

## SEMI-AUTOMATIZAÇÃO PARA CRIAÇÃO DE COLEÇÕES DE IMAGENS PARA O APPLICATIVO ALGAEMAP

WESLEY HUCKEMBECK DOS SANTOS<sup>1</sup>; FELIPE DE LUCIA LOBO<sup>2</sup>

<sup>1</sup>*Universidade Federal de Pelotas – wesleyhuckembeck@hotmail.com*

<sup>2</sup>*Universidade Federal de Pelotas – felipelobo@gmail.com*

### 1. INTRODUÇÃO

O advento dos sistemas orbitais de sensoriamento remoto data da década de 60 e desde seu surgimento, vem facilitando a aquisição de dados espaciais em larga escala, sem a necessidade de contato físico entre sensor e objeto (NOVO; PONZONI, 2001) (BAPTISTA, 2021). Mesmo que a maior parte dos sensores presentes em satélites orbitais hoje em dia, sejam voltados principalmente para estudos terrestres e oceânicos, nos últimos 30 anos grupos de pesquisa vem apropriando-se dessas ferramentas para investigações em águas interiores, em especial, para características relacionadas à qualidade da água, que vem ampliando cada vez mais possibilidades do uso do sensoriamento remoto (BARBOSA; NOVO; MARTINS, 2019).

A ampliação constante das tecnologias e investimentos, permitiu que hoje em dia, 2.666 satélites de observação da Terra estejam em operação (MELO *et al*, 2021). Com a imensidão de dados, em sua maioria gratuitos, o usuário se depara com um novo desafio: um eficiente uso desses dados.

Uma série de ferramentas passaram a ser desenvolvidas para auxiliar o processamento em larga escala de imagens de satélite, contudo, essas requerem um conjunto muito grande de conhecimentos, que dificultam esse acesso, sem contar nas ferramentas computacionais que o usuário necessitará. Da necessidade de facilitar ainda mais esse projeto, é criado o *Google Earth Engine* (*GEE*), uma plataforma que combina uma linguagem de programação simples (JavaScript e Python) a computação em nuvem de ponta, para entregar ao usuário um processamento de imagens em larga escala espaço-temporal (GORELICK *et al*, 2017).

Além de todos os benefícios supracitados, o *GEE* também oferece ferramentas para os pesquisadores desenvolverem aplicações que possam ser compartilhadas com outros usuários (GORELICK *et al*, 2017). Com isso, nasce o *AlgaeMAP*, um aplicativo desenvolvido para prover ao usuário uma coleção de imagens com alta resolução espacial (30m) e temporal (~5 dias) com informações diversas sobre floração de algas em reservatórios da América Latina (LOBO *et al*, 2021).

Contudo o aplicativo possui limitação na taxa de atualização, devido a própria interface do *GEE* que necessitava de um usuário para carregar as imagens uma a uma para o servidor da *Google*. Nesse sentido é onde esse trabalho se enquadra, tendo como objetivo estudar e aplicar um método de semi-automatização para o carregamento de imagens para o servidor do *GEE*.

## 2. METODOLOGIA

Ressalta-se que, desde o princípio deste estudo, já existia um código dentro do *GEE*, que possibilita inserir dados de um reservatório e com isso gerar uma coleção de imagens do satélite Sentinel-2 corrigidas e com informações sobre o *Normalized Difference Chlorophyll-a Index (NDCI)*.

As limitações eram as seguintes: 1. O carregamento da coleção de imagens deveria ser feito mês a mês, desde o ano de 2015 (ano de início de operação do Sentinel-2); 2. Para cada um dos meses, uma média de 12 tarefas era gerada pelo aplicativo, que deveriam ser manipuladas manualmente pelo usuário, gerando um processo oneroso; 3. Essa manipulação também exigia que o usuário modificasse alguns dados como a data das imagens, para que essas fossem adicionadas à coleção de imagens do *AlgaeMap*.

A metodologia aplicada então foi essencialmente exploratória. Em um primeiro momento foram analisados o problemas e realizado uma revisão de conceitos na literatura, junto a uma pesquisa estruturada em fóruns para tentar encontrar soluções similares, que gerou uma possibilidade a ser explorada: a aplicação de um código que semi automatiza o processo sem a necessidade de grandes modificações.

Tendo delimitado a solução, o segundo momento foi dedicado a otimizar o processo, principalmente no código de atualização das coleções para que este operasse da melhor forma possível, fosse auto explicativo, com comentários e livre de erros. O terceiro momento foi apenas para sua aplicação e análise dos resultados obtidos.

O processo utilizado que proporcionou esse avanço trata-se de uma automatização feita na própria interface do navegador do usuário, ou seja, é um código que ao invés de ser inserido dentro do site do *GEE*, é inserido no próprio navegador por meio de atalhos. Ele atua por meio de duas funções, uma para executar toda lista de tarefas geradas e outra para executá-la logo em seguida, possibilitando uma melhora significativa no tempo demandado para a inserção das imagens no banco de dados do aplicativo.

## 3. RESULTADOS E DISCUSSÃO

Ao final dos processos de análise, pesquisa e resolução de problemas, foi possível gerar uma coleção de imagens corrigidas e com dados de *NDCI* de sete anos em torno de uma série de reservatórios na região de Córdoba na Argentina, em um tempo médio estimado de dois dias.

