente, instalou-se o software QGIS 3.6.3 *Noosa*, versão mais recente até a data de elaboração deste trabalho, com suas configurações padrão. Ao iniciar o software, procura-se o menu “*Complementos*” e então “ *Gerenciar e Instalar Complementos*”.

Dois complementos são necessários para o desenvolvimento, o *Plugin Builder 3* e o *Plugin Reload*. O *Plugin Builder 3* fornece toda estrutura de arquivos necessária para a correta ambientação do complemento no QGIS e o *Plugin Reload* permite recarregar todos os arquivos de um complemento sem a necessidade de reiniciar o ambiente QGIS, o que acelera o desenvolvimento.

Ao clicar em “ *Gerenciar e Instalar Complementos*”, uma janela é aberta com quatro abas do lado esquerdo (*Tudo*, *Instalados*, *Install from ZIP* e *Opções*), uma barra de busca e se houver conexão com a internet, uma lista com todos os complementos disponíveis, caso não, é apresentada uma lista com os complementos previamente instalados. No entanto, para instalar novos complementos é necessário ter conexão com a internet ou obtê-los através de um arquivo ZIP, esta forma de instalação será abordada posteriormente.

Na barra de busca, digitou-se o nome dos complementos necessários e selecionou-se o complemento desejado e então se clicou em “*Instalar Complemento*”. Os complementos foram instalados no diretório “*python/plugins*” do diretório padrão do perfil ativo (default). Porém para encontrar o complemento *Plugin Reloader* foi necessário acessar a aba “*Opções*” e ativar a opção “*Mostrar também os complementos experimentais*”.

Dois ícones surgiram na barra de ferramentas do QGIS (Figura 7).

**Figura 7** - Ícones do *Plugin Builder* e *Plugin Reload*.



Fonte: O autor, 2019

### 3.1.2.2 Execução do Plugin Builder.

Ao clicar no ícone  a interface do *Plugin Builder* foi apresentada conforme a Figura 8. Notaram-se diversos campos que devem ser preenchidos para a estruturação adequada do complemento. Todos estes campos estão listados no quadro 9 com uma breve explicação e os valores inseridos para o *Texture Analyzer*.

**Figura 8** - Janela Inicial do *Plugin Builder*.

Fonte: O autor, 2019

**Quadro 9** - Campos requeridos pelo *Plugin Builder*.

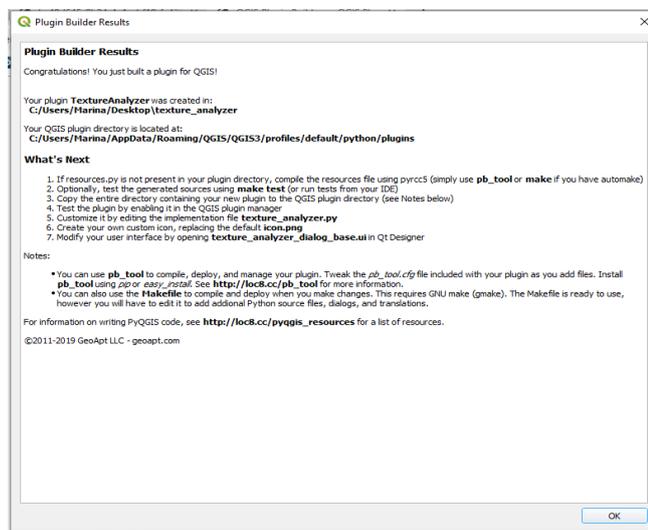
<b>Campo</b>	<b>Descrição</b>	<b>Valor inserido</b>
<b>Classname</b>	Nome utilizado para criar a classe Python do complemento.	Texture Analyzer
<b>Plugin name</b>	Título que o gerenciador de complementos do QGIS exibirá.	Texture Analyzer
<b>Description</b>	Descrição em uma linha que será exibido pelo gerenciador de complementos.	Procura identificar as texturas presentes em um conjunto de imagens raster.
<b>Module name</b>	Nome utilizado para criar o módulo Python do complemento.	texture_analyzer
<b>Versionnumber</b>	Número da versão do complemento.	1.0
<b>Minimum QGIS version</b>	Versão mínima do QGIS necessária para o complemento funcionar.	3.0
<b>Author/Company</b>	Nome do autor ou nome da empresa. Esta informação é usada para escrever a declaração de direitos autorais nos arquivos de origem do complemento.	Givaldo Cesar
<b>Email adress</b>	E-mail para contato.	givaldocesar@live.com
<b>About</b>	Descrição das funções exercidas pelo complemento.	Este plugin procura identificar as texturas presentes na área de estudo utilizando os números digitais de um conjunto de imagens de um sensor.  Desenvolvido como TCC por Givaldo Cesar sob a orientação de João Tavares para o curso de Engenharia Cartográfica e de Agrimensura da Universidade Federal de Pernambuco no ano de 2019.
<b>Template</b>	O tipo de complemento que se deseja criar.	Tool button with dialog
<b>Text for item menu</b>	Texto que irá aparecer no menu. Geralmente não se usa o mesmo nome do complemento.	Iniciar
<b>Menu</b>	O menu apropriado para qual o complemento será adicionado	Raster
<b>Internationalization</b>	Um esboço para adicionar textos traduzidos ao complemento.	Não marcada
<b>Help</b>	Cria um template para o desenvolvimento de uma ajuda em HTML.	<b>Não marcada</b>
<i>Continua</i>		

<b>Unit tests</b>	Gera um conjunto básico de unidades testes para o complemento.	Não marcada
<b>Helper scripts</b>	Adiciona um script auxiliar para publicação do complemento e scripts adicionais para internacionalização e teste.	Não marcada
<b>Makefile</b>	Adiciona um Makefile para construção do complemento com o GNU make. <sup>3</sup>	Não marcada
<b>Pb_tool</b>	Gera uma configuração para o pb_tool <sup>4</sup> .	Não marcada
<b>Bug Tracker</b>	URL para apresentação de erros e problemas do complemento.	<a href="https://github.com/givaldocesar/textanalyzer">https://github.com/givaldocesar/textanalyzer</a>
<b>Repository</b>	URL do repositório do código fonte do complemento.	<a href="https://github.com/givaldocesar/textanalyzer">https://github.com/givaldocesar/textanalyzer</a>
<b>Home Page</b>	URL da home page do complemento.	<a href="https://github.com/givaldocesar/textanalyzer">https://github.com/givaldocesar/textanalyzer</a>
<b>Tags</b>	Lista separada por vírgulas de palavras-chave que descrevem a função do complemento.	python, texturas, análise, sensoriamento, remoto, raster
<b>Flag the plugin as experimental</b>	Marcar esta opção se complemento for considerado experimental, o que significa que está incompleto ou pode causar consequências indesejadas.	Não marcada

Fonte: Sherman [2011] e o autor (2019)

Após o preenchimento de todos os campos, escolheu-se o diretório de saída e clicou-se em “Generate”. Então surgiu uma janela conforme a Figura 9.

**Figura 9** - Janela resultante do Plugin Builder.



Fonte: O autor, 2019

Esta janela indica onde o complemento foi salvo, qual o diretório padrão dos complementos Python e informações sobre como publicar e quais arquivos devem ser customizados.

<sup>3</sup> Mais informações em: <http://g-sherman.github.io/Qgis-Plugin-Builder/#using-the-makefile>

<sup>4</sup> Mais informações em: <http://g-sherman.github.io/Qgis-Plugin-Builder/#using-pb-tool>

### 3.1.2.3 Instalação do complemento gerado no QGIS.

Antes de instalar o complemento, gerado no *Plugin Builder*, no QGIS foi necessário compilar o arquivo “*resources.qrc*”, presente no diretório indicado anteriormente, com o comando “*pyrcc5*” do software OSGeoW4 Shell. O OSGeoW4 Shell é um aplicativo de linha de comando que é instalado junto com o software QGIS.

A compilação é necessária para que as imagens presentes na interface do complemento, p. ex. o ícone, sejam exibidas corretamente. Para tal, os seguintes passos foram executados:

1. Executou-se o OSGeoW4 Shell.
2. Executou-se o comando “*py3\_env*”.
3. Executou-se o comando “*qt5\_env*”.
4. Executou-se o comando “*cd ~diretório~*”. Onde *~diretório~* é o caminho para o diretório em que foi salvo o complemento gerado pelo Plugin Builder.
5. Executou-se o comando “*pyrcc5 resources.qrc -o resources.py*”.

O resultado é a criação do arquivo “*resources.py*” no diretório do complemento.

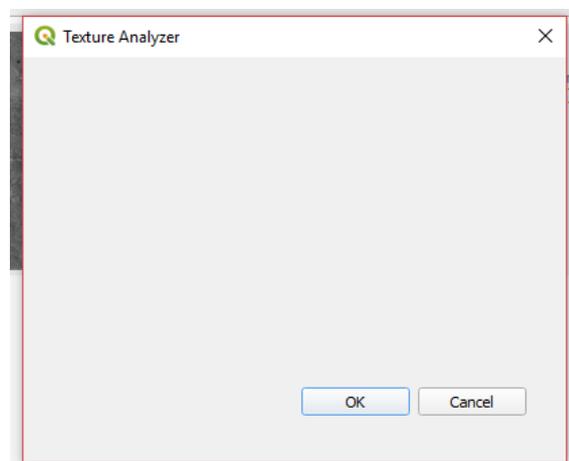
Os passos 2 e 3 são uma preparação do ambiente OSGeoW4 para a correta execução do comando “*pyrcc5*”.

Após a compilação, compactou-se o diretório em formato ZIP.

Para a instalação no QGIS, clicou-se na opção “ Gerenciar e Instalar Complementos”, na aba “*Install from ZIP*”, em ..., selecionou-se o arquivo ZIP e então se clicou em “*Instalar Complemento*”. O ícone  surgiu na barra de ferramentas.

Ao clicar no ícone, apareceu uma janela conforme a Figura 10.

**Figura 10** - Janela inicial *Texture Analyzer*.



Nota-se que é apenas uma janela base para a construção do complemento. É necessário instalá-lo mesmo sem suas funções implementadas, pois assim é possível utilizar o *Plugin Reloader* para testar alterações feitas no código-fonte, praticamente, em tempo real.

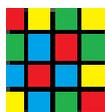
Para utilizar o Plugin Reloader, clicou-se no ícone , selecionou-se o *Texture Analyzer* e então clicou-se em “Ok”. A partir disso, todas as vezes que se clicou novamente no ícone, o código do *Texture Analyzer* era atualizado dentro do ambiente QGIS.

