
ARTIGO CURTO / SHORT PAPER

Automatização do processo de alimentação do NetBox como banco de dados com Paramiko e PyNetBox

 **Alexandre Cauduro** •  alexandre.cauduro@pop-rs.rnp.br

Universidade Federal do Rio Grande do Sul (UFRGS)

 **Cesar Loureiro** •  cesar.loureiro@pop-rs.rnp.br

RESUMO. Este trabalho apresenta uma solução para automatizar a coleta e atualização de dados de dispositivos de rede no NetBox, utilizando as bibliotecas Python PyNetBox, Paramiko e Netmiko. A proposta visa simplificar e acelerar a documentação de redes, coletando informações por SSH e atualizando o NetBox via API REST. A solução foi desenvolvida para suportar dispositivos de diferentes fabricantes, como Juniper, Cisco, Huawei e Extreme, ajustando comandos e processamento conforme necessário. Os resultados mostram que a solução facilita a manutenção de uma base de dados centralizada, melhorando a agilidade, confiabilidade e reduzindo erros manuais. O trabalho destaca o uso das ferramentas Paramiko e Netmiko como essenciais para a automação e integração com o NetBox, oferecendo uma alternativa robusta para o gerenciamento de infraestruturas de TI complexas.

ABSTRACT. This work presents a solution to automate the collection and updating of network device data in NetBox, using the Python libraries PyNetBox, Paramiko, and Netmiko. The proposal aims to simplify and speed up network documentation by collecting information via SSH and updating NetBox through the REST API. The solution was developed to support devices from different manufacturers, such as Juniper, Cisco, Huawei, and Extreme, adjusting commands and processing as needed. The results show that the solution facilitates the maintenance of a centralized database, improving agility, reliability, and reducing manual errors. The work highlights the use of Paramiko and Netmiko as essential tools for automation and integration with NetBox, offering a robust alternative for managing complex IT infrastructures.

PALAVRAS-CHAVE: Paramiko • Netmiko • NetBox • API Rest • PyNetBox

KEYWORDS: Paramiko • Netmiko • NetBox • API Rest • PyNetBox

1 Introdução

O NetBox [1] atua como um repositório central para ativos de rede e pode ser utilizado como um banco de dados acessível via API, permitindo sua automação através da biblioteca PyNetBox. Seu diferencial está na interface web intuitiva e na capacidade de armazenar informações detalhadas sobre equipamentos de rede. No entanto, o processo manual de inserção de dados no NetBox pode ser demorado devido à variedade de campos disponível, o que torna a automação uma solução eficaz para agilizar e simplificar essa tarefa.

Para viabilizar a automação, utilizamos a biblioteca Python chamada Paramiko [2] que permite estabelecer conexões SSH, facilitando a execução de comandos shell e a transferência de arquivos entre dispositivos. Com ela, é possível conectar-se a diversos equipamentos de rede e executar os comandos necessários, basta criar um dicionário com as credenciais (usuário e senha) e informações sobre os fabricantes dos dispositivos.

Um aspecto essencial desse processo é a diversidade de fabricantes envolvidos, como Extreme, Juniper, Cisco e Huawei. Cada um deles utiliza comandos de configuração e formatos de saída específicos, o que de-

manda ajustes no código para lidar adequadamente com essas variações, garantindo a eficiência da automação.

O objetivo deste trabalho é apresentar uma solução automatizada para a coleta, organização e inserção de dados de dispositivos de rede no NetBox, utilizando as bibliotecas Python Paramiko e PyNetBox. A automação proposta visa otimizar o processo de gestão de redes, permitindo a coleta eficiente de informações via conexões SSH, seguida da atualização automática desses dados no NetBox através de sua API REST. Ao reduzir o tempo necessário para a manutenção de inventários de rede e minimizar erros manuais, a solução contribui para uma gestão mais ágil e confiável de infraestruturas de TI complexas, especialmente aquelas que envolvem dispositivos de múltiplos fabricantes.

Nas próximas seções deste trabalho, será apresentada uma revisão dos principais estudos relacionados à automação de redes com o uso de bibliotecas como Paramiko e PyNetBox, destacando suas aplicações e benefícios. Em seguida, a seção de Metodologia descreverá os experimentos realizados, o ambiente de testes e os procedimentos adotados para comparar o desempenho das ferramentas. A seção de Resultados trará uma

análise dos dados coletados, com foco na eficiência da automação proposta. Posteriormente, a Discussão avaliará as implicações desses resultados para a gestão de redes, apontando vantagens e limitações. Por fim, as Considerações Finais sumarizarão as principais contribuições do estudo e sugerirão direções para trabalhos futuros.

