



MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

**DESENVOLVIMENTO DE ROTINAS, EM PYTHON, PARA O  
GERENCIAMENTO DA BASE DE DADOS RADIOMÉTRICOS NA  
PLATAFORMA MAPAQUALI DE MONITORAMENTO DE  
SISTEMAS AQUÁTICOS POR SENSORIAMENTO REMOTO**

Gustavo Kenji Ando

Relatório de Iniciação Científica do  
programa PIBIC orientada por Cláudio  
Clemente Faria Barbosa.

URL do documento original:  
<<http://urlib.net/>>

INPE  
São José dos Campos  
2023





MINISTÉRIO DA CIÊNCIA, TECNOLOGIA E INOVAÇÕES  
**INSTITUTO NACIONAL DE PESQUISAS ESPACIAIS**

**DESENVOLVIMENTO DE ROTINAS, EM PYTHON, PARA O  
GERENCIAMENTO DA BASE DE DADOS RADIOMÉTRICOS NA  
PLATAFORMA MAPAQUALI DE MONITORAMENTO DE  
SISTEMAS AQUÁTICOS POR SENSORIAMENTO REMOTO**

Gustavo Kenji Ando

Relatório de Iniciação Científica do  
programa PIBIC orientada por Cláudio  
Clemente Faria Barbosa.

URL do documento original:  
<<http://urlib.net/>>

INPE  
São José dos Campos  
2023

## RESUMO

O desenvolvimento de rotinas no sensoriamento remoto vem se tornando cada vez mais relevante na atualidade, devido a processos que podem ser facilitados através da programação. Desse modo, durante o período de 2022 e 2023 no INPE foram implementadas novas funcionalidades em pacotes na linguagem de programação Python com finalidade de manipular imagens de satélite e gerenciar dados radiométricos de sistemas aquáticos. Assim, em conjunto de uma orientadora e outros bolsistas, as principais atividades realizadas foram: estudo do pacote AWP (funções dedicadas a redução de efeito da adjacência em corpos de água em imagens MSI/Sentinel-2) e a implementação de um fluxo de processamento adicional para imagens OLCI/Sentinel-3 (o fluxo se resume a imagens que são convertidas em TIFF, recortadas, submetidas a correção atmosférica usando o 6S como base, e são salvas e, após isso, é utilizado um modelo híbrido para estimar a ficocianina para o reservatório de Promissão). A partir disso, foi possível adquirirmos conhecimento e realizar rotinas para a correção das imagens, adquirindo dados de reflectância e ficocianina condizentes com a realidade, já que foram comparadas com dados resultantes de outra colaboradora do projeto, através de rotinas que criam gráficos com dados de pontos específicos para melhor visualização. Além disso, houve uma atividade secundária que foi a atualização do site do Laboratório de Instrumentação de Sistemas Aquáticos (LabISA), que envolvia a adição de novidades do laboratório (notícias, publicações, colaboradores etc.), sendo de grande relevância para a exposição dos resultados da nossa equipe. Portanto, nesse projeto foi possível adquirir experiência em relação à programação no sensoriamento remoto para imagens MSI/Sentinel-2 e OLCI/Sentinel-3 para a execução das correções nas imagens de forma eficaz e concisa, na qual foram avaliadas a partir de gráficos com os dados de reflectância e ficocianina para que de fato estes dados estivessem corretos, além de ter sido trabalhado na exposição da nossa equipe através de um site.

Palavras-chave: Programação em sensoriamento remoto. Sistemas aquáticos. Efeito da adjacência. Correção atmosférica. Modelo híbrido.



