OPEN JOURNAL SYSTEMS
ISSN: 2675-5491

Revista **B**rasileira de **S**ensoriamento **R**emoto, v.4, n.1. 002-017 (2023)

Revista Brasileira de Sensoriamento Remoto

Melo & Holanda

Automação de downloads diários das observações GNSS das estações da RBMC e das efemérides precisas do IGS

Marcella Fernandes de Oliveira Melo[1], Tiago Fernando de Holanda[2],

[1] Mestranda em Ciências Geodésicas e Tecnologias da Geoinformação, Universidade Federal de Pernambuco (marcella.fernandes@ufpe.br)

[2] Doutorando em geografia, Universidade Federal Fluminense

Histórico do Artigo: Submetido em: 12/12/2022 – Revisado em: 23/03/2023 – Aceito em: 02/04/2023

R E S U M O

O trabalho objetiva a construção de uma rotina para automatizar a aquisição dos dados diários das observações GNSS das estações da Rede Brasileira de Monitoramento Contínuo - RBMC e das efemérides precisas, com intuito de criar uma infraestrutura de acervo geodésico para consulta em processamentos de dados e de séries temporais. Para isso, foi desenvolvido um programa em linguagem *Python* com o auxílio das bibliotecas adicionais *Wget* e *Selenium*, podendo ser executado tanto em sistema *Linux* quanto em sistema *Windows* por meio de um compilador, desde que tenha o *Mozilla Firefox* instalado e conexão com a *Internet*. Como resultado foi obtido um programa com interface e funcionamento intuitivo, mostrando-se uma alternativa de grande importância principalmente aos usuários que tenham necessidade de realizar o *download* de grande quantidade de dados, evitando o envolvimento excessivo com tarefas repetitivas, impedindo possíveis falhas humanas. Além disso, houve êxito em adquirir grandes quantidades de dados organizados em uma estruturação de armazenamento, sendo estes de grande contribuição para a criação da infraestrutura de acervo geodésico para futuras consultas em processamentos de dados e de séries temporais. Ao final, são listados os benefícios atingidos pelo usuário com a utilização do programa em comparação com o método tradicional.

Palavras-Chaves: Aquisição, *Downloads*, RMBC, IGS, *Python*.

Automation of daily downloads of GNSS observations from RBMC stations and accurate IGS ephemeris

A B S T R A C T

The objective of this work is to build a routine to automate the acquisition of daily data from GNSS observations from the stations of the Brazilian Network of Continuous Monitoring - RBMC and the precise ephemeris, with the aim of creating a geodetic collection infrastructure for consultation in data processing and time series. For this, a program was developed in Python language with the help of additional libraries Wget and Selenium, which can be executed both on Linux and Windows systems through a compiler, provided that Mozilla Firefox is installed and Internet connection is available. As a result, a program with an intuitive interface and operation was obtained, proving to be an alternative of great importance, especially for users who need to download a large amount of data, avoiding excessive involvement with repetitive tasks, preventing possible human errors. In addition, there was success in acquiring large amounts of data organized in a storage structure, which made a great contribution to the creation of the geodetic collection infrastructure for future queries in data processing and time series. At the end, the benefits achieved by the user with the use of the program in comparison with the traditional method are listed.

Keywords: Acquisition, Downloads, RMBC, IGS, Python.

Melo, M.F.O., Holanda, T.F. (2023). Automação de downloads diários das observações GNSS das estações da RBMC e das efemérides precisas do IGS. **Revista Brasileira de Sensoriamento Remoto**., v.4, n.1, p.02-17.

1. Introdução

Nas últimas décadas o interesse em se realizar o posicionamento de feições terrestres com alta acurácia tem sido cada vez maior. Nesse sentido, as tecnologias espaciais vêm sendo amplamente empregadas e seu pleno domínio é fundamental para que se tenha um melhor aproveitamento do sistema. O GNSS (*Global Navigation Satellite System*), uma das tecnologias espaciais de posicionamento mais avançadas, tem revolucionado as atividades relacionadas com posicionamento. Devido a isso, a quantidade de aplicações é bastante ampla e continua aumentando, indo desde Geodésia, Geodinâmica, Agricultura de Precisão, Meteorologia, Navegação, até as atividades de lazer (Alves, 2013).

Neste âmbito, o Brasil dispõe de uma rede de estações GNSS de operação permanente, a Rede Brasileira de Monitoramento Contínuo (RBMC), sendo a estrutura geodésica de referência mais precisa do país, cujas informações atendem tanto a comunidade científica quanto a prática. As observações advindas das estações da RBMC desempenham um papel muito importante, principalmente para a realização de processamento GNSS no posicionamento utilizando o método relativo (Costa, 2008). Conforme IBGE (2021a), instituto responsável pela implantação e manutenção da RBMC, sua infraestrutura geodésica é constituída atualmente por 150 estações que se encontram distribuídas ao longo do território nacional. Em algumas situações específicas, principalmente acadêmicas e científicas, é necessário obter grande quantidade de dados, de meses a anos de observações para uma ou mais estações da RBMC. No entanto, todo o procedimento padrão para obtenção dos dados, torna esse procedimento moroso e susceptível a erros, demandando a elaboração de soluções para suprir esta dificuldade.

De acordo com IBGE (2021b), as estações da RBMC são compostas por receptores que coletam e armazenam continuamente as observações do código e da fase das ondas portadoras transmitidas pelos satélites das constelações GPS ou GLONASS. As observações são organizadas, ainda na memória do receptor, em arquivos diários, correspondendo a sessões iniciando às 00h01min e encerrando às 24h00min (tempo universal), com intervalo de rastreio de 15 seg. Após o encerramento de uma sessão, os arquivos com as respectivas observações são transferidos do receptor para o Centro de Controle da RBMC, na Coordenação de Geodésia (Rio de Janeiro-RJ). A partir deste ponto, são criados novos arquivos em formato padrão RINEX2 (*Receiver INdependent EXchange format 2*), nos quais é realizado um controle de qualidade das observações. Em seguida os arquivos de dados RINEX2 e as órbitas transmitidas são compactados e disponibilizados na área de *download* do portal do IBGE (IBGE, 2021c) via FTP (*File Transfer Protocol*).

De maneira similar, a obtenção de efemérides precisas, representa parte integrante do procedimento padrão para tratamento de observações pelo método do Posicionamento por Ponto Preciso, método bastante utilizado na geodésia e em outras áreas que requerem alta acurácia. De acordo com Seeber (2003), as efemérides precisas possuem acurácia superior às efemérides transmitidas, pois são estimadas com base em observações de uma rede global de confiança e alta qualidade, como, por exemplo, a rede do IGS (*International GNSS Service*). De modo análogo à obtenção de dados da RBMC, todo o procedimento padrão para obtenção dos dados, requer elevado grau de interação entre o usuário e o portal, tornando esse procedimento moroso e susceptível a erros na obtenção de grande quantidade de arquivos, requerem esforços para proporcionar soluções mais práticas.