2 Trabalhos Relacionados

Mutiara; Isnaini; Suhartono [3] abordaram o uso do Paramiko principalmente para a detecção e manutenção de redes, limitando sua aplicação a um escopo mais básico. No entanto, o verdadeiro potencial do Paramiko vai muito além dessa abordagem inicial, permitindo a automação completa da coleta de dados e da documentação de redes de forma mais eficiente e precisa. Neste trabalho, exploramos essa versatilidade ao integrar o Paramiko com o NetBox, não apenas automatizando a detecção, mas também facilitando a organização e atualização das informações sobre os dispositivos. Essa integração ajuda a minimizar erros manuais e otimiza o processo de gerenciamento de redes complexas. Essa ampliação do uso do Paramiko demonstra um potencial ainda não totalmente explorado para operações mais avançadas no gerenciamento de infraestruturas de rede..

A etapa de automatização envolve o envio de informações para o NetBox, utilizando a biblioteca PyNetBox[4], que facilita o uso da API REST [5] para o gerenciamento de dados. Essa abordagem se assemelha ao trabalho de Novák; Airflow [4] que também utiliza APIs para automatizar processos, embora focado na integração de múltiplas ferramentas visando cibersegurança. No entanto, difere do trabalho de Walther; Jovicic [6], que, embora também utilizem o NetBox, têm como objetivo expandir suas funcionalidades por meio de plugins e integração com ferramentas adicionais, como Nornir e Napalm, para permitir uma automação mais abrangente de serviços de rede. Enquanto o trabalho deles se compromete a estender as capacidades do NetBox para além de suas funções originais, nosso foco está em utilizar as ferramentas nativas do NetBox ao máximo, mantendo a simplicidade e a eficiência na coleta e atualização de dados.

O diferencial deste trabalho está na simplicidade e eficiência na automação da coleta e atualização de dados diretamente no NetBox. Por meio da biblioteca PyNetBox, que facilita a interação com a API REST, e da integração via SSH utilizando Paramiko, a abordagem permite a inserção e atualização contínua de informações de dispositivos de rede de forma prática e direta.

Ao focar no uso otimizado das funcionalidades nativas do NetBox, este projeto oferece uma solução acessível e de fácil implementação, ideal para administradores que buscam automatizar a documentação de redes sem a necessidade de extensões complexas ou configurações adicionais.

3 Metodologia

Paramiko e Netmiko [7] são bibliotecas Python projetadas para a automação de redes via SSH. O Paramiko permite estabelecer conexões SSH seguras e executar comandos remotamente, sendo amplamente utilizado para tarefas de automação e gerenciamento de servidores. Por outro lado, o Netmiko é uma biblioteca que se baseia no Paramiko, oferecendo um nível adicional de abstração voltado especificamente para a automação de dispositivos de rede. Ao contrário do Paramiko, que exige uma configuração manual mais detalhada para cada dispositivo, o Netmiko simplifica esse processo ao fornecer interfaces prontas para diversos tipos de dispositivos de rede, como switches e roteadores de diferentes fabricantes. Isso torna a automação mais acessível e eficiente.

Realizamos um comparativo entre as ferramentas Paramiko e Netmiko para avaliar sua eficiência na coleta de informações de dispositivos de rede. Foram conduzidos dois testes principais: o primeiro buscou o número de série (serial number) dos dispositivos, enquanto o segundo coletou informações sobre a rede e os nomes das interfaces. Esses testes foram aplicados a três grupos distintos de equipamentos (Juniper, Extreme, Cisco), com o tempo de execução de cada ferramenta medido por meio da biblioteca time[8] do Python. Diferente de estudos anteriores, que se concentraram em uma única ferramenta, nossa análise foca na comparação direta do desempenho de ambas em um cenário variado de dispositivos e fabricantes.

É viável estabelecer uma comparação entre o NetBox e outras ferramentas, como o Zabbix[9]. Embora o Zabbix não tenha como foco principal o registro de equipamentos, mas sim o monitoramento de sistemas, ele também disponibiliza a funcionalidade de documentar os dispositivos que estão sob sua supervisão. Assim, é possível avaliar as capacidades de documentação de ambas as ferramentas, levando em conta suas abordagens e funcionalidades específicas no gerenciamento de ativos de rede.