## LISTA DE FIGURAS

	<b><u>Pág.</u></b>
Figura 2.1 – Parâmetros iniciais do pacote MSI AWP .....	4
Figura 3.1 – Linhas de código do arquivo runcode.py .....	5
Figura 3.2 – Arquivos do módulo OLCI .....	6
Figura 3.3 – Arquivos de dados auxiliares .....	7
Figura 3.4 – Gráfico de comparação de ficocianina .....	8
Figura 4.1 – Exemplo de arquivo para a adição de novas publicações.....	9



## SUMÁRIO

	<u>Pág.</u>
1 INTRODUÇÃO.....	1
2 PACOTE MSI AWP.....	3
3 Módulo OLCI.....	5
3.1 Metodologia .....	5
3.2 Resultados.....	7
4 SITE LABISA .....	9
4.1 Atualização das publicações.....	9
4.2 Automatizando a compilação e a correção .....	9
4.3 Outras atualizações .....	10
5 CONCLUSÃO .....	11
REFERÊNCIAS BIBLIOGRÁFICAS .....	13





## 1 INTRODUÇÃO

Nos últimos anos, o campo de pesquisa de sensoriamento remoto tem testemunhado uma crescente relevância devido à sua capacidade de simplificar e aperfeiçoar uma variedade de processos por meio da aplicação de técnicas de programação. O desenvolvimento de rotinas no âmbito do sensoriamento remoto têm se mostrado especialmente promissores, permitindo a manipulação eficiente de imagens de satélite para extração de dados radiométricos de sistemas aquáticos. Nesse contexto, entre os anos de 2022 e 2023, no Instituto Nacional de Pesquisas Espaciais (INPE), foram realizadas implementações significativas de funcionalidades em pacotes de programação Python, visando aprimorar a capacidade de processar imagens de satélite e gerenciar dados radiométricos associados a ecossistemas aquáticos.

Neste contexto, houve o estudo do pacote AWP em Python, que se dedica a mitigar os efeitos de adjacência em corpos de água presentes em imagens do satélite MSI/Sentinel-2, que serviram de auxílio para o fluxo de processamento adicional destinado às imagens OLCI/Sentinel-3. Nesse fluxo foram incluídos alguns processamentos para as imagens de satélite e se extraiu alguns dados radiométricos (reflectância e ficocianina) que devem ser validados corretamente. Além das atividades principais, uma tarefa secundária foi realizada: a atualização do website do Laboratório de Instrumentação de Sistemas Aquáticos (LabISA). Desse modo, esse projeto tem intuito de uma extração mais facilitada dos dados radiométricos de imagens de satélite em um pacote de códigos em Python.



## 2 PACOTE MSI AWP

Primeiramente, foi realizada uma análise aprofundada do pacote AWP em Python, o qual compreende um conjunto de funções desenvolvidas para mitigar os efeitos de adjacência presentes em imagens de corpos de água capturadas pelo satélite MSI/Sentinel-2.

A partir do estudo do artigo de mestrado "THE ADJACENCY EFFECT ON INLAND WATER REFLECTANCE: OCCURRENCE AND CORRECTION", por Rejane Paulino, foi buscado compreender o conceito do pacote com base da teoria imposta.

Assim, explicando melhor o pacote, o módulo "msi\_awp", ele foi desenvolvido para abordar o efeito de adjacência em lagos de várzea na região do médio curso do rio Juruá. Esse módulo, escrito em Python, é destinado a imagens do satélite Sentinel-2 (A e B). Ele compreende três etapas principais:

- a) Construção da APSF (Função do Espalhamento Pontual da Atmosfera): Esta etapa envolve a criação de uma função que descreve a probabilidade de um fóton refletido de um alvo adjacente ser espalhado pela atmosfera em direção ao alvo de interesse. A APSF é crucial para modelar e corrigir os efeitos de adjacência.
- b) Determinação do Alcance do Efeito de Adjacência: O alcance do efeito de adjacência é definido através de uma matriz de pixels centrada no pixel de água. Ele é ajustado iterativamente para aperfeiçoar a proporção de alvos diferentes de água dentro do kernel. O índice espectral MNDWI é utilizado para calcular essa proporção, e ela influencia o tamanho da APSF.
- c) Correção do Efeito de Adjacência: Processo para corrigir o Efeito da Adjacência.