O IGS trata-se de um Centro Técnico do IERS (*International Earth Rotation and Reference System Service* - Serviço Internacional de Rotação da Terra e de Sistema de Referência) em assuntos relacionados com o GPS, o GLONASS e outros sistemas de navegação por satélite planejados para a próxima década. O IGS é um Serviço da IAG (*International Association of Geodesy*) e também da FAGS (*Federation of Astronomical and Geophysical Data Analysis Services*) (Monico, 2008). O serviço IGS está baseado em uma rede global de monitoramento e estações de rastreamento GNSS onde disponibiliza inúmeros produtos, dentre os quais efemérides precisas dos satélites GPS e GLONASS. O Brasil está integrado à rede IGS

através da RBMC, e os dados das estações são repassados para um centro global da rede, situado no *Crustal Dynamics Data Center* (CDDIS) da NASA (*National Aeronautics and Space Administration*) (IGS, 2021).

O IGS produz três tipos de efemérides e correções para o relógio dos satélites, denominadas de efemérides IGS (resultante da combinação das órbitas de vários centros de análises disponíveis a partir do 13º dia após a coleta dos dados), IGR (órbitas IGS rápidas, disponíveis com latência de 17 horas) e IGU (órbitas IGS ultrarrápidas disponível com latência de 3 horas) (IGS, 2021). O arquivo das efemérides precisas é composto pelas coordenadas X, Y e Z dos satélites, em quilômetros, atualmente referenciadas ao ITRF2005 (*International Terrestrial Reference Frame* 2005), além das correções dos relógios dos satélites, em microssegundos, os quais são dados, em épocas equidistantes, a cada 15 minutos (Monico, 2008). Conforme Spofford e Remondi (1996), a identificação das efemérides precisas se dá com base na sigla do órgão que a produziu, na semana GPS correspondente e no dia da semana GPS (começa com 0 no domingo e vai até 6 no sábado). A extensão utilizada é o sp3 (*Standard Product 3*), representação ASCII definido pelo NGS (*National Geodetic Survey*).

Partindo das dificuldades supracitadas, este trabalho tem como objetivo apresentar soluções para automatizar *downloads* diários de observações GNSS das estações da RBMC e efemérides precisas do IGS, visando à criação de uma infraestrutura de acervo geodésico para consulta em processamento de dados e de séries temporais. Para tanto, foi elaborada uma rotina em linguagem *Python*, usando diversas bibliotecas para concluir o objetivo final além de técnicas de *Web Scraping* fazendo requisições HTTP (*Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto) a um servidor da *Web*, na finalidade de extrair páginas HTML (*HyperText Markup Language* - Linguagem de Marcação de Hipertexto) (Mitchell, 2016; Vargiu & Urru, 2013), demandando apenas conexão com a internet e a instalação das bibliotecas adicionais *Wget* e *Selenium*.

2. Materiais e Métodos

2.1. Idealização da rotina Python

Embora o acesso aos bancos de dados do IBGE (Instituto Brasileiro de Geografia e Estatística) e do IGS seja livre, em algumas situações específicas, principalmente acadêmicas e científicas, há uma dificuldade em se obter esses dados em grande quantidade ou até mesmo dados mais antigos que já não existem mais no acervo das duas instituições. A motivação de se obter uma rotina de automatização de *downloads* para criação de acervo geodésico ocorreu por uma constante demanda de alunos do Laboratório de Geodésia do Departamento de Engenharia Cartográfica da Universidade Federal de Pernambuco, sobre estudos relacionados com séries temporais de dados posicionais. Como atividade rotineira, os alunos desempenham a árdua tarefa de obter e processar grande quantidade de dados GNSS da RBMC e das efemérides precisas do IGS. Apesar do processamento possibilitar uma automatização por meio de *softwares* para processamento GNSS, a obtenção dos dados ainda é uma tarefa altamente desgastante e trabalhosa, podendo consumir semanas de trabalho para realização das atividades repetitivas.

Envolvidos nesse contexto, este problema abrange um problema comum a todos os usuários de dados GNSS que precisam obtê-los em grande escala ou dados mais antigos que já não existem mais no acervo das duas instituições. Dessa forma, a solução foi a idealização de uma infraestrutura de acervo geodésico para consulta em processamentos de dados e de séries temporais por meio de uma rotina em linguagem *Python*, uma das linguagens mais populares na comunidade de desenvolvedores de soluções para aquisição e tratamento de dados geoespaciais, e com funcionamento por meio de acesso aos bancos de dados das duas instituições provedoras via protocolo HTTP.

2.2. Linguagem Python

Inventado por Guido van Rossum em 1989, *Python* está entre as linguagens de programação mais utilizadas ao redor do mundo. Desde sua estreia no mercado em 1991, ela vem se destacando e evoluindo de acordo com seu tempo. Essa linguagem possui tipagem forte e dinâmica como sua principal característica. Deixando o código mais limpo e eficiente (Yeda et al, 2018).

O *Python* é muito conhecido por sua consistência e transparência. Segundo Piotrowski (2016), essa linguagem é interpretada, todavia programadores podem utilizá-la como *script*. Suporta os mais comuns paradigmas, como orientação a objetos, até aqueles pouco utilizados, como a funcional. Com uma variedade de bibliotecas e uma comunidade *opensource* sempre ativa, o *Python* se mostra uma linguagem muito popular e de fácil acesso. Além disso, os interpretadores de *Python* estão disponíveis para os principais sistemas operacionais (*Linux, Mac OS, Windows, Android, iOS*, BSD, etc.).

2.3. Integrated Development Environment (IDE)

A forma mais comum para programar em *Python* é através do uso de um ambiente de programação, também conhecido como ambiente integral de desenvolvimento (*Integrated Development Environment* – IDE). De acordo com Python Brasil (2021), existem inúmeras IDE's para se programar em *Python*. Dentre essa ampla gama de possibilidades, tem-se: *Idle, PyCharm community, Komodo-Edit, NetBeans, Spyder 2, Eclipse, IPython* e *PyScripter*. Para o presente trabalho será abordado o *PyCharm community*, o qual é uma multiplataforma desenvolvida pela companhia *JetBrains*. Esta edição é liberada sob a licença da *Apache*. Essa IDE fornece análise de código, um depurador gráfico, um testador de unidade integrado, integração com sistemas de controle de versão (VCSes), e suporta desenvolvimento de *web* com *Django*.