4 Ambiente de Uso

A Rede Nacional de Ensino e Pesquisa (RNP)[10] oferece acesso à internet para organizações de pesquisa,

como universidades e hospitais. No Rio Grande do Sul, a RNP, por meio da Rede Tchê[11], conecta centros de pesquisa em todo o estado, tanto na capital quanto no interior. Foram testados 49 dispositivos Juniper, 20 da Extreme, 14 da Cisco e 8 da Huawei, entre switches e roteadores, permitindo uma análise abrangente da eficácia da rede.

A escolha da configuração de hardware foi feita para garantir que o ambiente tivesse recursos suficientes para a execução eficiente dos testes e análise de desempenho das ferramentas utilizadas. A máquina virtual Ubuntu [12] com 12 GB de memória RAM e a CPU Intel(R) Xeon(R) E5-2670 0 com clock de 2.60 GHz foram selecionadas para garantir que o processamento das requisições e a execução simultânea das ferramentas (como Paramiko, Netmiko e NetBox) não fossem limitadas por recursos insuficientes, especialmente em testes que envolvem múltiplos dispositivos simultaneamente.

As ferramentas utilizadas não possuem requisitos mínimos de hardware extremamente rigorosos, mas, dado o volume de dispositivos e a necessidade de processamento em paralelo, uma configuração mais robusta foi adotada para evitar gargalos e garantir a precisão na análise de desempenho. O uso de uma máquina com 12 GB de RAM e uma CPU com múltiplos núcleos permite lidar com a carga de trabalho sem comprometimentos significativos no tempo de resposta.

5 Resultados

Nos testes realizados, o Paramiko apresentou um desempenho levemente superior em ambas as avaliações. A Figura 1 ilustra os resultados obtidos, que incluem a coleta de informações sobre interfaces e a obtenção de números de série dos dispositivos, evidenciando a eficiência da biblioteca em tarefas de automação de redes.

Como pode ser observado, o Paramiko apresentou um desempenho superior em quase todos os testes realizados. A única exceção foi no teste de busca de informações com dispositivos Juniper, onde o Netmiko foi ligeiramente mais rápido. Nos demais testes, o Paramiko ficou à frente, variando de 1 a 20 segundos, sendo que no teste de interface com dispositivos Juniper, a diferença foi de 20 segundos a favor do Paramiko.

A primeira etapa do *script* consiste em uma função projetada para retornar uma lista de comandos a serem executados nos dispositivos, de acordo com o fabricante. Cada comando é ajustado conforme o objetivo e o fabricante do equipamento, assegurando a execução precisa das instruções específicas para cada tipo de dispositivo.

Para melhorar a organização e a legibilidade, a função foi dividida em partes menores, permitindo a modularidade do código e a simplificação do tratamento de comandos variados. O principal desafio foi lidar com a diversidade de fabricantes e os diferentes formatos de saída associados, um ponto que a implementação aborda com sucesso.

Os detalhes completos dessa implementação, incluindo o tratamento modular para fabricantes como Juniper, Cisco, Huawei e Extreme, estão disponíveis no repositório GitHub do projeto Automação <https://github.com/Alexandre-B-Cauduro/automacao>. No arquivo correspondente, a estrutura do código segue o princípio DRY (Don't Repeat Yourself) [13], evitando repetições desnecessárias e garantindo maior eficiência na execução das funções.

Na próxima etapa, o *script* realiza a conexão SSH com os dispositivos utilizando o Paramiko. Para isso, são necessários cinco parâmetros: "hostname", "username" e "password" para autenticação, e o fabricante e a escolha anterior para determinar quais comandos serão executados. Os três primeiros garantem o login correto, enquanto os dois últimos guiam o script na execução dos comandos apropriados para cada dispositivo.

O funcionamento do Paramiko é sustentado por três objetos principais em seu código. O primeiro é o *stdin*, que é responsável por enviar dados de entrada para o dispositivo. O segundo é o *stdout*, que permite capturar a saída dos comandos executados pelo *stdin*. Por fim, o *stderr* é utilizado para registrar quaisquer erros que possam ocorrer durante a execução dos comandos. Essa estrutura torna a comunicação com os dispositivos mais eficiente, facilitando a manipulação de entradas e saídas de forma eficaz.