A partir daqui as alterações foram feitas no diretório do *Texture Analyzer* gerado na instalação do complemento no QGIS. O diretório de instalação de complementos, como dito anteriormente, é o “*python/plugins*”, este diretório pode ser encontrado ao clicar em “*Open Active Profile Folder*”, opção presente no menu “*Configurações*” na aba “*Perfis do usuário*”

#### 3.1.2.4 Customização do ícone

Para customizar o ícone do complemento foi necessário editar o arquivo “*icon.png*” com o software *Paint*, presente no OS Windows 10. No entanto, qualquer ferramenta de edição de imagens pode ser utilizada. O resultado da edição é apresentado na figura 11.

**Figura 11** - Novo ícone do *Texture Analyzer*.



Fonte: O autor, 2019

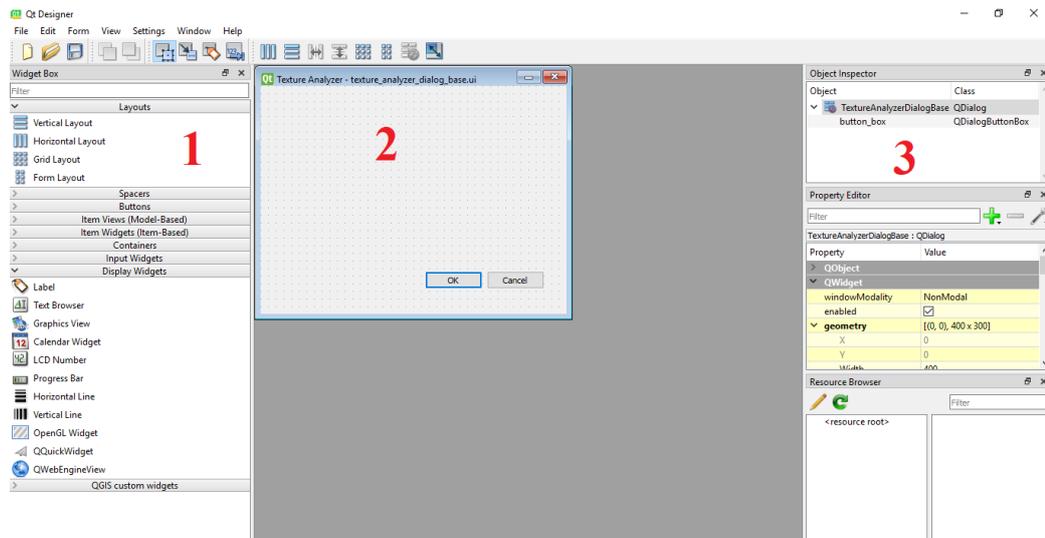
Após a alteração foi necessário recompilar o arquivo “*resources.qrc*”.

#### 3.1.2.5 Customização da interface do usuário

Para a edição da interface do usuário foi necessário alterar o arquivo “*texture\_analyzer\_dialog\_base.ui*” com a utilização do software *Qt Designer*, que é um software também instalado junto com o QGIS e segundo a *The Qt Company* (2019), é uma ferramenta Qt para projetar e construir interfaces gráficas de usuário com Qt Widgets, podendo-se compor e personalizar janelas ou diálogos de uma maneira simples e testá-los usando diferentes estilos e resoluções.

Inicialmente, executa-se o *Qt Designer* e na sua tela inicial clica-se em “*Open...*”, então se seleciona o arquivo “*texture\_analyzer\_dialog\_base.ui*”. Daí, uma janela como a figura 12 foi apresentada.

**Figura 12 - Tela Inicial Qt Designer.**



Fonte: O autor, 2019

Onde:

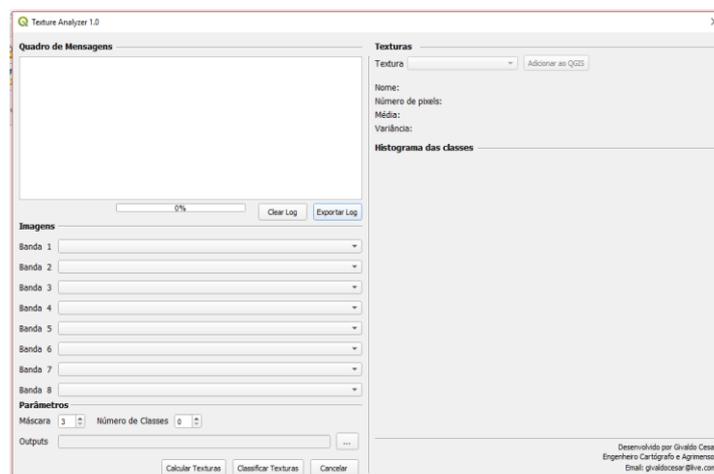
1. É onde estão disponibilizados todos os widgets necessários para organização do layout e interação com o usuário.
2. Exibe a janela que está sendo editada. Foi necessário arrastar para cá, os widgets necessários para o layout do *TextureAnalyzer*.
3. Lista os widgets utilizados, onde é possível alterar suas configurações.

Para a construção da interface foi necessário, apenas, clicar no widget e arrastá-lo para a posição desejada, então clicar em seu nome na área 3 e editar suas configurações, para que este seja exibido de forma adequada. Devido à grande quantidade de widgets usados, ocultou-se a informação sobre estes, no entanto o arquivo “*texture\_analyzer\_dialog\_base.ui*” será disponibilizado no link <https://github.com/givaldocesar/textanalyzer>.

Como houve inserção de figuras na interface, foi necessário editar o arquivo “*resources.qrc*”, inserindo o caminho para estas figuras e recompilá-lo. O código do arquivo “*resources.qrc*” será disponibilizado no link <https://github.com/givaldocesar/textanalyzer>.

O resultado da edição é apresentado na Figura 13.

**Figura 13** - Tela editada do *Texture Analyzer*.



Fonte: O autor, 2019

Notou-se que as figuras e o ícone não foram exibidos como se desejava, pois ainda é necessário alterar os arquivos “*texture\_analyzer.py*” e “*texture\_analyzer\_dialog.py*”. A interface apesar de completa não tem nenhuma função atribuída a seus widgets.

### 3.1.2.6 Edição dos arquivos Python

Para editar tais arquivos utilizou-se o software *PyCharm Community Edition*. O *PyCharm* é um IDE<sup>5</sup> dedicado para Python e Django que fornece uma ampla gama de ferramentas essenciais para desenvolvedores, totalmente integradas para criar um ambiente conveniente para o desenvolvimento produtivo. (JetBrains, 2019)

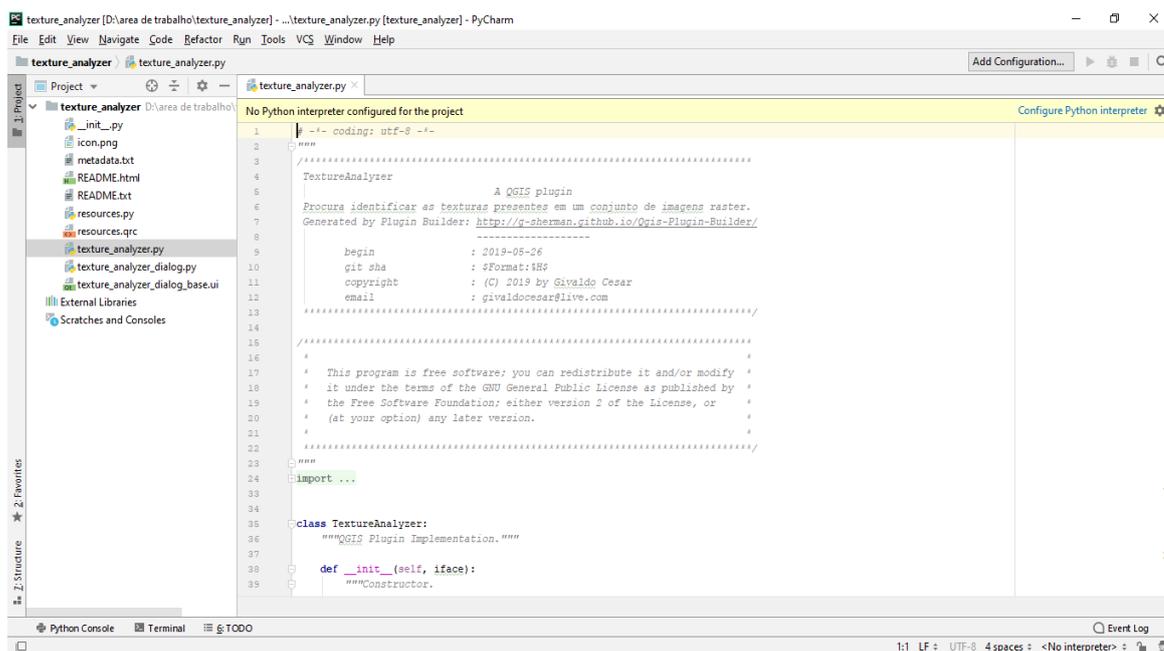
No entanto, qualquer editor de texto puro, isto é, sem formatação, é capaz de editar os arquivos.

Durante a instalação do *PyCharm*, marcou-se a opção “Add “*Open Folder as Project*””, pois isto torna muito conveniente a criação de projetos dentro da plataforma.

Após a instalação, clicou-se com o botão direito do mouse sobre o diretório onde se encontra o complemento *Texture Analyzer* e então na opção “*Open Folder as PyCharm Community Edition Project*”. Uma janela conforme a Figura 14 foi apresentada.

<sup>5</sup> Integrated Development Environment ou Ambiente de Desenvolvimento Integrado, em tradução livre.

Figura 14 - Tela inicial PyCharm.



Fonte: O autor, 2019

Daí, acessou-se o arquivo “*texture\_analyzer.py*”, clicando duas vezes sobre o mesmo.

Após abrir o arquivo, o *PyCharm* exibiu uma mensagem de erro, requisitando a configuração de um interpretador Python. Para corrigir isto foi necessário acessar o menu “*File*”, a opção “*Settings*”, a aba “*Project*” e então “*Project Interpreter*”. Daí clicou-se em  e em “*Add*”. Na janela apresentada, marcou-se “*Inherit global site-packages*”, selecionou-se um local em “*Location*”, onde o ambiente virtual será gerado, e então se selecionou um interpretador Python em “*Base Interpreter*”. Aqui foi selecionado o arquivo “*python.exe*” presente no diretório “*Apps/Python37*” no diretório de instalação do QGIS.