2.4. Web Scraping

Embora *Web Scraping* não seja uma terminologia nova, há alguns anos esta prática tem sido mais comumente conhecida como *screen scraping*, *data mining*, *web harvesting* ou variações similares. Em tese, *Web Scraping* é a prática de coletar dados de maneira automatizada, fazendo requisições HTTP (*Hypertext Transfer Protocol* - Protocolo de Transferência de Hipertexto) a um servidor da *Web*, na finalidade de extrair páginas HTML (*HyperText Markup Language* - Linguagem de Marcação de Hipertexto) ou outros tipos de arquivos encontrados em um site (Mitchell, 2016; Vargiu & Urru, 2013).

2.5. *Hypertext Markup Language* (HTML)

O HTML é uma linguagem utilizada para representação visual de conteúdo na *Web*, inicialmente proposta por Tim Berners-Lee (1989). Embora esteja em constante evolução desde o seu início, a gramática básica dos documentos HTML tem se mantido, fazendo com que esta linguagem seja um dos padrões mais importantes para o trabalho na *Web* (Munzert, 2015).

Ainda segundo a referência supracitada, um arquivo HTML é essencialmente um arquivo de texto, o que torna a linguagem poderosa é a sua estrutura em marcação, que permite definir quais partes do documento deve ser projetada como cabeçalho, *links*, tabelas e afins. Estas definições em marcação acontecem por meio de sequências de caracteres predefinidas, conhecidas como *tags*, que envolvem uma determinada parte do texto. A marcação informa ao navegador como o documento está estruturado. A Figura 1 ilustra um exemplo simples de um código-fonte com as *tags* de abertura e fechamento.

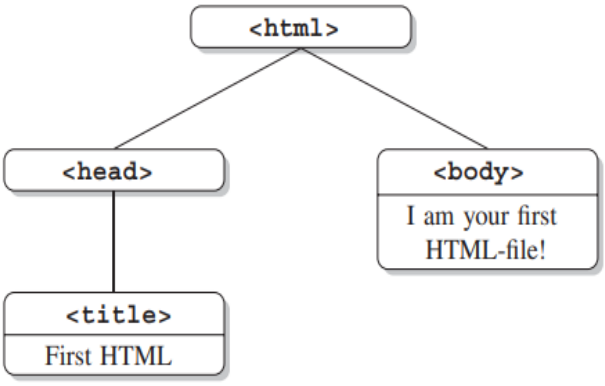
Figura 1 - Exemplo de uma estrutura de um documento HTML.

```
1  <!DOCTYPE html>
2   <html>
3     <head>
4      <title>First HTML</title>
5     </head>
6     <body>
7      I am your first HTML file!
8     </body>
9   </html>
```

Fonte: Munzert (2015).

A estrutura de um documento HTML pode ser interpretada como uma árvore. Como ilustrado na Figura 2, seus elementos internos seguem uma regra rígida de aninhamento: cada *tag* tem seu início e fim. A infração desta regra implica em uma má formação do documento (Munzert, 2015).

Figura 2 - Estrutura em árvore de um documento HTML.



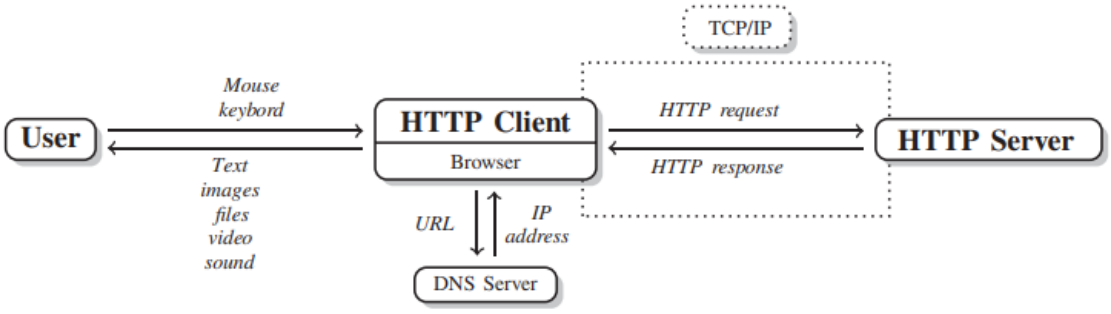
Fonte: Munzert (2015).

O acesso visual a um documento HTML possibilita o conhecimento de sua estrutura, o que implica que é possível então realizar o *parsing* do mesmo. O *parsing* é realizado por meio de expressões regulares, percorrendo sua árvore para encontrar informações que correspondam a alguns padrões e armazenando suas informações de maneira estruturada, na finalidade de formar uma base de dados (Penman, 2009; Vargiu & Urru, 2013).

2.6. *Hypertext Transfer Protocol* (HTTP)

Para retirar dados da internet, é necessário permitir que o software se comunique com servidores e serviços *Web*. O HTTP, Protocolo de Transferência de Hipertexto, é o protocolo mais comum para a comunicação entre um cliente *Web* (navegadores, por exemplo) e servidores. Virtualmente, qualquer página HTML, imagem ou vídeo que é visto por meio de um navegador é entregue por este protocolo. O protocolo HTTP pode requisitar praticamente qualquer tipo de recurso disponível em um servidor e também pode ser usado para o envio de dados para o servidor (Munzert, 2015). A Figura 3 ilustra como ocorre a interação entre usuário e servidor.

Figura 3 - Interação entre usuário e servidor.

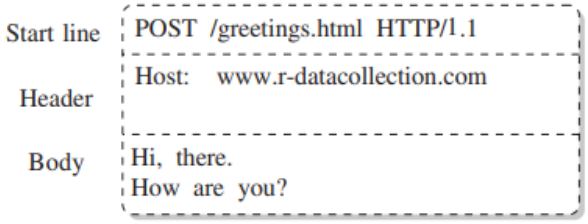


Fonte: Munzert (2015).

Para acessar o conteúdo da *Web*, costuma-se digitar URLs no navegador ou simplesmente clicar em *links*. Por trás dessa camada projetada para interação com o usuário, existem várias outras camadas (técnicas, padrões e protocolos) que fazem tudo funcionar. Juntos, são chamados de IPS (*Internet Protocol Suite*). Os protocolos mais proeminentes desse conjunto são TCP (*Transmission Control Protocol*) e IP (*Internet Protocol*), no qual representam a camada de Internet (IP) e a camada de transporte (TCP), sendo responsáveis pela transferência confiável de dados entre computadores na rede (Forouzan, 2002; Fall, 2011; Munzert, 2015).

Quando um website é acessado, o navegador se torna cliente HTTP. O servidor pergunta ao servidor DNS (*Domain Name System*) qual endereço IP está associado com o domínio URL alvo. Após receber esta resposta, o navegador estabelece uma conexão com o servidor HTTP requisitada via TCP/IP. Uma vez que a conexão é estabelecida, cliente e servidor podem trocar informações (Munzert, 2015). No caso do *Web Scraper*, estas informações serão requisições HTTP. Como ilustrado na Figura 4, tais mensagens possuem uma estrutura definida em três partes: linha inicial, cabeçalho e corpo.