O desafio da automatização vai além da simples execução de comandos. Por exemplo, ao buscar o número de série (serial number) em um equipamento Juniper, o comando `show chassis hardware` retorna uma saída extensa, repleta de informações adicionais. Identificar e extraer o número de série desejado em meio a esse volume de dados requer um tratamento específico da saída. Para abordar essa questão, o código foi desenvolvido para filtrar a saída do comando, isolando apenas as informações relevantes, como o número de série. A lógica implementada garante que diferentes formatos de números de série sejam corretamente identificados e extraídos, eliminando todo o conteúdo desnecessário.

Essa capacidade de manipulação de dados é crucial para a eficácia da automação proposta, que, embora enfrente desafios como a complexidade das saídas de comandos, demonstra a robustez do NetBox como

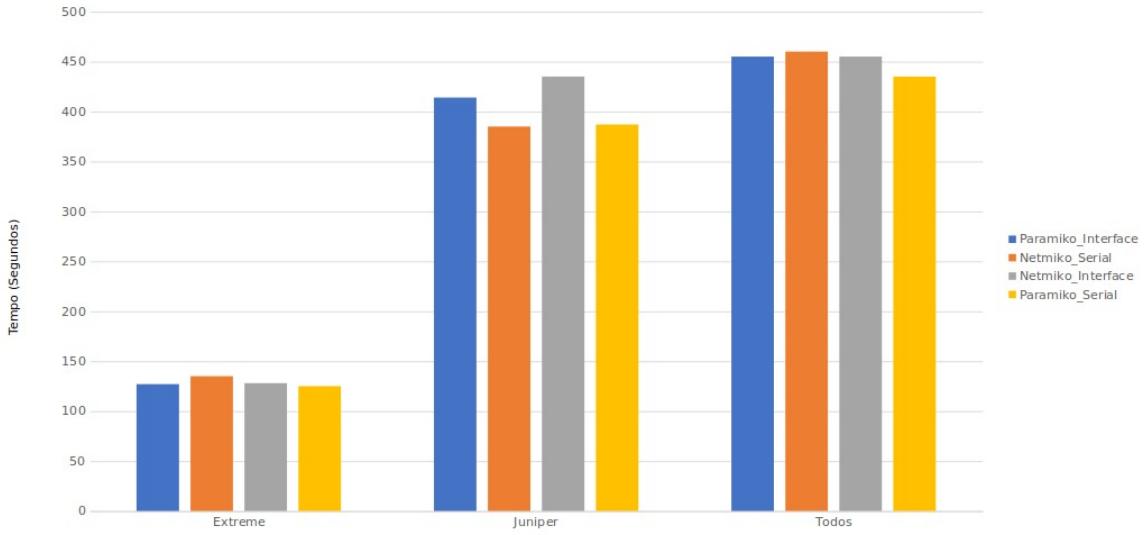


Figura 1. Tempo médio de execução em segundos para a coleta de informações de dispositivos de rede utilizando as bibliotecas Paramiko e Netmiko. O gráfico destaca que o Paramiko apresenta tempos de resposta inferiores, especialmente na coleta de números de série, evidenciando sua maior eficiência em relação ao Netmiko.

uma ferramenta de gerenciamento de redes. O NetBox, sendo uma aplicação bem documentada e de código aberto, se destaca como uma alternativa viável no mercado, apesar das inconsistências que podem surgir na API e no PyNetBox. Essas inconsistências, embora possam impactar a experiência do usuário, não diminuem a eficácia do NetBox, especialmente quando combinado com soluções de automação que otimizam a coleta e atualização de dados.

Dessa forma, o código retornará apenas o número de série, que estará pronto para ser enviado ao NetBox. A próxima etapa consiste em automatizar o envio dessas informações para a aplicação. A utilização do PyNetBox possibilita que esses dados sejam transmitidos de maneira análoga ao processo de coleta de informações nos switches e roteadores. Isso garante uma integração contínua e eficiente entre os dispositivos e o NetBox, facilitando a atualização e o gerenciamento das informações na rede.

Há duas principais diferenças a serem destacadas. A primeira é que a conexão com o NetBox é feita por meio de um token gerado pela própria aplicação. A segunda é que o script converte as informações em formato JSON[14] para que a aplicação possa processá-las corretamente. Essas características tornam o uso da biblioteca PyNetBox praticamente indispensável ao

interagir com o NetBox, facilitando a automação e a atualização de dados na aplicação.