Após clicar em “*Ok*”, uma lista com todos os módulos presentes no interpretador é apresentada. Então se clicou em “*Ok*” novamente, esperou-se o processo de update terminar para iniciar as alterações.

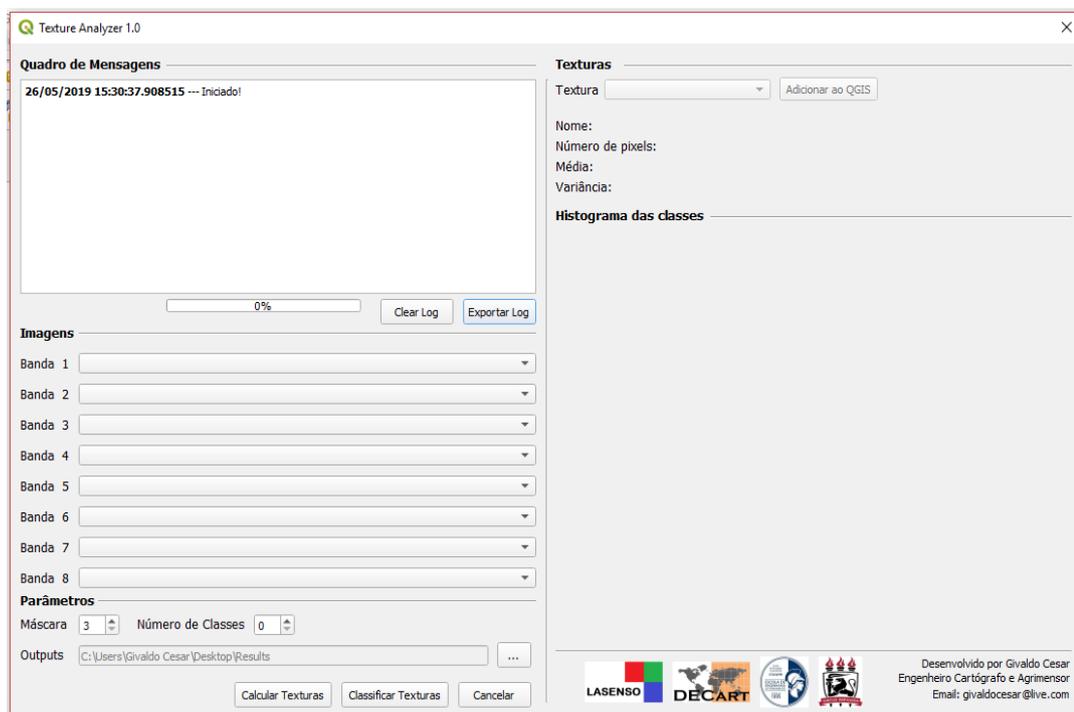
O arquivo “*texture\_analyzer.py*” é a base do complemento, ele é o arquivo python executado quando se ativa o complemento. No entanto, aqui foram feitas pequenas alterações, apenas para garantir uma reinicialização completa deste. O arquivo com o código também se encontra no link <https://github.com/givaldocesar/textanalyzer>.

O arquivo “*texture\_analyzer\_dialog.py*” é a conexão entre a interface gráfica e as rotinas de processamento. Nele foram escritos todos as rotinas e feitas as conexões com seus

respectivos botões. Os códigos com comentários também estão no link <https://github.com/givaldocezar/textanalyzer>.

Após a alteração dos arquivos, o resultado é apresentado na Figura 15.

**Figura 15 - Texture Analyzer no ambiente QGIS.**



Fonte: O autor, 2019

No Quadro de Mensagens são exibidas informações sobre o processamento e resultados. Na área de imagens são inseridas as imagens de entrada, pode-se inserir até oito imagens, desde que tenham a mesma extensão. Há dois parâmetros passíveis de alteração, o tamanho da máscara e o número de classes, por padrão os valores são 3 e 0, respectivamente. O valor 0 para o número de classes, indica que o usuário deseja utilizar a fórmula de Sturges para o cálculo do número de classes.

Na área Texturas é apresentado informações sobre as classes encontradas ao clicar em Classificar Texturas. Na área do Histograma é a apresentada a frequência de cada classe encontrada, isto é útil quando se deseja adicionar uma determinada classe utilizando o botão Adicionar ao QGIS.

Há também a possibilidade de escolher o local onde serão salvos os resultados, imagem-textura e imagens binárias, utilizando o campo Outputs, que por padrão é a área de trabalho do usuário, em OS Windows.

## 4 RESULTADOS E DISCUSSÕES

Aqui são estão descritos os resultados dos testes e análises feitas com respeito ao desempenho do complemento.

### 4.1 TESTE INICIAL COM IMAGEM SINTÉTICA

Este teste foi executado com o intuito de observar o comportamento inicial do complemento. Para tal foi gerada uma imagem sintética com cores diferentes e bem marcadas no software Paint e observada as classes geradas.

Durante o teste observou-se que o QGIS não exibe o resultado, apesar de o complemento gerar uma imagem, caso a imagem de entrada não possua georreferenciamento. Após georreferenciar a imagem-teste, obteve-se o resultado apresentado na Figura 16.

**Figura 16** - Comparação Imagem-teste X Resultado.



Fonte: O autor, 2019

Observa-se uma boa diferenciação entre as quatro cores no resultado, onde se apresentam quatro classes mais homogêneas. Cada borda acabou por ser classificada como uma classe que difere da sua textura vizinha. Porém, ao analisar as imagens binárias, percebeu-se uma confusão entre uma classe e uma borda de outra classe (Figura 17).

**Figura 17** - Imagem binária com confusão de classes.



Fonte: O autor, 2019

Para corrigir isto foi feita uma pequena alteração na Equação (12), sendo o valor de textura foi redefinido como a Equação 13:

$$T_i(x, y) = \sqrt{\bar{x}^2 + Md^2 + DMd^2 + \sigma^2} \quad (14)$$

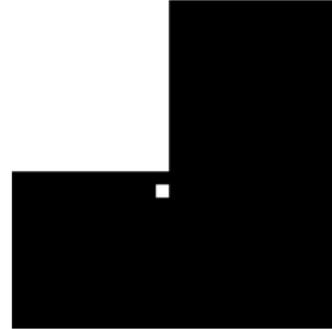
A diferença de entre as medidas sem e com média é apresentado na Figura 18.

**Figura 18** - Comparação entre resultados sem a medida da média e com a medida.

**Resultado sem a medida da média**



**Resultado com a medida da média**



Fonte: O autor, 2019

É fácil observar a melhora significativa na classificação da imagem. Isto ocorre, pois a média é mais uma informação sobre a característica da amostra, diminuindo a confusão do algoritmo.

## 4.2 ANÁLISE DO TEMPO DE EXECUÇÃO

O complemento atribui valores de textura pixel a pixel, sendo assim o tempo de execução depende da quantidade de pixels que uma imagem possui e do tamanho da máscara utilizada, pois quanto maior a máscara mais operações são executadas para obter-se a textura de um único pixel, isto vale para o primeiro passo, o cálculo das texturas.

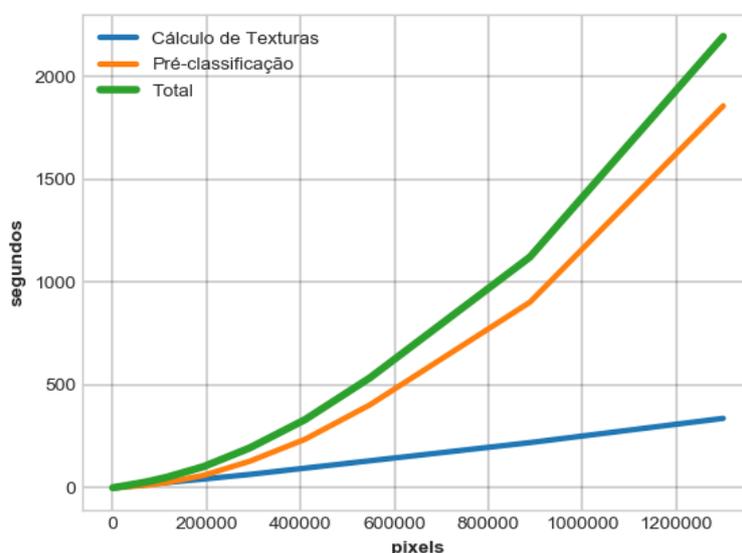
Na classificação, o tempo de execução na pré-classificação é influenciado pela quantidade de texturas analisadas, ou seja, também a quantidade de pixels presentes. Na aplicação do algoritmo de quebras naturais, o tempo de execução depende apenas do comportamento do GVF, pois o processo só é mantido se ele aumenta durante a execução, na primeira degradação a ação é interrompida. Como isso depende da imagem, não é possível ter uma medida generalizada.

Além disto, o tempo de execução depende muito da robustez do computador utilizado, quanto mais robusto, mais rápido o processamento, essa variação não será avaliada aqui.

Assim, nesta etapa, foi analisado o tempo de execução de uma mesma imagem teste redimensionada para diversos tamanhos, sendo medido o tempo de cálculo de texturas e o tempo de pré-classificação. Os parâmetros iniciais são mantidos em 3, para o valor da máscara e 0 para a quantidade de classe, ou seja, a quantidade de classes é calculada pelo método de Sturges.

O tempo de execução entre as etapas de processamento foi obtido através do uso do método “*perf\_counter*” do módulo “*time*” existente no Python nativo do ambiente QGIS. Este método retorna o valor (em segundos fracionários) de um contador de desempenho, ou seja, um relógio com a maior resolução disponível para medir uma duração curta (PSF, 2019).

**Figura 19** - Evolução do tempo de execução.



Fonte: O autor, 2019

A Figura 19 demonstra que o maior peso para o tempo de execução é a pré-classificação (em laranja), no entanto, esta fase é necessária para a otimização de Jenks.

O resultado da pré-classificação na maioria dos testes reduziu o número de loops necessários no cálculo do GVF, pois como usa a equação de Sturges e os dados ordenados, as classes obtidas possuem boa homogeneidade inicial. No complemento, as fases de cálculo de textura e classificação são independentes, apesar de ser necessária uma imagem para classificar. Assim o usuário pode optar por trabalhar com os dados de textura brutos, efetuando uma graduação ou outro tipo de classificação.

### 4.3 TESTES EM IMAGENS GERADAS POR SENSORES ORBITAIS

Nesta etapa foram feitos testes em imagens geradas por sensores orbitais de uma área de estudo com o intuito de revelar o potencial de mapeamento do complemento e comparar o resultado dos diversos sensores utilizados.