O pacote é executado através de um script contido no diretório "runcode", com sete parâmetros de entrada, incluindo caminhos para as imagens MSI em .tiff (input\_path\_band), limiar de separação entre alvos de água e não água (threshold\_nonwater), valores referentes ao mínimo e máximo da proporção

ótima de alvos (p\_min e p\_max), se os valores mínimo e máximo devem ser especificados (default), profundidade óptica do aerossol (aod\_), e altitude do alvo (target\_altitude).

Figura 2.1 – Parâmetros iniciais do pacote MSI AWP

```
input_path_band=r''
path_metadata=r''
threshold_nonwater=0.5
default=True # Default # If False, you have to specify 'p_min' and 'p_max' values.
p_min=40 # Value from 0 to 100.
p_max=50 # Value from 0 to 100 and more than p_min.
aod_=0.14
target_altitude=0.080
```

O arquivo runcode.py executa uma função do bmain.py que faz a união de várias funções em outros arquivos dentro deste pacote. As saídas resultantes deste código, além de imagens intermediárias (com imagens recortadas, por exemplo), são as bandas espectrais (VNIR e SWIR) com 20 metros de resolução espacial corrigidas do efeito de adjacência.

A aplicação prática do módulo msi\_aws foi testada em uma imagem do médio curso do rio Juruá. Os resultados mostraram uma melhora significativa nas estimativas de reflectância da água após a correção do efeito de adjacência, especialmente nas regiões do infravermelho próximo, que são mais afetadas por esses efeitos em águas interiores.

Apesar do sucesso da correção, foram observadas contribuições adjacentes remanescentes em lagos de águas escuras, atribuíveis a fatores como correção atmosférica residual ou o efeito de "glint" na superfície da água.

Além disso, a aplicação do módulo msi\_aws melhorou significativamente a estimativa da profundidade de Secchi, um parâmetro que reflete a transparência da água. Isso destaca a influência dos efeitos de adjacência na recuperação de parâmetros de qualidade da água a partir de dados de satélite.

Em resumo, o módulo msi\_aws oferece uma correção do efeito de adjacência em imagens de satélite Sentinel-2, melhorando a precisão das estimativas de reflectância e parâmetros de qualidade da água, especialmente em ambientes aquáticos de várzea.

### 3 MÓDULO OLCI

A partir do pacote MSI AWP, ocorreu o desenvolvimento do módulo OLCI, um procedimento abrangente foi desenvolvido para tratar as imagens provenientes do sensor OLCI/Sentinel-3. Nesse fluxo, as imagens foram primeiramente convertidas para o formato .TIFF, seguidas de etapas de recorte, aplicação de correção atmosférica baseada no modelo 6S e armazenamento dos resultados. Além disso, foi empregado um modelo híbrido para estimar os níveis de ficocianina específicos para o reservatório de Promissão.

#### 3.1 Metodologia

A criação das funções do código foi realizada a partir de Python e outras referências pesquisas em projetos anteriores (o próprio AWP citado anteriormente) e na internet.

O arquivo que inicia as operações é o runconde.py, sendo necessário especificar as variáveis path\_bands e path\_ROI com os caminhos das bandas da imagem do satélite e o ROI (Figura 3.1).

Figura 3.1 – Linhas de código do arquivo runcode.py

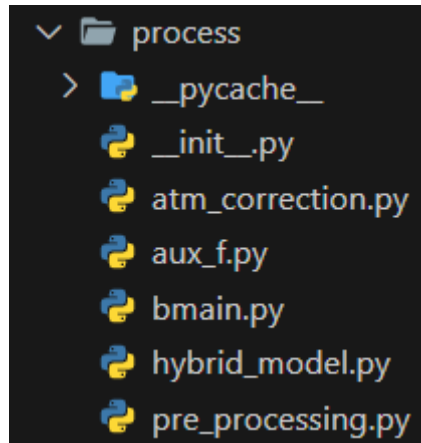
```
## Library:
from OLCIprocess.process import bmain

## Input Data:
path_bands=r''
path_ROI=r''
aod_value=0.07325
target_altitude=0.38607053

if __name__ == '__main__':
    runcode = bmain.RunOLCI(path_bands=path_bands,
                             path_ROI=path_ROI,
                             aod_value=aod_value,
                             target_altitude=target_altitude)
```