Figura 4 - Modelo de uma mensagem HTTP.



Fonte: Munzert (2015).

No caso de uma requisição a linha inicial define o método utilizado, seguida pelo caminho para o recurso e a versão HTTP mais alta que o cliente pode manipular. O cabeçalho contém metadados e o corpo contém os dados carregados pela mensagem.

Existem vários métodos de requisições HTTP, mas para um *Web Scraper* os métodos mais relevantes são GET e POST. Enquanto o método GET solicita documentos ao servidor, o método POST pode ser utilizado para enviar arquivos (Munzert, 2015).

2.7. Projeto de desenvolvimento da rotina Python

O projeto foi realizado exclusivamente na linguagem *Python*, usando diversas bibliotecas para concluir o objetivo final. A linguagem foi selecionada para este trabalho devido a sua flexibilidade e sintaxe limpa, dispondo de um grande conjunto de bibliotecas com recursos adicionais permitindo, portanto, construir programas sofisticados. Além disso, pode executar os códigos de *Web Scraping* exigidos, usando as bibliotecas *Wget* e *Selenium*.

Para melhor organização, controle, reutilização de linhas de comando e o entendimento do todo, de modo a facilitar a expansão organizada e a manutenção do programa, foi realizada uma modularização. Dessa forma, o código foi dividido em três módulos, a saber: "*downloader.py*", "*looper.py*" e "*menu.py*".

Como ambiente integral de desenvolvimento, foi utilizada a IDE da *Jetbrains*, o *PyCharm Community*. Ademais, para compilar os módulos na IDE é necessário que seja instalado na máquina o interpretador do *Python* na versão a qual se deseja programar. Diante disso, nesta seção é apresentado brevemente como baixar e instalar tanto o interpretador *Python* quanto a IDE *Pycharm*, o qual será utilizado para compilar os códigos, além de uma breve descrição dos módulos "*downloader.py*", "*looper.py*" e "*menu.py*", bem como algumas observações relevantes ao leitor.

2.7.1. Download e instalação do interpretador python

Para realizar o *download* do interpretador *Python* basta acessar o site do *Python*, na seção de *downloads*, ou através do link: https://www.python.org/downloads/. Dessa forma, é possível baixar a versão do interpretador *Python* de preferência do usuário, para este trabalho foi utilizada a versão 3.6.0. Após essa etapa, o *download* de um executável será iniciado, ao fim do *download*, basta executar o arquivo com dois cliques. A instalação recomendável segue-se clicando em *Install Now*, onde será instalado em um diretório padrão. Em seguida, uma tela de progresso irá aparecer. Após a finalização, basta clicar em close para concluir a instalação.

2.7.2. Baixando e instalando a IDE PyCharm

Para instalar o *PyCharm*, basta acessar o site da empresa *JeBrains*, na aba Tools e na seção IDEs selecionando a opção *PyCharm*, ou acessar diretamente através do link: https://www.jetbrains.com/pt-br/pycharm/download/#section=windows. Ressalta-se que a opção de *download* que deve ser selecionada é a *Community* por ser a opção gratuita deste IDE. A seguir, é realizado o download de um executável, sendo necessário executar o instalador e aceitar as permissões de segurança. Dessa forma, uma janela será aberta e para concluir a instalação, basta ir clicando em "*Next >*". Após esses passos, basta selecionar se deseja reiniciar agora (*Reboot now*) ou reiniciar depois manualmente *(I want to manualy reboot later*) e clicar em "*Finish*". Por fim, o *PyCharm* estará instalado e pronto para uso.

2.7.3. Instalação de bibliotecas PyCharm

Uma das grandes vantagens de se programar em *Python* é o grande número de bibliotecas disponíveis para inúmeras operações diferentes. Muitas bibliotecas vêm instaladas no interpretador do *Python*, porém caso haja interesse, é possível baixar outras bibliotecas que disponibilizam outras funções, classes, entre outros.

Para realizar a instalação das bibliotecas adicionais através da IDE utilizada, ou seja, utilizando o *PyCharm*, basta seguir o caminho: *File → Settings → Project → Project Interpreter → Install* (ícone de "+

” no canto superior direito da tela). Nesta etapa, basta digitar o nome da biblioteca desejada, no caso do presente trabalho, a biblioteca *Wget* e *Selenium* e seguir sua instalação de forma intuitiva.

2.7.4. “menu.py”

O “*menu.py*” é o módulo para interatividade com o usuário, onde cada opção solicita uma função do módulo “*downloader.py*”. No que diz respeito aos dados de entrada necessários para seu funcionamento, o programa demanda apenas o fornecimento das datas do atual calendário Gregoriano desejadas.

2.7.5. “looper.py”

O “*looper.py*” é o módulo que automatiza o *download* dos dados diários, ou seja, possui uma estrutura de repetição *while* baseada em contagem regressiva, a qual é responsável por executar o módulo “*downloader.py*” periodicamente, no caso, toda 01h00min. Para que os dados sejam baixados diariamente o “*looper.py*” deve estar executando no computador. Além disso, o módulo também informa quanto tempo falta para baixar os dados do IBGE e da NASA de modo que o usuário consiga visualizar o programa funcionando.

2.7.6. “downloader.py”

O "*downloader.py*" é o módulo que possui as funções para realizar os *downloads* das observações GNSS das estações da RBMC e das efemérides precisas, localizadas, respectivamente, no site do IBGE e no site da NASA. Tal módulo é utilizado como uma biblioteca pelos módulos “*menu.py*” e “*looper.py*”, não sendo necessário executá-lo. O "*downloader.py*" foi dividido em funções com a finalidade de organizar o programa em blocos de código para que algumas instruções não precisassem ser repetidas. É necessário ressaltar que a forma de *downloads* de ambos os sites é diferente, o do IBGE é realizado via FTP e o da NASA via HTTP, sendo este último mais complexo por possuir *login* e senha, dificultando assim, a automação.

O *download* via IBGE, como descrito anteriormente, é voltado para obtenção das observações GNSS das estações da RBMC. Seu funcionamento baseia-se em acesso à base de dados via FTP para obtenção dos arquivos desejados. Para o *download* dos dados foi utilizada a biblioteca adicional *wget*, a qual é uma ferramenta, não interativa, para realizar transferências de arquivos através de comunicação com o banco de dados do IBGE via protocolo FTP, utilizando poucas linhas de código, sendo necessário apenas importar a biblioteca, declarar a URL desejada em uma variável e utilizar a função *download()*.