Se o objetivo for apenas atualizar uma classe específica dentro do NetBox, não é necessário implementar uma solução complexa. O PyNetBox se destaca nesse processo pela sua capacidade de carregar várias informações simultaneamente de maneira eficiente. Essa biblioteca é especialmente útil para lidar com múltiplas atualizações de dados de forma automatizada e rápida. A implementação detalhada dessa funcionalidade, incluindo exemplos práticos, está documentada no repositório GitHub do projeto Automação.

A solução desenvolvida é composta por um núcleo de código que estabelece uma conexão com o NetBox utilizando um token de autenticação, possibilitando a atualização de informações específicas, como o número de série (serial number) dos dispositivos. Além disso, o código foi projetado para permitir a execução sequencial de múltiplas atualizações, otimizando o processo e reduzindo significativamente o tempo necessário para manter os dados da rede atualizados.

Os resultados obtidos demonstram a eficácia da automação proposta, especialmente na coleta e atualização de informações sobre dispositivos de rede. O Paramiko apresentou um desempenho superior em diversos cenários, graças à sua rapidez e flexibilidade

na execução de comandos SSH, enquanto o PyNetBox garantiu uma integração contínua e eficiente com o NetBox. Essa abordagem automatizada resultou em uma redução significativa no tempo necessário para atualizar os dados da rede, minimizando erros manuais e promovendo uma gestão centralizada mais eficaz. Esses achados reforçam a viabilidade da solução, posicionando-a como uma alternativa robusta para o gerenciamento automatizado de infraestruturas de rede complexas.

6 Discussão

Algumas questões precisam ser destacadas: o Paramiko se mostrou mais rápido que o Netmiko, o que é compreensível, já que o Netmiko é construído sobre o Paramiko e adiciona uma camada extra de complexidade. No entanto, ambos se mostraram bastante flexíveis quando se trata de integração com a rede, oferecendo soluções eficazes para automação de dispositivos e coleta de informações, cada um com suas vantagens específicas.

O NetBox organiza as informações em categorias que definem tanto o tipo de dado que pode ser armazenado quanto o local de armazenamento. Ele permite o registro detalhado de diversos tipos de equipamentos, abrangendo praticamente todos os dispositivos relacionados a uma rede, como racks, servidores, roteadores, switches e muito mais. Essa estrutura torna o NetBox uma base de dados centralizada e eficiente para a gestão e documentação dos ativos de rede.

Além disso, o NetBox é uma aplicação bem documentada e de código aberto, o que a torna uma opção bastante atrativa em comparação com outras soluções disponíveis no mercado. No entanto, é importante notar que a API pode apresentar algumas inconsistências, que também se refletem no PyNetBox, uma vez que este último atua como uma abstração da API REST. Embora essas inconsistências possam afetar a experiência do usuário, elas não comprometem a robustez e a eficácia do NetBox como uma ferramenta para o gerenciamento de redes.

Apesar dessas dificuldades, é importante destacar que o NetBox se mostra extremamente completo e versátil em termos de funcionalidades e diversidade de informações que pode gerenciar. As inconsistências na API, embora indesejadas, são compreensíveis dado o tamanho e a complexidade da aplicação. Por ser uma ferramenta robusta, que abrange uma vasta gama de dispositivos e categorias de rede, é natural que surjam essas pequenas incoerências no processo de integração. No entanto, esses problemas não comprometem o valor da aplicação, que continua sendo uma solução poderosa para gerenciamento e documentação de redes.

7 Considerações Finais

A automatização do processo de alimentação do NetBox, como apresentado neste trabalho, demonstra uma significativa melhoria na gestão e documentação de redes complexas. Ao utilizar ferramentas como Paramiko e PyNetBox, foi possível simplificar a coleta e a inserção de dados, reduzindo o tempo necessário para atualizar informações sobre ativos de rede e minimizando o risco de erros humanos. A importância dessa automatização reside na sua capacidade de aumentar a eficiência operacional, permitindo que administradores de rede dediquem mais tempo a atividades estratégicas e menos tempo em tarefas repetitivas. Além disso, a integração contínua de dados por meio da API do NetBox proporciona um ambiente de gestão centralizado e atualizado, essencial para a manutenção de redes dinâmicas e em constante evolução. Essa abordagem destaca o papel da automação na modernização das infraestruturas de TI, garantindo uma operação mais ágil e confiável.