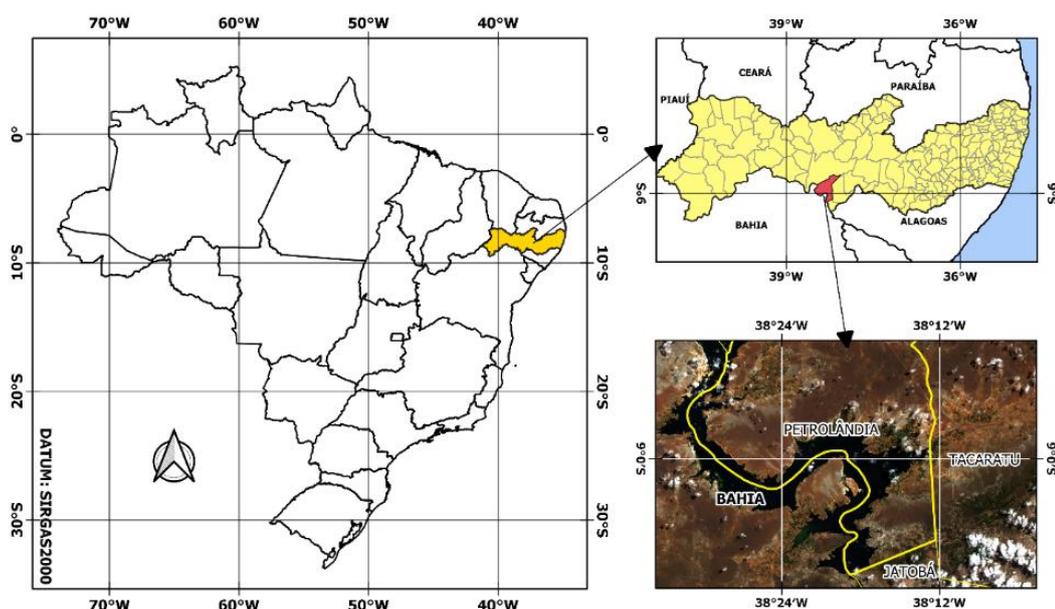
#### 4.3.1 Características da área de estudo

A escolha da área de estudo levou em conta a diversidade do uso do solo presente na região, por isso foi escolhida parte do lago da Usina Luiz Gonzaga, antiga Itaparica, que de acordo com a CHESF (2016) está localizada no Rio São Francisco 25 km a jusante da cidade de Petrolândia em Pernambuco (Figura 20).

Segundo a CODEVASF (2010), a área irrigável gerada pela inundação totaliza 9601 ha em Pernambuco e 3949,5 ha na Bahia, tendo, em 2008, 75% desta área sendo utilizada por alguma cultura agrícola. Há grande diversidade de cultivos na área, como produção de banana, coco, amendoim, goiaba, entre outras, o que gera uma grande heterogeneidade do uso do solo, sendo assim uma ótima área de testes para o complemento.

A usina teve sua construção iniciada na década de 80, tendo assim grande histórico de imagens registradas por diversos sensores inoperantes, o que aumenta a quantidade de sensores disponíveis para teste.

**Figura 20 - Área de Estudo.**



Fonte: O autor, 2019

### 4.3.2 Materiais

Os materiais consistem nas imagens TM, ETM+, OLI, MUX, IRS, PAN e PALSAR caracterizadas no quadro 10.

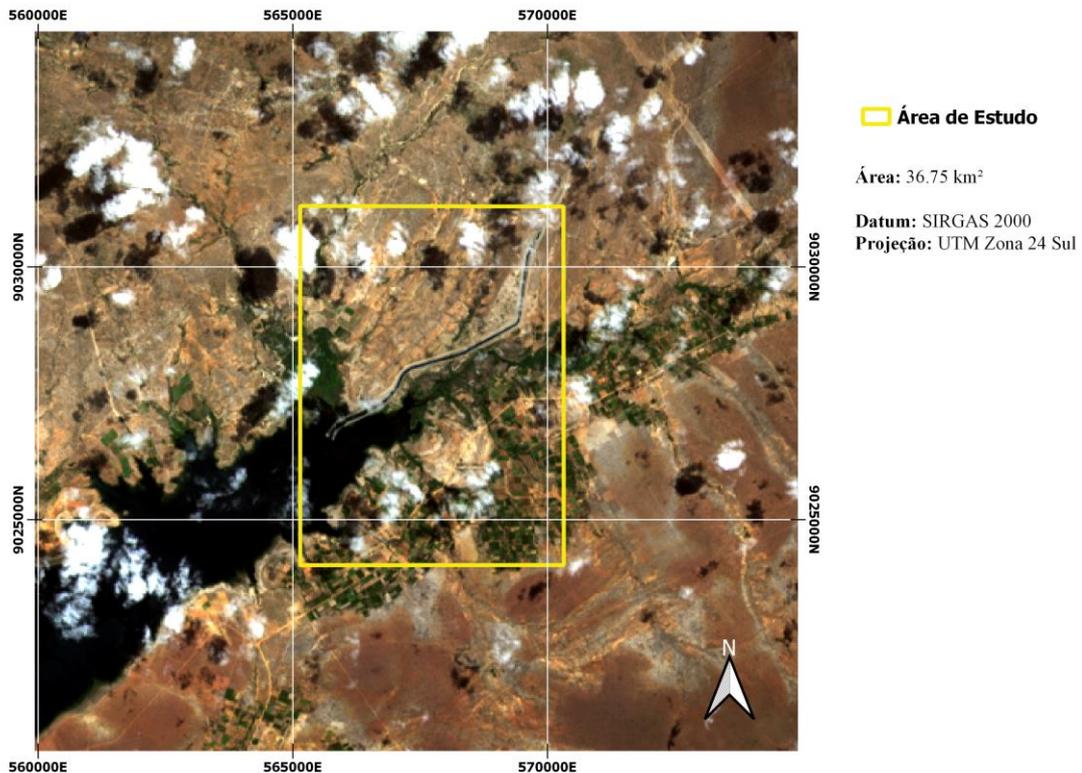
**Quadro 10** - Características dos materiais de estudo.

Sensor	Data de aquisição	Bandas utilizadas	Path	Row	Local de Aquisição	
TM	06/08/2011	1,2,3,4,5,7	216	66	<a href="https://earthexplorer.usgs.gov/">https://earthexplorer.usgs.gov/</a>	
ETM+	05/08/2002	1,2,3,4,5,7	216	66	<a href="https://earthexplorer.usgs.gov/">https://earthexplorer.usgs.gov/</a>	
OLI	25/08/2018	1,2,3,4,5,6,7	216	66	<a href="https://earthexplorer.usgs.gov/">https://earthexplorer.usgs.gov/</a>	
MUX	17/08/2016	5,6,7,8	150	110	<a href="http://www.dgi.inpe.br/catalogo/">http://www.dgi.inpe.br/catalogo/</a>	
IRS	17/08/2016	9,10,11	150	110	<a href="http://www.dgi.inpe.br/catalogo/">http://www.dgi.inpe.br/catalogo/</a>	
PALSAR R	FBS (RTC)	28/11/2007	HH	38	7000	<a href="https://vertex.daac.asf.alaska.edu/#">https://vertex.daac.asf.alaska.edu/#</a>
	FBD (RTC)	28/08/2007	HH, HV	38	7000	<a href="https://vertex.daac.asf.alaska.edu/#">https://vertex.daac.asf.alaska.edu/#</a>

Fonte: Autor.

Como as imagens possuem grande extensão os testes foram executados em um recorte dos materiais. A área selecionada é apresentada na Figura 21.