As observações GNSS das estações da RBMC estão disponíveis em estrutura de subdiretórios sendo estes anuais e dias corridos do ano, o diretório inicial para esses arquivos é: https://geoftp.ibge.gov.br/informacoes_sobre_.posicionamento_geodesico/rbmc/dados/.

A URL para o *download* de cada arquivo segue a seguinte lógica: URL do diretório inicial/ano/dias corridos do ano/nome do arquivo. O nome do arquivo se dá com base na sigla da estação RBMC, os dias do ano contados sequencialmente desde o dia 01 de janeiro e a sua ordem de gravação, por exemplo, o arquivo “perc0701.zip” contém em seu nome sua respectiva estação RBMC [RECIFE], o dia corrido do ano [070] e sua ordem de gravação [1]. Dessa forma, a URL de download para o arquivo citado anteriormente tem a forma:
https://geoftp.ibge.gov.br/informacoes_sobre_posicionamento_geodesico/rbmc/dados/2021/070/perc0701.zip.

Para facilitar a organização dos arquivos obtidos, o programa distribui os arquivos baixados em um sistema de pastas, utilizando para isso, a biblioteca “*os”*, a qual fornece funções relacionadas a

funcionalidades que são dependentes do sistema operacional. A partir dessa biblioteca foi possível fazer manipulações de estruturas de diretórios, tais como verificar se o diretório existe, criar diretórios e tratar exceções. Os dados obtidos são separados por ano, mês e o dia dentro de uma pasta denominada IBGE, no qual a nomenclatura do arquivo é a mesma utilizada pela instituição.

O *download* via NASA é realizado por meio do CDDIS, como já mencionado, é voltado para obtenção das efemérides precisas. Seu funcionamento baseia-se em acesso à base de dados via HTTP para obtenção dos arquivos desejados. Anteriormente, os arquivos CDDIS eram acessados via protocolo FTP anônimo que permitia os usuários automatizarem facilmente os *downloads* de arquivos, porém havia problemas do ponto de vista do sistema de segurança. Por causa das restrições de segurança da NASA, após o dia 31 de outubro de 2020, o CDDIS extinguiu o uso do FTP não seguro para *uploads* de arquivos de provedores de dados e passou a utilizar um sistema novo e atualizado, o qual foi projetado para usar protocolo HTTP para *upload* de arquivos (Michael & Noll, 2019; Michael & Noll, 2020).

Atualmente, as operações de *upload* de arquivo CDDIS via HTTP usa o processo de *login* EOSDIS *Earthdata*, requer que todos os provedores de dados se registrem no EOSDIS para uma conta de usuário, têm disponibilidade tanto na *web* quanto em linha de comando, permite a criação de *scripts* com URL e outras opções do usuário, além de eliminar o problema de duas portas com FTP e *firewalls* (Noll & Michael, 2018).

Para realizar o *download* dos dados foi utilizado o *framework Selenium* (Muthukadan, 2021). O *Selenium Python* possibilita a automação de um navegador ou testes em aplicações *web*, além de fornecer uma API simples para escrever testes funcionais ou de aceitação usando o *WebDriver*. O *WebDriver* é o programa que intermedia a comunicação entre um *script* e um navegador. Por meio das funcionalidades do *WebDriver* é possível simular ações do usuário dentro do navegador, como, por exemplo, realizar *login* em um site, rolar a barra da página do site, clicar em botões e redirecionamento.

Cada navegador possui seu próprio *WebDriver* e é geralmente disponibilizado pelos próprios desenvolvedores dos navegadores, sendo necessário, além de ter a biblioteca devidamente instalada em seu ambiente de desenvolvimento, baixar um *WebDriver*. No caso do presente trabalho foi utilizado o *GeckoDriver* (Mozilla, 2021) que é o *WebDriver* para o *Mozilla Firefox*, pois não é necessária nenhuma configuração adicional, basta que ele esteja instalado. Além disso, para manipulação da página *web* com o *WebDriver* foi necessário possuir conhecimento em HTML, pois o programa percorre a página *web* localizando *tags* para executar alguma ação.

As soluções processadas pelos centros de análise IGS estão disponíveis em uma estrutura de diretórios com as semanas GPS no acervo do CDDIS, onde as URLs para acessar os diretórios iniciais são:

Quadro 1 – URLs dos diretórios iniciais das soluções processadas pelo IGS.

Soluções	**URL**
GPS e GPS + GLONASS	https://cddis.nasa.gov/archive/gnss/products/
Somente GLONASS	https://cddis.nasa.gov/archive/glonass/products/

A URL para o *download* de cada arquivo segue a seguinte lógica: URL do diretório inicial/semana GPS/nome do arquivo. Onde o nome do arquivo se dá com base na sigla do órgão que a produziu, na semana GPS correspondente e no dia da semana GPS (começa com 0 no domingo e vai até 6 no sábado), por exemplo, o arquivo "igs21480.sp3.z" contém as efemérides precisas finais [IGS] do domingo [0] da semana GPS [2148] e sua extensão [sp3]. Dessa forma, a URL de *download* para o arquivo citado anteriormente tem a forma: https://cddis.nasa.gov/archive/gnss/products/2148/igs21480.sp3.Z.

Com a finalidade de facilitar a entrada dos dados no programa, foi criada uma função que calcula a semana GPS, fazendo com que o usuário trabalhe apenas com datas do atual calendário Gregoriano ao invés de entrar com a semana GPS. Tal cálculo foi realizado considerando a época do GPS (6 de janeiro de 1980) e

que as semanas são numeradas desde 0 e vão até 1.023 e então retornam para 0. Este ciclo da contagem ("*rollover cycle*") tem 1.024 semanas ou 7.168 dias, que representam aproximadamente 19,6 anos civis. Para tanto, foi utilizado o módulo *datetime* do *Python*, o qual fornece ferramentas para manipulação de datas e horas.

Assim como no *download* via IBGE, para facilitar a organização dos arquivos obtidos, o *download* via NASA também distribui os arquivos baixados em um sistema de pastas. Tendo como arquivo de saída as efemérides IGR, IGS e IGL, sendo a última relativa à constelação GLONASS. Os dados obtidos são separados por ano, mês, dia e sigla do órgão que a produziu dentro de uma pasta denominada NASA, no qual a nomenclatura do arquivo é a mesma utilizada pela instituição.

2.1 Observações relevantes ao leitor

O funcionamento do programa no geral só é possível havendo conexão com a internet durante todo o período de *download* e ter instalado no computador o *Mozilla FireFox*, podendo ser executado tanto em sistema *Linux* quanto em *Windows*. Além disso, os módulos devem estar localizados na mesma pasta juntamente com o *GeockoDriver*.