O Paramiko demonstrou ser uma ferramenta flexível, capaz de se adaptar a diversos dispositivos e sistemas operacionais. Além disso, mostrou-se mais rápido em comparação com outras ferramentas alternativas. Outro aspecto positivo é sua extensa documentação, que facilita o uso e a integração da ferramenta em diferentes cenários de automação e gerenciamento de redes. Essa documentação oferece suporte abrangente para desenvolvedores e administradores, tornando o Paramiko uma escolha valiosa para tarefas de automação.

O NetBox é extremamente robusto e capaz de suportar uma vasta quantidade de equipamentos. Foi uma grata surpresa descobrir que todos os dispositivos registrados já estavam catalogados de forma adequada. Por exemplo, ao registrar um dispositivo como o X460-G2-24x-10GE4, não foi necessário especificar ou explicar à aplicação quais eram as capacidades desse equipamento, pois todas essas informações já estavam organizadas de maneira eficiente. Isso simplifica significativamente o processo, deixando apenas a tarefa de criar um catálogo de equipamentos como etapa adicional.

Para trabalhos futuros, sugere-se uma pesquisa mais aprofundada, incluindo a realização de testes em uma rede maior e com uma variedade ampliada de fabricantes de dispositivos. Além disso, seria interessante integrar ao processo a funcionalidade de monitoramento do consumo de CPU e RAM dos dispositivos. Essa abordagem não apenas complementaria a coleta de informações básicas, mas também permitiria uma análise mais abrangente do desempenho dos equipamentos e da saúde da infraestrutura de rede, contribuindo para uma gestão mais eficiente e proativa.

Referências

- 1 Shaw, K. Net-Box is ambitious, but audience is unclear. *Network World*, IDG Communications, Inc., p. 34–34, 2005.
- 2 Zadka, M.; Zadka, M. Paramiko. *DevOps in Python: Infrastructure as Python*, Springer, p. 111–119, 2019.
- 3 Mutiara, D. A.; Isnaini, K. N.; Suhartono, D. Network Programmability for Network Issue Using Paramiko Library. *Jurnal Teknik Informatika (JUTIF) DOI: https://doi.org*, v. 10, n. 1, 2023.
- 4 Novák, M.; Airflow, A. GraphQL API Facade for IT Inventory Management Tool.
- 5 Perkasa, M. I.; Setiawan, E. B. Pembangunan Web Service Data Masyarakat Menggunakan REST API dengan Access Token. *Ultima Computing: Jurnal Sistem Komputer*, v. 10, n. 1, p. 19–26, 2018.
- 6 Walther, D.; Jovicic, D. *SSoT Based Network Service Deployment*. 2023. Tese (Doutorado) – OST Ostschweizer Fachhochschule.
- 7 Nugroho, K.; Abrariansyah, A. D.; Ikhwan, S. Perbandingan Kinerja Library Paramiko dan Netmiko Dalam Proses Otomasi Jaringan. *InfoTekJar J. Nas. Inform. dan Teknol. Jar*, v. 5, n. 1, p. 1–8, 2020.
- 8 Borges, L. E. *Python para desenvolvedores: aborda Python 3.3*. Novatec Editora, 2014.
- 9 Olups, R. *Zabbix Network Monitoring*. Packt Publishing Ltd, 2016.
- 10 Simões, N. et al. A Ciberinfraestrutura no Brasil—a contribuição da RNP. *Position Paper. 18p*, 2016.
- 11 Virti, É. S. et al. Controlando Tráfego Peer-to-Peer. *Escola Regional de Redes de Computadores (2.: 2004: Canoas). Anais. Porto Alegre: SBC*, 2004, 2004.
- 12 Tabassum, M.; Mathew, K. Software evolution analysis of linux (Ubuntu) OS. In: IEEE. 2014 International Conference on Computational Science and Technology (ICCST). 2014. P. 1–7.
- 13 Wilson, G. et al. Best practices for scientific computing. *PLoS biology*, Public Library of Science San Francisco, USA, v. 12, n. 1, e1001745, 2014.
- 14 Nurseitov, N. et al. Comparison of JSON and XML data interchange formats: a case study. *Caine*, Citeseer, v. 9, p. 157–162, 2009.