**Figura 21** - Recorte da Área de Estudo.

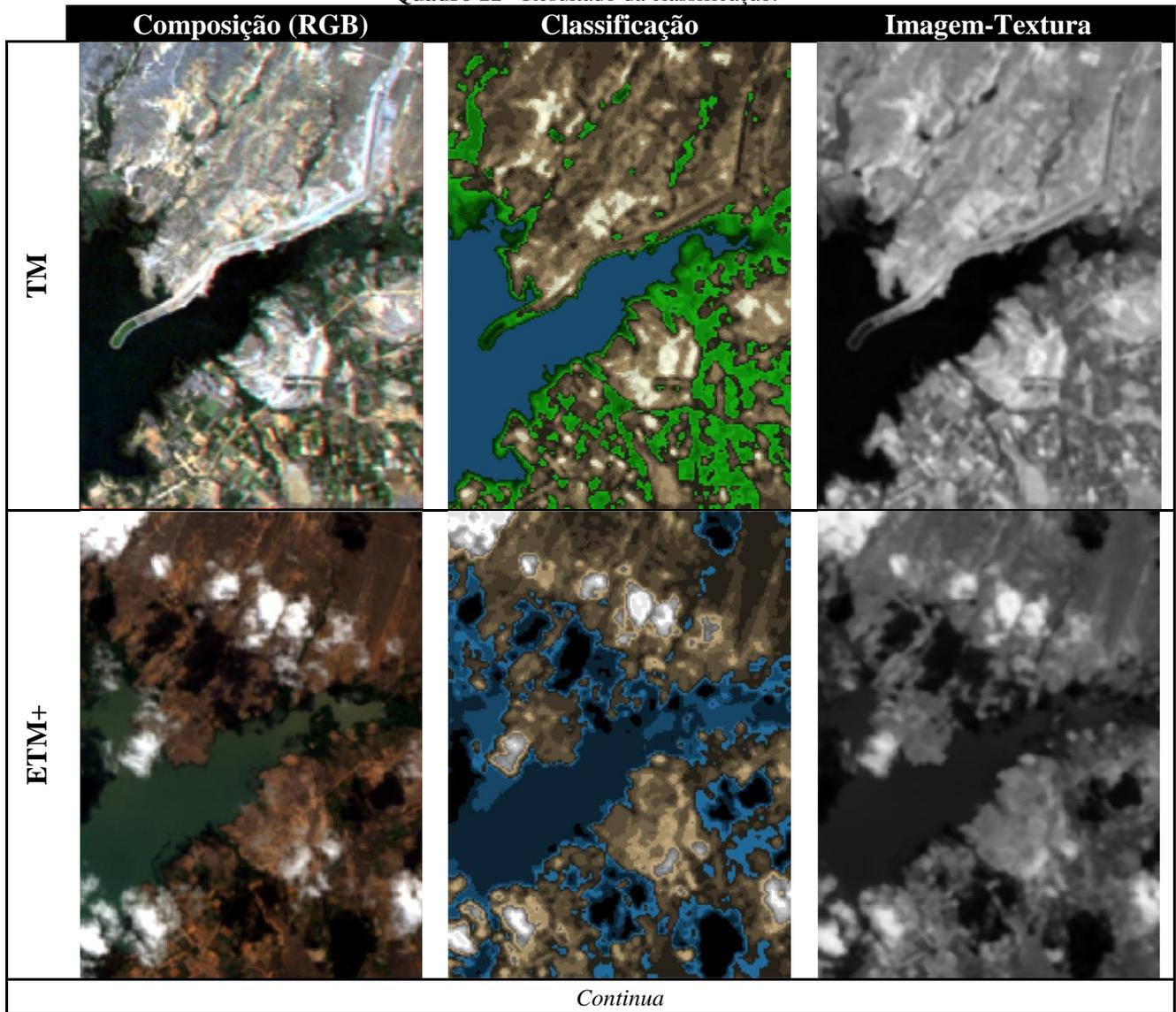


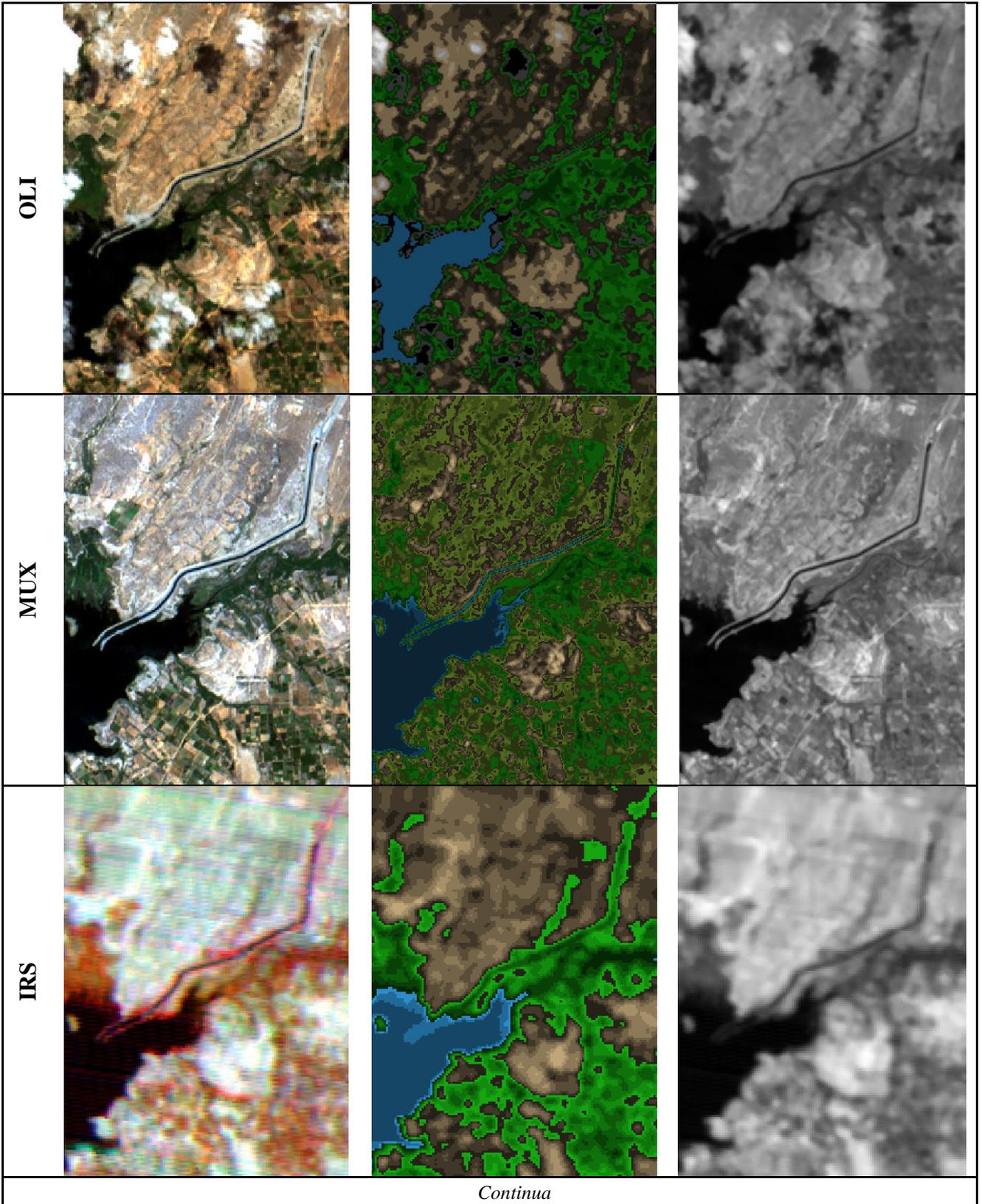
Fonte: O autor, 2019

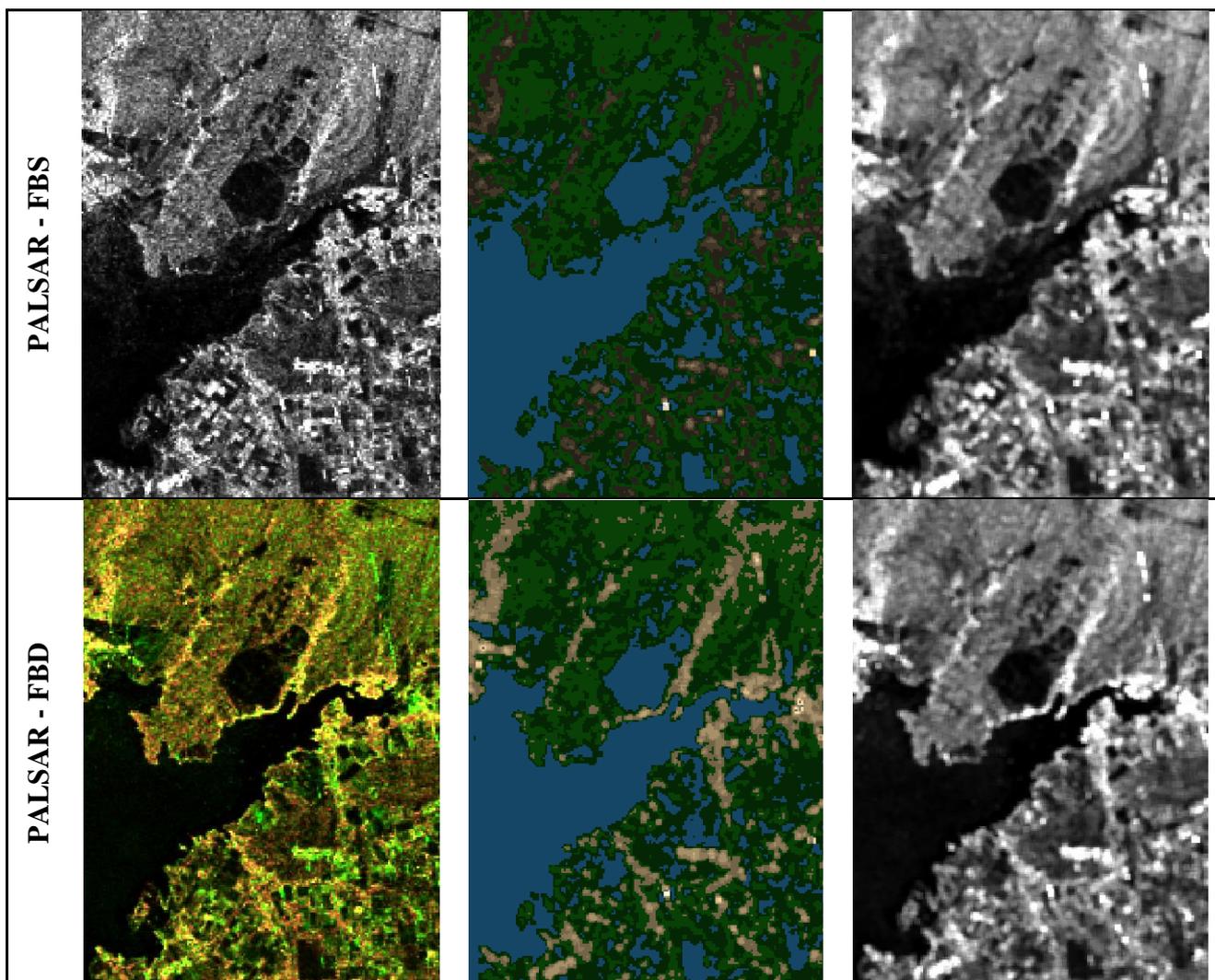
### 4.3.3 Classificação das imagens

Aqui são apresentados os resultados da classificação dos recortes da área de estudo (quadro 11) para uma comparação inicial. Os parâmetros iniciais do complemento foram mantidos no padrão: tamanho da máscara 3 e 0 para a quantidade de classes.

Quadro 11 - Resultado da classificação.







Fonte: O autor, 2019

Ao observar o Quadro 10, notou-se uma grande influência das nuvens e sombras, estas causaram confusão no algoritmo de classificação nas imagens do OLI e ETM+. Na imagem TM, MUX e IRS percebeu-se visualmente uma boa classificação. Nas imagens PALSAR, devido à pouca informação inserida (1 banda no FBS e 2 bandas no FBD) observou-se uma confusão de alvos, no entanto, há um bom mapeamento das áreas escura.

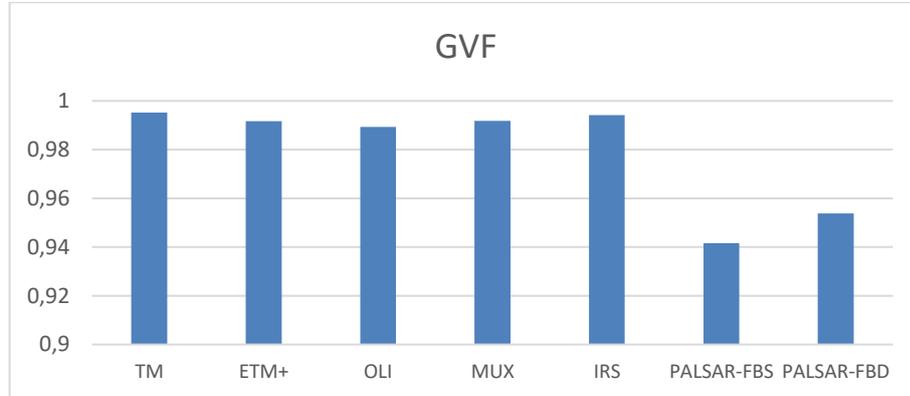
No OLI devido à grande resolução espectral e radiométrica, ou seja, mais informações sobre os alvos, esperava-se os melhores resultados, porém, a presença de nuvens afetou negativamente o desempenho.