No que se refere a sua interface, seguindo um propósito de minimização de custo computacional, optou-se por mantê-lo em linha de comando. Adicionalmente, ressalta-se que foram realizados todos os testes de consistência nas possíveis entradas do usuário. Para fins de brevidade, os códigos são omitidos do presente trabalho, estando estes disponíveis para livre acesso à comunidade em Melo (2021), um repositório virtual do *GitHub*. Neste item constam apenas algumas observações e indicações relevantes ao usuário.

Além disso, é importante salientar que na pasta do programa no *GitHub* encontra-se o *GeockoDriver* para o *Windows* na versão 64, caso o sistema do usuário seja diferente, é necessário fazer o *download* para o sistema adequado em Mozilla (2021).

3. Resultados e Discussão

Como mencionado anteriormente, o programa é um conjunto de três módulos denominados de "*downloader.py*", "*looper.py*" e "*menu.py*". Ambos os módulos estão disponíveis em https://github.com/MarcellaFMelo/Program, repositório virtual do *GitHub*, em uma pasta sob o nome de "*program*", com seus respectivos códigos abertos.

A interface inicial do "*menu.py*" é exibida na Figura 5, onde as opções do usuário são: [1] - Baixar dados desde 01/01/2010 (realiza o *download* dos dados desde o dia 01/01/2010, data determinada como mínima pelo desenvolvedor); [2] - Baixar dados desde uma data específica (realiza o *download* dos dados em um intervalo de tempo, variando desde o dia anterior ao inserido pelo usuário até a data atual); [3] - Baixar dados de ontem (realiza o *download* dos dados do dia anterior) e [4] – Sair (Finaliza o programa).

Figura 5 – Exibição da interface inicial do "*menu.py"*.

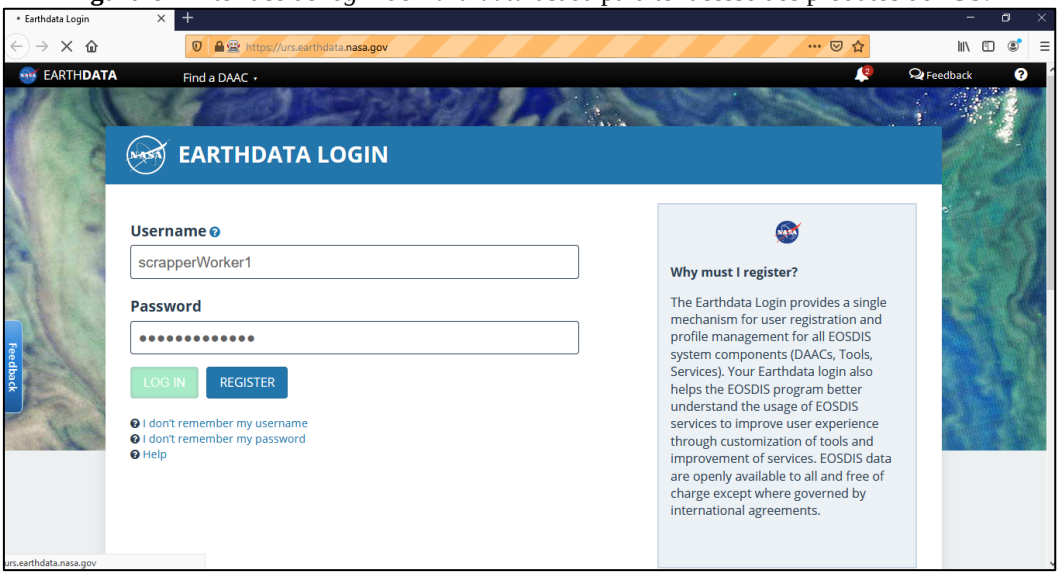
```
=== Downloads das observações GNSS das estações da RBMC e das efemérides precisas do IGS ===

[1]-Baixar dados desde 01/01/2010
[2]-Baixar dados desde uma data específica
[3]-Baixar dados de ontem
[4]-Sair

Favor digitar o número da opção desejada:
```

Todas as opções do "*menu.py*" e o "*looper.py*" demandam o download das efemérides precisas do IGS via NASA. Por consequência, o *GeckoDriver*, *WebDriver* do *Selenium*, realiza a automação do navegador *Mozilla Firefox*, simulando ações do usuário dentro do navegador. Assim, o *GeckoDriver* abre a janela de navegação do *Mozilla Firefox* com a URL de *login* do *Earthdata*, insere o *username* e o *password* já cadastrados nos campos apropriados e clica no botão de *login* automaticamente, como demonstrado na Figura 6.

Figura 6 – Interface de *login* do *Earthdata* usada para ter acesso aos produtos do IGS.



Posteriormente, o programa aguarda 60s antes de iniciar o processo de *download*, pois após o *login*, o site realiza o redirecionamento da página, como ilustrado na Figura 7. Em seguida, o *GeckoDriver* inicia o processo de *download* das efemérides precisas trabalhando com alternância de janelas do navegador. O programa inicia o *Firefox* com o diretório principal (ilustrado na Figura 8), posteriormente, abre o diretório secundário (pasta da semana GPS desejada) em outra janela, baixa o arquivo pretendido e depois fecha a janela, voltando para o diretório principal.

Figura 7 – Página de redirecionamento do *Earthdata*.

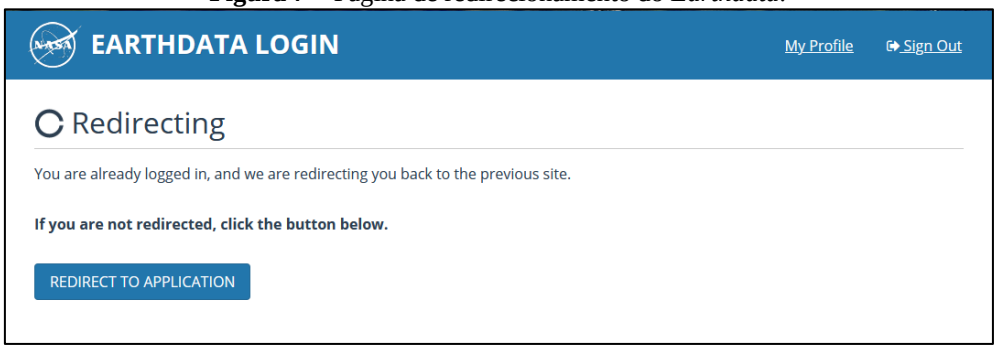
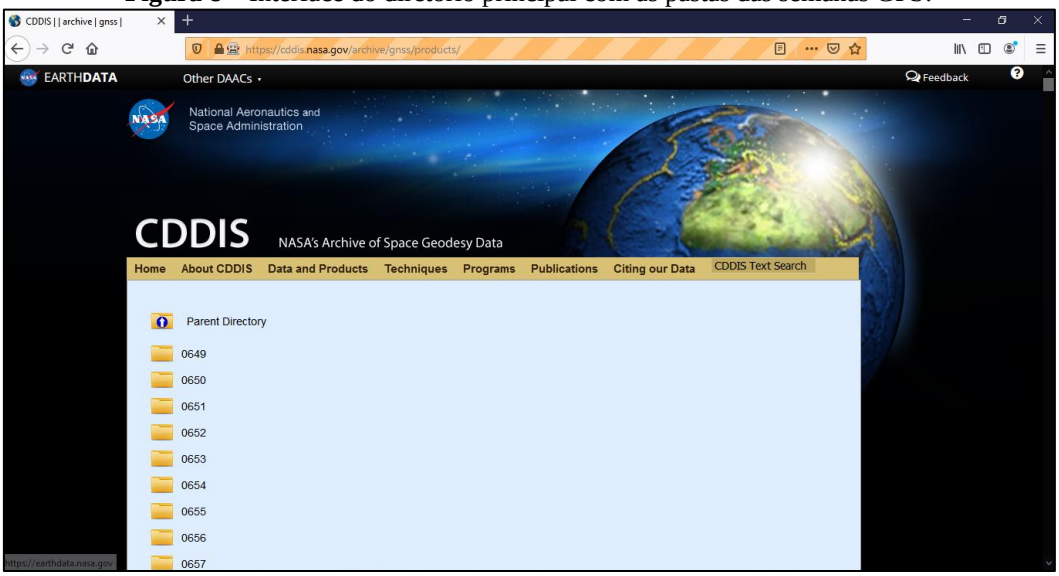