Quando este arquivo é rodado, ele executa a função principal do arquivo bmain.py que chama todas as funções organizados em outros arquivos, sendo eles exibidos na Figura 3.2.

Figura 3.2 – Arquivos do módulo OLCI

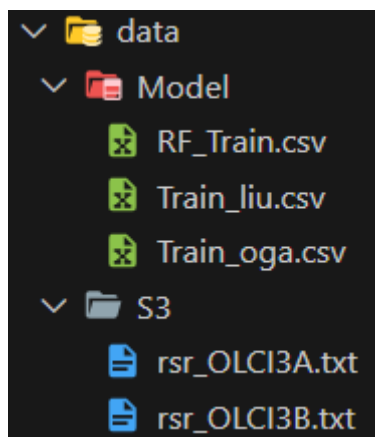


Cada arquivo na pasta process realiza um papel importante no pacote e contém várias outras funções dentro delas, sendo assim, a função de cada um é:

- a) **atm\_correction.py**: responsável pela correção atmosférica das bandas utilizando 6s (py6s foi a biblioteca em Python utilizado);
- b) **aux\_f.py**: realiza funções auxiliares dentro do código, como o de criação de pastas, por exemplo.
- c) **bmain.py**: chama as outras funções na ordem correta de execução;
- d) **hybrid\_model.py**: faz o modelo híbrido para a estimativa de concentração de ficocianina;
- e) **pre\_processing.py**: faz o pré-processamento antes de executar a rotina, realiza a conversão da imagem de netcdf para tiff, por exemplo.

Existem alguns arquivos de suporte na pasta data, com dados externos para a execução da rotina na correção atmosférica e na estimativa de ficocianina (Figura 3.3).

Figura 3.3 – Arquivos de dados auxiliares



Por fim, para execução do módulo, é necessário instalar as bibliotecas, que foram adicionadas por meio de um gerenciador de ambientes virtuais, chamado Anaconda, sendo instalado com o comando no prompt de comando: `conda env create -f environment.yml`.

### 3.2 Resultados

Após o código ser executado, são obtidas quatro pastas:

