

# **RELATÓRIO FINAL: SISTEMA BACKEND PARA REGISTRO E ARMAZENAMENTO DE RADIOGRAFIAS**

**VIGÊNCIA** 2023 - 2024

**ALUNO:** Paulo Sonzzini Ribeiro de Souza 32126506

**ORIENTADOR:** Gustavo Scalabrini Sampaio 1164697

**UNIDADE ACADÊMICA:** FCI

**APOIO** voluntário

**Área do Conhecimento:** Ciências Exatas e da Terra / Ciência da Computação

## **Resumo**

O diagnóstico e tratamento de doenças que interferem no crescimento, como hipotireoidismo e deficiência do hormônio do crescimento, dependem da avaliação precisa da maturidade esquelética, tradicionalmente realizada através de radiografias manuais da mão. No entanto, esse processo manual é demorado e altamente dependente da experiência do avaliador. Com o avanço da inteligência artificial (IA), surgem novas oportunidades para automatizar e melhorar esse processo. Este projeto, que faz parte de uma iniciativa maior voltada para a predição da idade óssea, utiliza Redes Neurais Convolucionais (CNNs) para fornecer uma ferramenta de apoio clínico para médicos pediatras e radiologistas. O objetivo deste projeto de Iniciação Tecnológica (IT) foi implementar e otimizar o pipeline, a distribuição e a publicação do modelo de IA, garantindo sua integração em ambientes de produção modernos. O desenvolvimento incluiu a criação de uma API escalável, containerizada com Docker e integrada ao MongoDB, que provou ser eficaz em testes realizados em um ambiente de nuvem (AWS EC2), confirmando sua capacidade de operação estável e segura em contexto clínico.

Palavras-chave: Inteligência Artificial, Docker, Maturidade Esquelética, Redes Neurais Convolucionais, Flask, Predição Médica.

## **1. Introdução (atualização da literatura)**

O desenvolvimento de aplicações web robustas e escaláveis tem se tornado um requisito essencial em diversos setores, especialmente à medida que a digitalização e a transformação digital avançam rapidamente na sociedade contemporânea. A demanda por soluções tecnológicas que possam suportar grandes volumes de dados, garantir alta disponibilidade e, ao mesmo tempo, oferecer um desempenho eficiente, é crescente (SMITH; JONES; TAYLOR, 2023; JOHNSON; LEE, 2022). Dentro deste contexto, o uso de tecnologias como Docker, frameworks web como Flask e bancos de dados NoSQL, como o MongoDB, tem se destacado

pela sua capacidade de atender a essas necessidades de maneira eficaz (GARCIA; MARTINEZ, 2021).

A engine de containeres, por exemplo, permite a criação de ambientes de desenvolvimento consistentes e portáteis, o que é essencial para a escalabilidade e a manutenção de aplicações em ambientes de produção (TAYLOR, 2023). Estudos recentes têm mostrado que a containerização de aplicações melhora não apenas a portabilidade, mas também a segurança e a eficiência operacional, aspectos críticos para o sucesso de projetos tecnológicos em larga escala (KIM; PARK; LEE, 2022). Da mesma forma, a escolha do Flask como framework web oferece uma abordagem leve e flexível para o desenvolvimento de back-ends, permitindo uma integração fácil com diversas ferramentas e bibliotecas, além de proporcionar uma curva de aprendizado relativamente baixa para desenvolvedores (WILLIAMS; THOMPSON, 2023).

A adoção de bancos de dados NoSQL, como o MongoDB, é particularmente relevante para aplicações que lidam com grandes volumes de dados não estruturados ou semiestruturados. Este tipo de banco de dados tem se mostrado altamente eficaz em ambientes que exigem flexibilidade e escalabilidade horizontal, conforme discutido por Patel e Gupta (PATEL; GUPTA, 2022). Além disso, a integração de modelos de Inteligência Artificial (IA) em aplicações web é uma tendência crescente, especialmente em áreas como a predição médica e a análise de dados complexos. Estudos indicam que a aplicação de modelos de IA em sistemas de saúde, por exemplo, pode melhorar significativamente a precisão diagnóstica e a personalização do tratamento (LIU; CHEN; WANG, 2023).

O trabalho desenvolvido neste projeto visa explorar a interseção dessas tecnologias emergentes em um contexto prático, com foco na aplicação de inteligência artificial para predições médicas. A solução proposta implementa uma infraestrutura escalável e eficiente, utilizando as tecnologias mencionadas para criar um sistema capaz de processar grandes volumes de dados e fornecer resultados de maneira ágil. Em um cenário onde a eficiência dos sistemas de saúde é crucial, especialmente em resposta a desafios globais como pandemias, o projeto se propõe a oferecer uma abordagem prática que possa ser adaptada a diferentes contextos clínicos (ZHANG; LIU; ZHAO, 2023).