Figura 8 – Interface do diretório principal com as pastas das semanas GPS.



A opção [1] do "*menu.py*" realiza o *download* das observações GNSS das estações da RBMC e das efemérides precisas do IGS desde o dia 01/01/2010, tal data foi definida como mínima devido ao fato de ser a data mínima de *download* no servidor FTP do IBGE. Além disso, o programa pergunta se o usuário tem certeza de que quer baixar todos os dados desde essa data, caso a resposta seja sim, o programa realiza o *download*, como demonstrado na Figura 9. Essa opção se torna relevante na criação da infraestrutura para acervo geodésico, podendo realizar o *download* dos dados disponíveis dos servidores de ambas as instituições.

Figura 9 – Execução da opção [1] do "*menu.py*".

```
===== Downloads das observações GNSS das estações da RBMC e das efemérides precisas do IGS ===

[1]-Baixar dados desde 01/01/2010
[2]-Baixar dados desde uma data especifica
[3]-Baixar dados de ontem
[4]-Sair

Favor digitar o número da opção desejada: 1
Certeza que voce deseja baixar todos os dados desde 01/01/2010?
Sim(S) ou Não(N): S
downloadMinimalYear
01/01/2010 - 07/04/2021
```

A opção [2] do "*menu.py*" realiza o *download* dos dados em um intervalo de tempo, variando desde o dia anterior ao inserido pelo usuário até o atual momento, onde o programa solicita o fornecimento da data desejada, indicando o formato exigido (dd/mm/aaaa). Tal procedimento é ilustrado na Figura 10.

Figura 10 – Execução da opção [2] do "*menu.py*".

```
=== Downloads das observações GNSS das estações da RBMC e das efemérides precisas do IGS ===

[1]-Baixar dados desde 01/01/2010
[2]-Baixar dados desde uma data específica
[3]-Baixar dados de ontem
[4]-Sair

Favor digitar o número da opção desejada: 2
Favor informar a data no formato dd/mm/aaaa. Ex: 31/01/2020
05/04/2021
Download de Periodo: 04/04/2021 - 07/04/2021
```

O "*looper.py*" é o módulo que realiza a automação dos *downloads* dos dados diários, devendo este permanecer executando no computador para o seu funcionamento. Os *downloads* diários são programados para que ocorram todos os dias à 01h00min. Ademais, com o intuito de o usuário conseguir visualizar o programa funcionando, o módulo faz uma contagem regressiva do tempo restante para baixar os dados, como demonstrado na Figura 11.

Figura 11 – Execução do módulo "*looper.py*".

```
Download de ontem: 06/04/2021
10:13:49|para baixar dados do IBGE e NASA
10:13:48|para baixar dados do IBGE e NASA
10:13:47|para baixar dados do IBGE e NASA
10:13:46|para baixar dados do IBGE e NASA
10:13:45|para baixar dados do IBGE e NASA
10:13:44|para baixar dados do IBGE e NASA
10:13:43|para baixar dados do IBGE e NASA
10:13:42|para baixar dados do IBGE e NASA
10:13:41|para baixar dados do IBGE e NASA
10:13:40|para baixar dados do IBGE e NASA
10:13:39|para baixar dados do IBGE e NASA
10:13:38|para baixar dados do IBGE e NASA
10:13:37|para baixar dados do IBGE e NASA
10:13:36|para baixar dados do IBGE e NASA
```

A fim de facilitar a estruturação de armazenamento dos dados para criação de uma infraestrutura de acervo geodésico, o programa distribui os arquivos baixados em um sistema de diretórios dentro da pasta em que os módulos se encontram. A distribuição dos sistemas de pastas dos dados das observações GNSS das estações da RBMC e das efemérides precisas são dadas, respectivamente, por:

(i) *Program* → Data → ano → mês → dia → IBGE → Estações da RBMC;
(ii) *Program* → Data → ano → mês → dia → NASA → Efemérides precisas (IGL, IGR E IGS).

Com o intuito de dimensionar as melhorias impostas pelo programa desenvolvido em relação aos métodos tradicionais de obtenção dos dados das observações GNSS das estações da RBMC e das efemérides precisas, foi utilizada uma série de parâmetros comparativos informais entre ambos, conforme disposto no Quadro 2.

Quadro 2 – Comparativo de obtenção dos dados alvo do trabalho entre o método convencional e o uso do programa.

***Downloads* dos dados pelo método convencional**	***Downloads* dos dados com o uso do programa**
Demanda conexão com a *internet*	Demanda conexão com a *internet*
Processo manual, exigindo tempo e atenção do usuário a cada passo.	Processo automático, permitindo ao usuário a execução de outras atividades enquanto os dados são adquiridos.
Possibilidade de falha humana durante a aquisição.	Não há possibilidade de falha humana durante a aquisição, desde que os dados de entrada estejam corretos.
Demanda organização do usuário ao realizar os *downloads* dos dados.	Facilita a estruturação de armazenamento dos dados, pois o programa distribui os arquivos baixados em um sistema de diretórios.
Maior tempo para aquisição	Menor tempo para aquisição (dependendo da velocidade da conexão com a *internet*).
Processo mecânico e repetitivo, sem aproveitamento entre as etapas consecutivas.	Código livre com aproveitamento de funções intermediárias.
Os sites de ambas as instituições podem ficar fora do ar.	Os usuários terão acesso aos dados através do servidor.