Durante o processo de aprimoramento, também foi possível solucionar a limitação que forçava o usuário a modificar as datas das imagens antes de inseri-las na coleção de imagens, o que diminui ainda mais a onerosidade da tarefa, desta forma, por meio de alguns aprimoramentos internos, agora ao informar o período de correção e dar início a conversão das imagens, estas já iriam ser geradas com informações das datas em seus nomes, facilitando a análise de possíveis erros que possam surgir.

Inspector   Console   Tasks

Manage tasks.   UPDATING...

Search or cancel multiple tasks in the Task Manager.

UNSUBMITTED TASKS

NDCln_mean1648857600000	RUN
NDCln_mean1649030400000	RUN
NDCln_mean1649462400000	RUN
NDCln_mean1649289600000	RUN
NDCln_mean1649721600000	RUN
NDCln_mean1650153600000	RUN
NDCln_mean1649894400000	RUN
NDCln_mean1650585600000	RUN
NDCln_mean1650326400000	RUN

**Figura 1** - Lista de tarefas geradas pelo código, com informações de data em milisegundos.

```
> function runTaskList(){
    var tasklist = document.getElementsByClassName('task local
type-EXPORT_IMAGE awaiting-user-config');
    for (var i = 0; i < tasklist.length; i++)
        tasklist[i].getElementsByClassName('run-button')
[0].click();
    $$('.run-button', $$('ee-task-pane'))
[0].shadowRoot.forEach(function(e) {
    e.click();
})
}
function confirmAll() {
    var ok = document.getElementsByClassName('goog-buttonset-
default goog-buttonset-action');
    for (var i = 0; i < ok.length; i++)
        ok[i].click();
    $$('ee-table-config-dialog, ee-image-config-
dialog').forEach(function(e) {
        var eeDialog = $$('ee-dialog', e.shadowRoot)[0]
        var paperDialog = $$('paper-dialog',
eeDialog.shadowRoot)[0]
        $$('.ok-button', paperDialog)[0].click()
    })
}
```

**Figura 2** - Código de automatização das tarefas

Nas imagens acima podemos observar primeiramente a lista de tarefas gerada a partir do código de conversão de imagens e em seguida o código adaptado para semi-automatizar a inserção destas na coleção de imagens do *AlgaeMap*.

Esse processo ainda se encontra em constantes atualizações e busca por melhorias. Algumas delas são básicas, como converter a data utilizada na nomenclatura das imagens de milissegundos para dia/mês/ano e a finalização de um tutorial completo para que esse trabalho possa ser explorado por qualquer pesquisador que tenha interesse em colaborar com o aplicativo

Outras aplicações são um pouco mais complicadas e passam por algumas limitações da interface do *GEE* tanto como de conhecimento, principalmente em tornar possível a conversão de um ano de imagens a cada ciclo de código, visto que atualmente esse processo é feito mês a mês, o que iria acelerar ainda mais todo o processo.

#### 4. CONCLUSÕES

Conclui-se com esse trabalho que por meio de correções e passos simples, foi possível semi-automatizar um trabalho que antes era oneroso e cansativo, além de retomar os trabalhos e abrir novos ares para um aplicativo que possui tanta aplicabilidade como o *AlgaeMAp*.

Com isso, agora é possível que este seja ampliado para outras localidades e no futuro próximo, não dependa dos seus criadores para ser atualizado, difundido cada vez mais o seu uso no meio acadêmico.

#### 5. REFERÊNCIAS BIBLIOGRÁFICAS

- BAPTISTA, G. M. M. Gestão de áreas protegidas por sensoriamento remoto. In: LADWIG, N. I.; CAMPOS, J. B. (org.). **Planejamento e gestão territorial: áreas protegidas**. Criciúma, SC: UNESC, 2021. Cap. 1. p.33-47.
- BARBOSA, C. C. F; NOVO, E. M. L. M; MARTINS, V. S. **Introdução ao sensoriamento remoto de sistemas aquáticos: princípios e aplicações**. São José dos Campos: INPE, 2019.
- GORELICK, N; HANCHER, M; DIXON, M; ILYUSHCHENKO, S; THAU, D; MOORE, R. Google Earth Engine: Planetary-scale geospatial analysis for everyone. **Remote sensing of Environment**, v. 202, p. 18-27, 2017.
- KONG, D. **Gee Monkey**. GitHub. Disponível em: [https://github.com/gee-hydro/gee\\_monkey](https://github.com/gee-hydro/gee_monkey).
- LOBO, F. L; NAGEL, G. W; MACIEL, D. A; CARVALHO, L. A. S; MARTINS, V. S; BARBOSA, C. C. F; NOVO, E. M. L. M. AlgaeMAp: algae bloom monitoring application for inland waters in latin america. **Remote Sensing**, v. 13, n. 15, p. 2874, 2021.
- MELO, D. H; MENDONÇA, L. F. F; SANTANA, J. O; RAIMUNDO, R. D. P. Evolução da observação da terra por Sensoriamento Remoto. **Revista Brasileira de Sensoriamento Remoto**, v. 2, n. 2, 2021.
- NOVO, E. M. L. M; PONZONI, F. J. **Introdução ao sensoriamento remoto**. São José dos Campos: INPE, 2001.