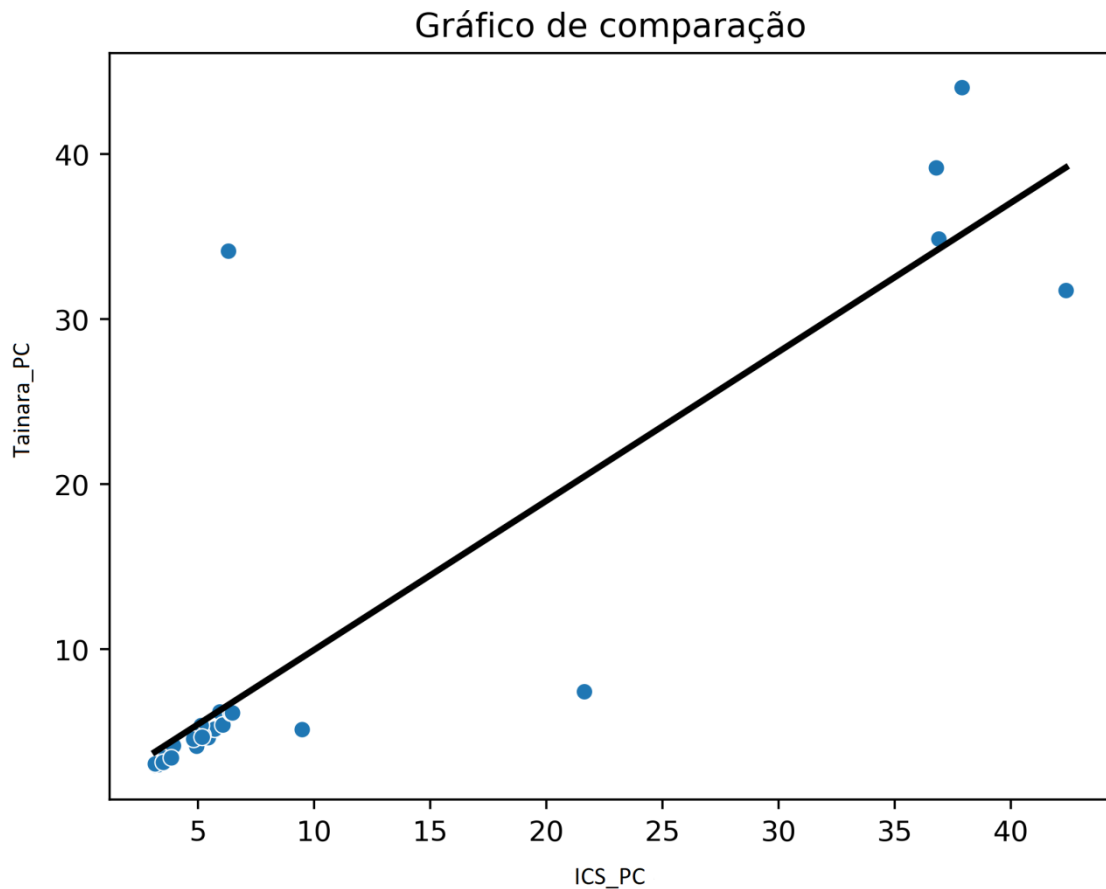
- a) **olciTIFF**: imagens convertidas em tiff;
- b) **cropIMG**: imagens recortadas;
- c) **outputIMG**: imagens com correção atmosférica;
- d) **outputPC**: imagens com estimativa de ficocianina.

Com o auxílio do programa QGIS é possível ter uma melhor análise dos dados das imagens. Porém, para melhor comprovação, utilizamos as imagens com o mesmo processo de outra colaboradora do projeto, para assim conseguirmos validar se os dados estavam realmente corretos.

Com isso, criamos outra rotina para a criação de um gráfico para melhor visualização dos dados entre as imagens que foram produzidas por esta rotina e pelas imagens de nossa outra colaboradora que estavam corretas, escolhendo dados de 30 pontos específicos utilizando um shape. Desse modo, o gráfico obtido foi o seguinte:



Figura 3.4 – Gráfico de comparação de ficocianina



No eixo x temos os valores de ficocianina de uma colaboradora nossa, enquanto no eixo y temos os valores testados por nosso código desenvolvido. Quanto mais próximo os pontos da linha, melhor. São perceptíveis algumas diferenças nos últimos pontos, que apesar de divergirem um pouco, foi dado como um resultado satisfatório, já que não era necessário termos uma precisão de 100% das estimativas de ficocianina.

Portanto, ao longo do tempo foram testados os resultados com a geração de gráficos com alguns erros simples, que foram corrigidos ao longo do desenvolvimento do módulo. Assim, foi possível, através dos estudos feitos anteriormente, pesquisas na internet e com uma comparação analítica, chegar a um resultado satisfatório.

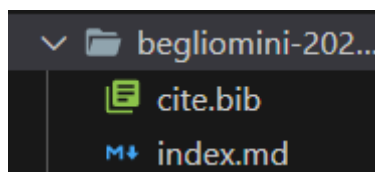
## 4 SITE LABISA

Uma atividade secundária realizada durante este período, foi a atualização do site LabISA, que é de grande importância para apresentação do que está sendo realizado no laboratório. O seu desenvolvimento consiste nas tecnologias: R, HUGO e Blogdown, e é utilizado Python para a correção dos arquivos.