#### 4.3.4 Valores GVF

Na Figura 22 estão apresentados os valores GVF obtidos em cada processamento, os dados PALSAR detém os menores valores, mas ainda próximos de 1, o que indica boa

homogeneidade. Estes valores podem estar abaixo dos demais devido à pouca informação inserida e a presença do speckle.

**Figura 22 - Valores GVF.**



Fonte: O autor, 2019

#### 4.3.5 Coeficientes de correlação

A Tabela 1 apresenta os valores do coeficiente de correlação calculados entre a imagem-textura e as bandas de entrada, com o uso da Equação (11), como as imagens-textura apresentam-se semelhantes as bandas de entrada, espera-se uma correlação linear entre ambas.

**Tabela 1 - Coeficientes de correlação.**

<b>TM</b>						
<b>Banda 1</b>	<b>Banda 2</b>	<b>Banda 3</b>	<b>Banda 4</b>	<b>Banda 5</b>	<b>Banda 7</b>	
0,892519	0,934050	0,940956	0,826807	0,983646	0,973328	
<b>ETM+</b>						
<b>Banda 1</b>	<b>Banda 2</b>	<b>Banda 3</b>	<b>Banda 4</b>	<b>Banda 5</b>	<b>Banda 7</b>	
0,843501	0,875938	0,968523	0,874458	0,873328	0,919002	
<b>OLI</b>						
<b>Banda 1</b>	<b>Banda 2</b>	<b>Banda 3</b>	<b>Banda 4</b>	<b>Banda 5</b>	<b>Banda 6</b>	<b>Banda 7</b>
0,749203	0,803637	0,913331	0,934807	0,805567	0,952431	0,944930
<b>MUX</b>				<b>Banda 7</b>	<b>Banda 8</b>	
<b>Banda 5</b>	<b>Banda 6</b>			0,892328	0,816297	
0,852575	0,924408					
<b>IRS</b>			<b>Banda 10</b>	<b>Banda 11</b>		
<b>Banda 9</b>			0,991636	0,945736		
0,822852						

*Continua*

PALSAR - FBS	
HH	
0,867694	
PALSAR - FBD	
HH	HV
0,843854	0,741931

Fonte: O autor, 2019

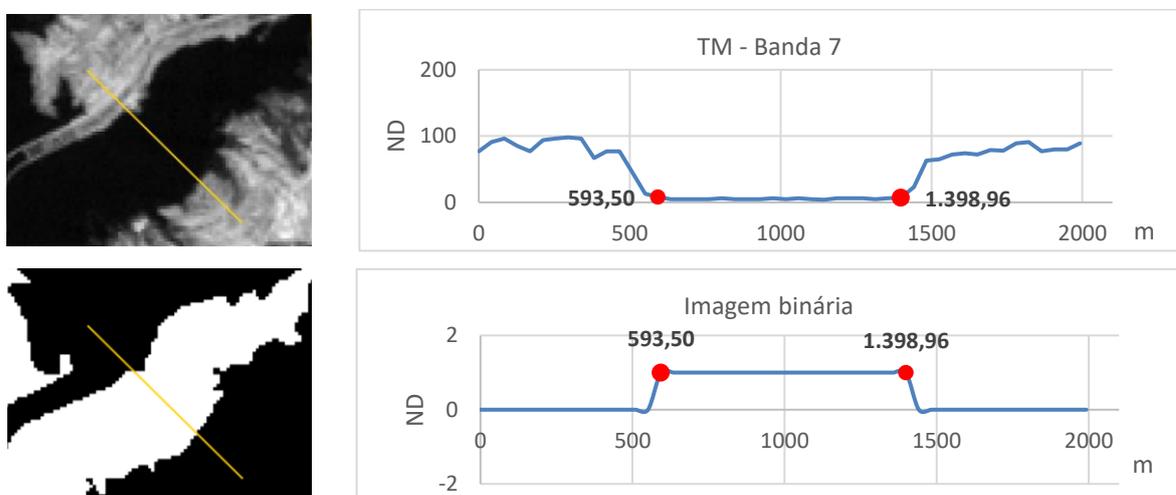
Todos os valores se apresentam acima de 0,7, então de acordo com o quadro 1, as bandas geradas pelos sensores têm uma correlação forte com a imagem-textura, como esperado. E ainda houve valores acima de 0,9 como a banda 5 do TM, o que indica uma correlação muito forte. Assim, a imagem-textura é transporta bem a informação presente em todas as bandas.

#### 4.3.6 Análises de perfis

Nesta etapa foram criados perfis ao longo de uma banda de entrada e uma imagem binária resultante do processamento, e comparado o comprimento coberto ao longo de perfil. Os perfis foram gerados com o auxílio do complemento “*Profile Tools*” disponibilizado no repositório oficial do QGIS e um software de planilhas. O Profile Tools é uma ferramenta que permite obter os números digitais uma imagem em pontos sobrepostos por uma feição linear.

Foi feita análise do potencial de mapeamento de massas d’água, utilizando uma banda SWIR ou NIR dos sensores óticos, pois nessas imagens a água se apresenta escura. Já no sensor RADAR, utilizou-se a polarização HH, pois está presente tanto no modo de feixe FBS quanto no FBD.

**Figura 23** - Comparação Perfil Banda 7 - TM x Imagem Binária.



Fonte: O autor, 2019

Notou-se na Figura 23 que a imagem binária concorda com os valores obtidos da banda 7, indicando bom mapeamento de massas d'água para o sensor TM.

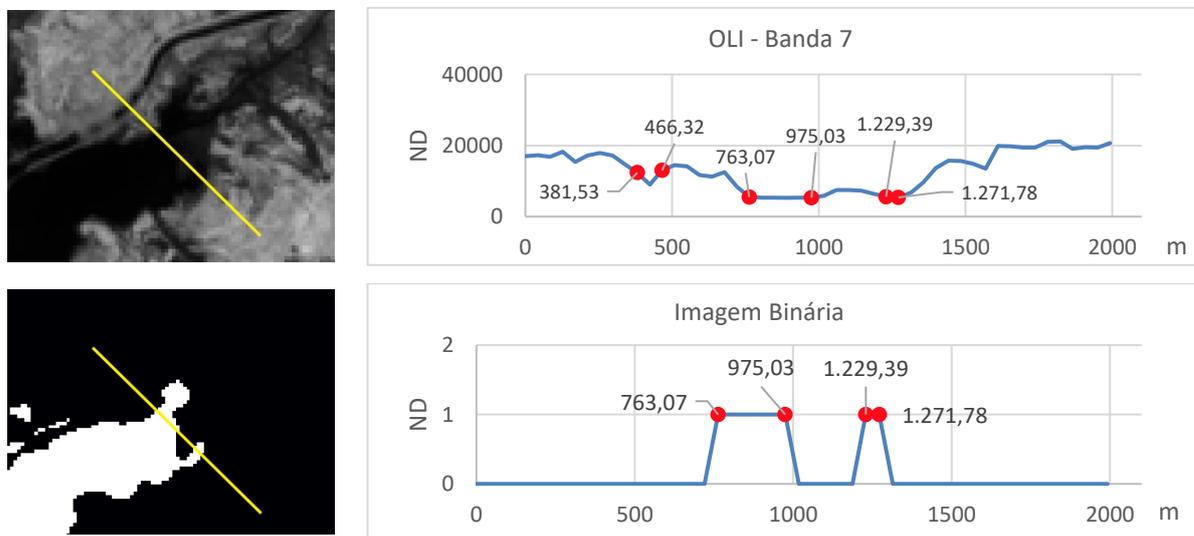
**Figura 24** - Comparação Perfil Banda 7 – ETM+ x Imagem Binária.



Fonte: O autor, 2019

Na Figura 24 também se nota um bom mapeamento.

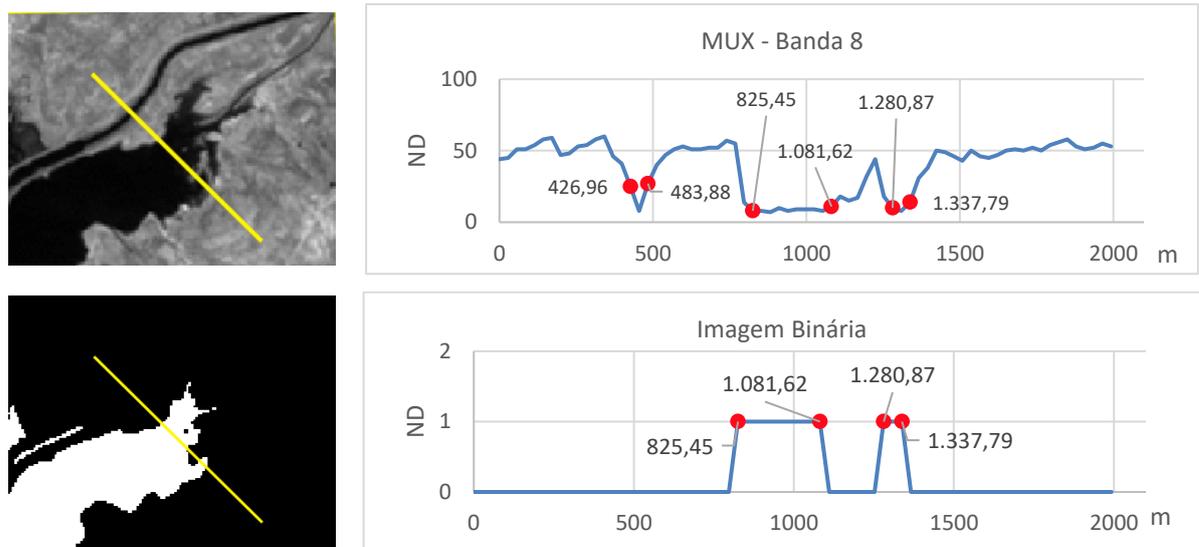
**Figura 25** - Comparação Perfil Banda 7 – OLI x Imagem Binária.



Fonte: O autor, 2019

A Figura 25 apresenta uma pequena diferença entre os perfis, a região referente a um fino trecho d'água que atravessa a imagem na direção SW-SE na parte superior, não foi atrelado a classe de massa d'água, provavelmente devido a sua pequena seção transversal e possível eutrofização.