Neste segmento, todos os elementos citados no Quadro 2 e todos os demais pontos abordados ao longo do presente trabalho atuam em prol do programa desenvolvido, ficando perceptível os benefícios relacionados ao uso do programa em comparação com o método tradicional.

4. Conclusão

O presente trabalho teve como objetivo automatizar *downloads* diários de dados GNSS com intuito de criar uma infraestrutura de acervo geodésico para consulta em processamentos de dados e de séries temporais. Para isso, foi desenvolvido um programa em linguagem *Python* com o auxílio das bibliotecas adicionais *Wget* e *Selenium*, podendo ser executado tanto em sistema *Linux* quanto em *Windows* por meio de um compilador, desde que tenha o *Mozilla Firefox* instalado. O programa encontra-se disponível para livre *download* à comunidade em um repositório virtual do *GitHub*, podendo este ser utilizado e implementado desde que devidamente referenciado.

Os módulos do programa possuem uma interface e funcionamentos intuitivos, apresentando-se como alternativa de grande importância principalmente aos usuários que tenham necessidade de realizar o *download* de grande quantidade de dados, evitando o envolvimento excessivo com tarefas repetitivas. Além disso, o programa impede possíveis falhas humanas, sendo capaz de proporcionar mais tempo para a execução de outras tarefas.

Como resultados obteve-se a automatização de *downloads* diários das observações GNSS das estações da RBMC e das efemérides providas, respectivamente, pelo IBGE e pelo IGS, além do êxito em adquirir grandes quantidades de dados organizados em uma estruturação de armazenamento. Tais dados contribuem

para a criação da infraestrutura de acervo geodésico para futuras consultas em processamentos de dados e de séries temporais.

E por fim, a criação de uma infraestrutura de acervo geodésico vai auxiliar os alunos do Laboratório de Geodésia do Departamento de Engenharia Cartográfica e Agrimensura da Universidade Federal de Pernambuco em suas pesquisas, facilitando o tratamento dos dados e permitindo utilizar o tempo gasto de *download* manual para se dedicar à análise e à produção científica. Além disso, futuramente os alunos terão à disposição todos os dados em um servidor apresentando vantagens a usuários que precisem realizar o *download* e se deparem com o inconveniente dos sites fora do ar.

5. Referências

Alves, D. B. M., de Abreu, P. A. G., & Souza, J. S. (2013). GNSS: status, modelagem atmosférica e métodos de posicionamento. **Revista Brasileira de Geomática**, 1(1), 2-7.

Costa, S. M. A., Lima, M. A. A., Júnior, N., Abreu, M. A., Silva, A., & Fortes, L. P. S. (2008). RBMC em tempo real, via NTRIP, e seus benefícios nos levantamentos RTK e DGPS. **Simpósio Brasileiro de Ciências Geodésicas e Tecnologias da Geoinformação**, 2, 8-11.

Fall, K. R., & Stevens, W. R. (2011). **TCP/IP illustrated, volume 1: The protocols**. addison-Wesley.

Forouzan, B. A. (2002). **TCP/IP protocol suite**. McGraw-Hill Higher Education.

IBGE (2021a). **Banco de Dados Geodésicos – BDG**. Disponível em: http://www.bdg.ibge.gov.br/appbdg/. Acesso em: 10/02/2021.

IBGE (2021b). **Rede Brasileira de Monitoramento Contínuo dos Sistemas GNSS - RBMC**. Disponível em: https://www.ibge.gov.br/geociencias/informacoes-sobre-posicionamento-geodesico/rede-geodesica/16258-rede-brasileira-de-monitoramento-continuo-dos-sistemas-gnss-rbmc.html?=&t=sobre. Acesso em: 10/02/2021.

IBGE (2021c). **Download de Produtos - RBMC**. Disponível em: https://www.ibge.gov.br/geociencias/informacoes-sobre-posicionamento-geodesico/rede-geodesica/16258-rede-brasileira-de-monitoramento-continuo-dos-sistemas-gnss-rbmc.html?=&t=downloads. Acesso em: 10/02 /2021.

Melo, M. F. O. (2021). **Program**. Repositório. Disponível em: https://github.com/MarcellaFMelo/Program. Acesso em: 10/02/2021.

Michael, B. P., & Noll, C. E. (2019). NASA CDDIS: Supporting Global Geodetic and Geophysical Research and Applications. *In*: **AGU Fall Meeting Abstracts**. 2019. p. IN31B-0794.

Michael, B. P., & Noll, C. E. (2020). Technology Changes and User Community Reaction-An Example from CDDIS. *In*: **AGU Fall Meeting 2020**. AGU, 2020.

Mitchell, R (2016). **Web Scraping com Python**. São Paulo: Novatec Editora Ltda.

Monico, J. F. G. (2008). **Posicionamento pelo GNSS: descrição, fundamentos e aplicações** – Editora

UNESP. São Paulo.

Mozilla (2021). **GeckoDriver**. Repositório. Disponível em: https://github.com/mozilla/gecko driver/releases. Acesso em: 22/03/2021.

Munzert, S. (2015). **Automated data collection with r a practical guide to web scraping and text mining**. John Wiley & Sons Inc.

Muthukadan, B. (2021). **Selenium with Python.** Disponível em: https://selenium-python.readthedocs.io/index.html. Acesso em: 18/03/2021.

Noll, C. E., & Michael, B. P. (2018). Important Changes to User Access at the NASA CDDIS. *In*: **AGU Fall Meeting Abstracts**. 2018. p. G31B-0675.

Penman, R. B., Baldwin, T., & Martinez, D. (2009). Web scraping made simple with sitescraper. **Penman Web Scraping**, 1-10.

Piotrowski, A. (2016). An Analysis of the use of the Python Language in Robot Applications. **Applied Computer Science**, *12*(2).

Python Brasil (2021). **Ides para python**. Disponível em: https://python.org.br/ferramentas/. Acesso em: 18/03/2021.

Seeber, G. (2003). **Satellite Geodesy**. 2. ed. New York: Walter de Gruyter.

Spofford, P.R. & Remondi, B.W. (2021). **The National Geodetic Survey Standard GPS Format SP3**, 1996. Disponível em: https://www.ngs.noaa.gov/orbits/sp3_docu.txt. Acesso em: 20/02/2021.

Vargiu, E. & Urru, M. (2013). Exploiting web scraping in a collaborative filtering- based approach to web advertising. **Artif. Intell. Research**, v. 2, n. 1, p. 44 – 54.

Yeda, L., Bing, Z., & Renqiang, W. (2018). Application of Python language in UOE molding simulation of pipeline steel. *In*: **MATEC Web of Conferences**. EDP Sciences, 2018. p. 01018.