### 4.1 Atualização das publicações

Foram implementadas adições das dissertações de mestrado, artigos publicados e trabalhos apresentados em conferências nacionais do ano passado e deste ano dentro do site. Para essa implementação, com um estilo pré-definido do HUGO e o projeto já desenvolvido, foi apenas necessário adicionar novas pastas em `content/pt/publication` e em `content/em/publication` (já que possui as versões em português e em inglês) com 2 arquivos: `index.md` e `cite.bib`, substituindo os campos já pré-definidos sobre a publicação (título, data de publicação, autor, etc.).

Figura 4.1 – Exemplo de arquivo para a adição de novas publicações



### 4.2 Automatizando a compilação e a correção

Para subir o site no servidor, antes é necessário um processo de compilação, que cria arquivos estáticos em uma pasta chamada docs, para que seja legível para o navegador. Além disso, após compilar, é necessário fazer uma correção dos arquivos estáticos, já que ele é codificado em UTF-8, sendo que o servidor apenas aceita em ISO-8859-1.

Anteriormente, este processo era realizado manualmente. Entretanto, o ideal é uma automatização para facilitar. Então, foi criado um bash (sequências de comandos no prompt) que consiste em criar um ambiente virtual no Anaconda ou no Virtualenv e realizar o comando de compilação do HUGO para depois

executar um arquivo em Python chamado `compile-and-correction.py` (rotina para a correção de caracteres de UTF-8 para ISO-8859-1).

Assim, foi possível a automatização do processo de compilação e correção dos arquivos estáticos para maior usabilidade e economia de tempo.

### **4.3 Outras atualizações**

Além dessas principais funcionalidades realizadas, ocorreram outras atualizações pequenas no site, como a adição de uma novidade na página de novidades, um novo parceiro na tela de início, um novo link na parte de links e a atualização das pessoas que trabalham no laboratório.

## 5 CONCLUSÃO

Neste projeto foi possível realizar diversas atividades na área de tecnologia da informação e sensoriamento remoto que auxiliaram no desenvolvimento de pacotes em Python na manipulação de dados das imagens de satélites e na apresentação do laboratório com o site desenvolvido em R, HUGO e blogdown.

O estudo do pacote AWP foi importante e precedente para a aplicação dos conhecimentos na adaptação do código para o sensor OLCI/SENTINEL-3. Esse que por sua vez, tinha como objetivo o processamento das imagens (conversão, recorte, correção atmosférica e aplicação do modelo híbrido) para a extração da estimativa de ficocianina dos corpos aquáticos.

O desenvolvimento do módulo resultou em um grande avanço para o cálculo de estimativa da ficocianina. Ao gerando quatro conjuntos de imagens dos processos anteriores, sendo o outputPC o principal para avaliação do código. A análise do pacote ocorreu a partir do desenvolvimento de uma nova rotina para criação de gráficos para maior visualização dos dados, que transpareceu um resultado satisfatório.

Além disso, a atualização do site trouxe, além de uma aprendizagem de novas tecnologias, a exposição da participação dos colaboradores do laboratório, que é o ideal para mostrar os resultados.

Portanto, este período no INPE foi de grande aprendizagem e levou ao desenvolvimento de diversas rotinas e do site que auxiliaram na área de sensoriamento remoto do laboratório.



## REFERÊNCIAS BIBLIOGRÁFICAS

Paulino, R. S. **THE ADJACENCY EFFECT ON INLAND WATER REFLECTANCE: OCCURRENCE AND CORRECTION**. 2022. 92 p. Dissertação (Mestrado em Sensoriamento Remoto) – Instituto Nacional de Pesquisas Espaciais (INPE), São José dos Campos, 2022. Disponível em: <http://urlib.net/8JMKD3MGP3W34T/46MN63E>. Acesso em: 08/08/2022.