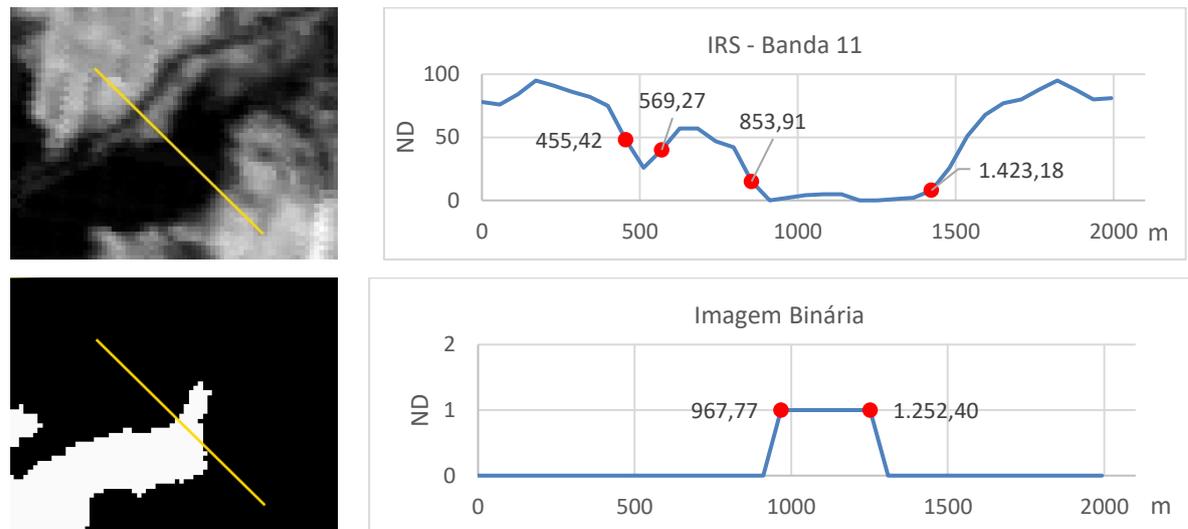
**Figura 26 - Comparação Perfil Banda 8 – MUX x Imagem Binária.**



Fonte: O autor, 2019

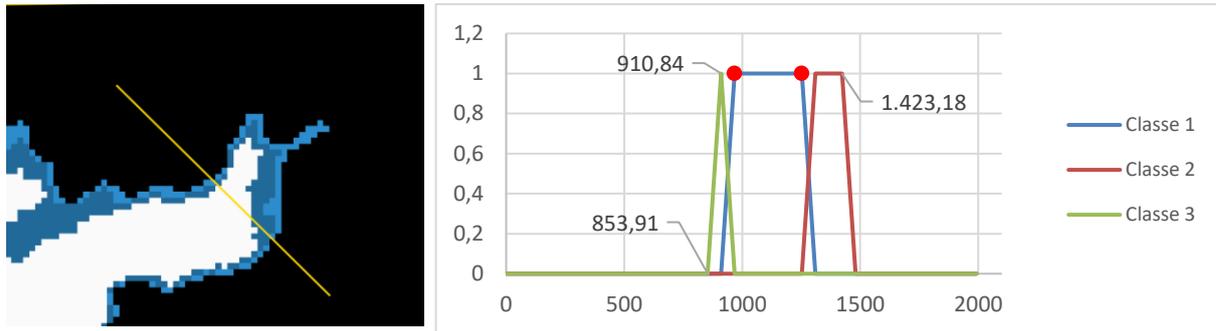
O mesmo aconteceu aqui, como apresentado na figura 26.

**Figura 27 - Comparação Perfil Banda 11 – IRS x Imagem Binária.**



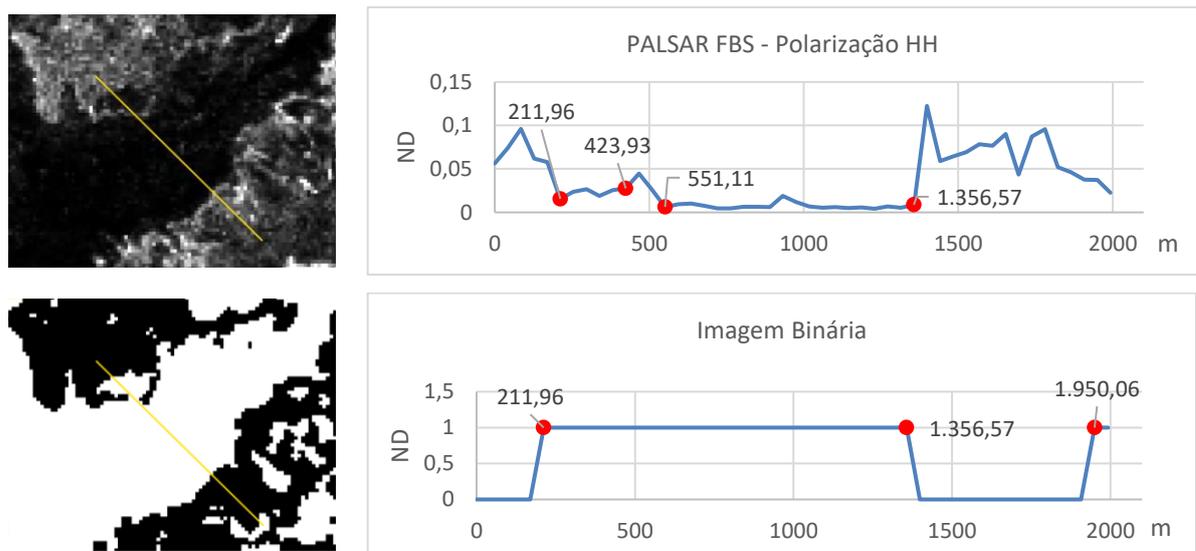
Fonte: O autor, 2019

Aparentemente, a Figura 27 apresenta uma falha na classificação do algoritmo. No entanto ao analisar as outras imagens binárias, percebeu-se que houve uma super-classificação dos objetos semelhantes a massa d'água. Visualmente, esse tipo de alvo apresentou-se em 3 classes distintas e observa-se um bom mapeamento através da união destas (Figura 28), onde a classe 1 está em branco, a classe 2 em azul escuro e a classe 3 em azul claro.

**Figura 28 -** Imagens-Binárias e perfis de classes.

Fonte: O autor, 2019

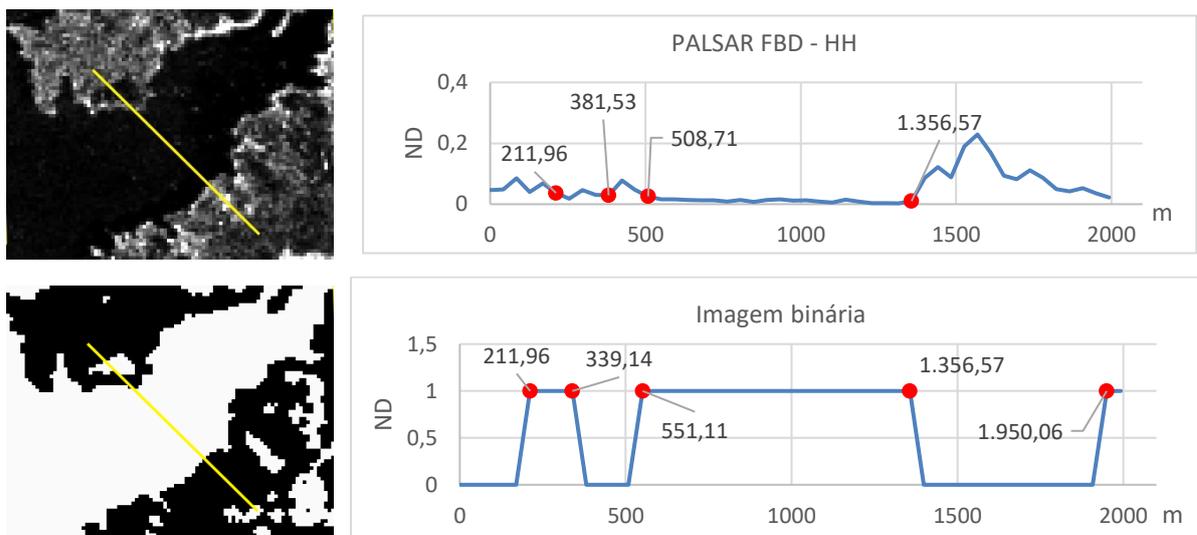
A diferença entre o que foi classificado como água e a posição do que foi identificado na banda 11 do IRS é de aproximadamente 57 m, ou seja, 1,5 pixel. Estas classes podem ser somadas em uma única imagem, pois cada imagem binária não possui interseção entre as outras.

**Figura 29 -** Comparação Perfil Polarização HH – PALSAR FBS x Imagem Binária.

Fonte: O autor, 2019

Houve uma confusão de alvos, ao olhar a imagem PALSAR, uma área escura na parte superior do perfil foi mapeada como massa d'água (Figura 29). Esta é cercada por um trecho claro que não foi, em parte, identificado no processamento. Além disto, uma área clara na parte inferior do perfil também foi identificada como massa d'água.

**Figura 30** - Comparação Perfil Polarização HH – PALSAR FBD x Imagem Binária.



Fonte: O autor, 2019

Já na imagem FBD (Figura 30), não houve esta confusão, no entanto, há uma pequena diferença entre o valor classificado e o identificado na imagem de entrada, esta diferença é de aproximadamente 40 m, cerca de 1,5 pixel. Aqui ocorreu a mesma classificação no fim do perfil, ou seja, o mesmo tipo de alvo existe nas mesmas imagens, sendo necessário, assim, ida a campo para identificação.

#### 4.4 PUBLICAÇÃO DO COMPLEMENTO

Com os bons resultados dos testes, surgiu a necessidade de adicioná-lo ao repositório oficial do QGIS3, para tal alguns passos e recomendações foram seguidos.

Primeiro, foi verificada a existência de complementos com nomes semelhantes no repositório oficial (<http://plugins.qgis.org/>), onde não foram encontrados resultados relevantes.

Após foi criada uma OSGeo Services UserID na página [https://www.osgeo.org/community/getting-started-osgeo/osgeo\\_userid/](https://www.osgeo.org/community/getting-started-osgeo/osgeo_userid/), no entanto para tal, foi necessário adquirir um mantra da OSGeo enviando um email para “[mantra-request@osgeo.org](mailto:mantra-request@osgeo.org)” com informações que verifiquem a identidade, como perfil do GitHub, Linked In ou redes sociais, vale frisar que estes perfis devem conter o email do qual foi enviada a mensagem.

Para upload do complemento é requisitado uma estrutura como apresentada na figura 31, sendo arquivos obrigatórios o “*metadata.txt*” e o “*\_\_init\_\_.py*”, além dos arquivos do código do complemento e interface.

**Figura 31** - Estrutura necessária

```
plugin.zip
  pluginfolder/
    |-- i18n
    |   |-- translation_file_de.ts
    |-- img
    |   |-- icon.png
    |   |-- iconsource.svg
    |-- __init__.py
    |-- Makefile
    |-- metadata.txt
    |-- more_code.py
    |-- main_code.py
    |-- README
    |-- resources.qrc
    |-- resources_rc.py
    |-- ui_Qt_user_interface_file.ui
```

Fonte: [https://docs.qgis.org/2.8/pt\\_BR/docs/pyqgis\\_developer\\_cookbook/releasing.html](https://docs.qgis.org/2.8/pt_BR/docs/pyqgis_developer_cookbook/releasing.html)

Ainda é recomendado, por um arquivo “*README.txt*” com informações de uso e funcionamento do complemento e manter o arquivo “*resources.qrc*”.

Após toda a reestruturação, foi feito o upload do complemento no repositório oficial e também no GitHub (<https://github.com/givaldocezar/textanalyzer>).

## 5 CONCLUSÃO E RECOMENDAÇÕES

Com tudo o que foi apresentado, confirma-se o potencial de mapeamento de alvos hídricos do complemento. Assim, houve sucesso na adaptação do complemento para o ambiente QGIS, que é a finalidade deste trabalho. E ainda houve uma expansão nos experimentos antes realizados, identificando potencial em uma variedade de sensores.

No entanto, é necessário melhorar o tempo de execução do complemento na etapa de classificação, que pode chegar a mais de 2000 segundos para processar imagens que possuam em torno de um milhão e duzentos pixels (ver figura 19).

O algoritmo de processamento não faz uso de nenhuma propriedade física, trabalhando apenas em cima de estatísticas dos valores ND, o que torna simples seu uso e elimina o limite dos tipos de entrada, podendo ser inseridos qualquer tipo de imagem raster, porém são necessários estudos para compreender os resultados que diferentes tipos de imagens apresentarão.

Com as imagens binárias é possível convertê-las em vetor através de diversas ferramentas, como a “*Raster para vetor (poligonizar)...*” do QGIS, e obter diversos outros produtos cartográficos, como polígonos de regiões de interesse e obter diversas propriedades como perímetros, sem a necessidade de vetorização manual. Ainda é possível obter recortes não-regulares com o uso de álgebra de imagens.

Se houver a necessidade de utilizar imagens com grande número de pixels, pode usar a imagem-textura e classificá-la utilizando ferramentas já existentes, abrindo mão das imagens binárias, mas obtendo ganho no tempo de processamento.

Recomenda-se fortemente a utilização de imagens sem nuvens, pois estas atrapalham o processamento. As nuvens geram sombras que afeta a reflectância dos alvos sob estas, alterando seus números digitais, confundindo o algoritmo. Também é recomendado utilizar sensores que possuam alta resolução espectral, pois quão maior for a quantidade de informação sobre os alvos, melhor o resultado esperado.

O complemento permite a alteração do número de classes, recomenda-se utilizar a fórmula de Sturges, valor 0 no campo “Número de classes”, apenas se não for conhecido um suposto número de classes existentes, caso contrário resultará em super-classificação.

Além disso, este trabalho, pode servir de base para aqueles que desejam construir seus próprios complementos, ampliando a diversidade de ferramentas e facilitando o desenvolvimento do QGIS e cartografia.

## REFERÊNCIAS

- Alaska Satellite Facility. **About ALOS PALSAR**. [2019?]. Disponível em: <<https://www.asf.alaska.edu/sar-data/palsar/about-palsar/>>. Acesso em: 23 mai. 2019.
- ARMSTRONG, M. P., XIAO, N., BENNETT, D. A. Using genetic algorithms to create multi criteria class intervals for choropleth maps. **Annals of the Association of American Geographers**, Oxford, v.93 n.3, p. 595–623, 2003.
- CHESF. **Luiz Gonzaga**. 2016. Disponível em: <<https://www.chesf.gov.br/SistemaChesf/Pages/SistemaGeracao/LuizGonzaga.aspx>>. Acesso em: 28 mai 2019.
- CODEVASF. **Sistema Itaparica**.2010. Disponível em: <[https://www2.codevasf.gov.br/programas\\_acoef/sistema-itaparica-1](https://www2.codevasf.gov.br/programas_acoef/sistema-itaparica-1)>. Acesso em: 28 mai 2019.
- DPI/INPE. **Sensoriamento Remoto**. [2006?]. Disponível em: <[http://www.dpi.inpe.br/spring/portugues/tutorial/introducao\\_sen.html](http://www.dpi.inpe.br/spring/portugues/tutorial/introducao_sen.html)>. Acesso em: 22 mai. 2019.
- ESRI. **FAQ: What is the Jenks optimization method?**. 2016. Disponível em: <<https://support.esri.com/en/technical-article/000006743>>. Acesso em: 23 mai. 2019
- GALANT, Katarzyna. **Data classification from cartographic point of view**. Electronic Journal of Polish Agricultural Universities (EJPAU), v. 9, n. 4, Nov, 2006.
- INPE. **Câmeras Imageadores CBERS-3 e 4**. 2019. Disponível em: <<http://www.cbbers.inpe.br/sobre/cameras/cbers3-4.php>>. Acesso em: 23 mai. 2019.
- JAXA. **About ALOS – PALSAR**. [1997?]. Disponível em: <<https://www.eorc.jaxa.jp/ALOS/en/about/palsar.htm>>. Acesso em: 23 mai. 2019.
- JetBrains. **Meet PyCharm**. 2019. Disponível em: <<https://www.jetbrains.com/help/pycharm/meet-pycharm.html>>. Acesso em: 24 mai. 2019.
- NASA. **Landsat 7**. 2019. Disponível em: <<https://landsat.gsfc.nasa.gov/landsat-7/>>. Acesso em: 24 mai. 2019.
- \_\_\_\_\_. **The Thematic Mapper**. 2019. Disponível em: <<https://landsat.gsfc.nasa.gov/the-thematic-mapper/>>. Acesso em: 23 mai. 2019.
- \_\_\_\_\_. **Operational Land Imager (OLI)**. 2019. Disponível em: <<https://landsat.gsfc.nasa.gov/operational-land-imager-oli/>>
- NOVO, E.M.L.M.; PONZONI, F.J. **Introdução ao sensoriamento remoto**. São José dos Campos: [S.n], 2001.
- MAGALHÃES, M.N., LIMA A.C.P. **Noções de Probabilidade e Estatística**. São Paulo: EdUSP, 2004.
- MENESES, P.R.; ROSA, A. N. C. S. **Introdução ao processamento de imagens de sensoriamento remoto**. Brasília: UnB, 2012. Cap 11, p. 168-190.

MENESES, P.R.; SANO, E.E.. **Introdução ao processamento de imagens de sensoriamento remoto**. Brasília: UnB, 2012. Cap 12, p. 191-208.

PAIXÃO JUNIOR, G.C.; TAVARES JUNIOR, J.R. Análise multifonte das texturas presentes na região do reservatório de Itaparica-PE. In: VII SIMPÓSIO BRASILEIRO DE CIÊNCIAS GEODÉSICAS E TECNOLOGIAS DA GEOINFORMAÇÃO, Nov, 2018, Recife. **Anais [...]**. Recife: Recife Praia Hotel, 2018. Disponível em: <<https://www.ufpe.br/documents/39451/1384233/Anais+do+VII+SIMGEO/2ff23c73-935d-4418-8728-fefb24630f34>> Acesso em: 21 mai. 2019.

\_\_\_\_\_. Perfilagem multifonte de borda com mapeador de texturas em Python: baía de Icó-Mandantes, reservatório de Itaparica-PE. In: XIX SIMPÓSIO BRASILEIRO DE SENSORIAMENTO REMOTO, Abr, 2019, Santos. **Anais [...]**. Santos: Mendes Convention Center, 2019. Disponível em: <<https://proceedings.science/sbsr-2019/papers/perfilagem-multifonte-de-borda-com-mapeador-de-texturas-em-python--baia-de-ico-mandantes--reservatorio-de-itaparica>> Acesso em: 21 mai. 2019.

PSF. **General Python FAQ – Python 3.7.3 documentation**. 2019. Disponível em: <<https://docs.python.org/3/faq/general.html>>. Acesso em: 22 mai. 2019.

\_\_\_\_\_. time — Time access and conversions. 2019. Disponível em: <[https://docs.python.org/3/library/time.html#time.perf\\_counter](https://docs.python.org/3/library/time.html#time.perf_counter)>. Análise em: 28 mai. 2019.

QGIS Development Team. **QGIS User Guide**. 2019. Disponível em: <<https://docs.qgis.org/3.4/pdf/en/QGIS-3.4-UserGuide-en.pdf>>. Acesso em: 22 mai. 2019.

SCOTT, D.W. **Sturges’ rule**. WIREs Computational Statistics, v. 1, n. 3, p. 303-306, dez, 2009.

SHERMAN, Gary. **QGIS Plugin Builder**. [2011?]. Disponível em: <<http://gsherman.github.io/Qgis-Plugin-Builder/>>. Acesso em: 26 mai. 2019.

SHIMAKURA, S. E. **Interpretação do coeficiente de correlação**. 2006. Disponível em: <<http://leg.ufpr.br/~silvia/CE003/node74.html>>. Acesso em: 30 mai. 2019.

The Qt Company. **Qt Designer Manual**. 2019. Disponível em: <<https://doc.qt.io/qt-5/qt designer-manual.html>>. Acesso em: 26 mai. 2019.

USGS. **Landsat 7**. 2017. Disponível em: <[https://www.usgs.gov/land-resources/nli/landsat/landsat-7?qt-science\\_support\\_page\\_related\\_con=0#qt-science\\_support\\_page\\_related\\_con](https://www.usgs.gov/land-resources/nli/landsat/landsat-7?qt-science_support_page_related_con=0#qt-science_support_page_related_con)>. Acesso em: 24 mai 2019.

\_\_\_\_\_. **Landsat 8**. 2017. Disponível em: <[https://www.usgs.gov/land-resources/nli/landsat/landsat-8?qt-science\\_support\\_page\\_related\\_con=0#qt-science\\_support\\_page\\_related\\_con](https://www.usgs.gov/land-resources/nli/landsat/landsat-8?qt-science_support_page_related_con=0#qt-science_support_page_related_con)> Acesso em: 23 mai. 2019.