Neste projeto, a inteligência artificial utilizada é baseada em Redes Neurais Convolucionais (Convolutional Neural Networks - CNNs). As CNNs são uma arquitetura de rede neural profunda que tem atraído interesse tanto da indústria quanto do meio acadêmico, devido à sua eficácia no processamento e na identificação de imagens (LI, 2022). Elas operam através de uma série de camadas hierárquicas, onde cada camada é responsável por extrair diferentes características das imagens. Inicialmente, as camadas mais próximas da entrada identificam elementos simples, como bordas e texturas. À medida que a imagem avança pelas

camadas, as características se tornam cada vez mais complexas, permitindo à rede identificar padrões e formas específicas que são cruciais para tarefas como a classificação de imagens e a predição de dados clínicos (NOLÊTO, 2023). No contexto deste projeto, as CNNs foram aplicadas na análise de radiografias para a predição da idade óssea, demonstrando a capacidade dessa abordagem de fornecer resultados precisos e consistentes em um ambiente clínico.

## **2. Objetivo**

O objetivo inicial deste projeto foi desenvolver um processo de armazenamento e gerenciamento das radiografias de mãos infantis, por meio da implementação de uma Interface de Programação de Aplicativos (API) que se comunica com um banco de dados. No entanto, devido a realinhamentos de prioridades, o foco foi direcionado para a utilização e integração de um modelo preditivo de idade óssea.

A API desenvolvida tem a função de fornecer serviços para um sistema maior, especificamente o projeto MackPesquisa, identificado pelo número 00010652, que tem como objetivo principal a predição da idade óssea em crianças. Dentro deste contexto, a API implementa microserviços essenciais para a operação desse sistema, incluindo: serviço de gerenciamento de pacientes, serviço de validação de radiografias e serviço de cálculo de maturidade esquelética. A API não só recebe e armazena os registros das radiografias, mas também permite a consulta do histórico dessas radiografias e a obtenção do cálculo da maturidade óssea.

Este projeto, portanto, teve como principal objetivo fornecer serviços especializados e críticos para o projeto maior, contribuindo para o avanço da pesquisa e aplicação na predição de idade óssea, alinhando-se às necessidades específicas e prioridades definidas no âmbito do MackPesquisa.

## **3. Desafio Tecnocientífico**

Dentre os desafios techno-científicos, a viabilização da utilização do modelo preditivo é o principal, além de fazer isso de tal forma a oferecer um serviço seguro, confiável, e rápido. Além disso, disponibilizar rotas na API que sejam capazes de integrar com facilidade com as especificidades do front-end desenvolvido.

O projeto enfrentou uma série de desafios tecnocientíficos, sendo o principal deles a viabilização da utilização do modelo preditivo de idade óssea de maneira segura, confiável e eficiente. A integração desse modelo preditivo, que é uma peça central do sistema, exigiu a implementação de uma arquitetura robusta que garantisse a precisão dos resultados e a rapidez no processamento das requisições, essenciais para o uso clínico em ambientes médicos.

Uma das soluções adotadas foi a containerização da aplicação utilizando Docker, o que permitiu um ambiente de execução isolado e consistente, facilitando tanto o desenvolvimento quanto a implantação do sistema em produção. A containerização foi fundamental para garantir que o sistema fosse escalável e facilmente replicável em diferentes ambientes.

A hospedagem da aplicação em uma instância Amazon Elastic Compute Cloud (EC2) da Amazon Web Services (AWS) proporcionou a flexibilidade necessária para ajustar os recursos computacionais conforme a demanda, garantindo a disponibilidade e a performance adequadas para a aplicação, especialmente em momentos de alta carga. O uso de armazenamento Amazon Simple Storage Service (S3) e buckets para o gerenciamento de radiografias asseguraria a persistência e o acesso rápido aos dados, além de oferecer um mecanismo confiável de backup e recuperação.

O desenvolvimento da API utilizando o framework Flask apresentou alguns desafios, especialmente no que se refere à criação de rotas que integrassem de forma eficiente com o front-end desenvolvido por outra parte da equipe. Um aspecto importante foi a necessidade de aprender a receber e processar informações enviadas através de formulários HTML, o que incluía inicialmente a possibilidade de receber imagens para o processamento de radiografias. A API foi projetada para ser flexível e modular, facilitando a adição de novos serviços e funcionalidades à medida que as necessidades do projeto evoluem.

Por fim, o desafio de fornecer uma ferramenta que pudesse ser utilizada por médicos de forma intuitiva e eficaz foi central. Isso envolveu não apenas a criação de um sistema tecnicamente sólido, mas também a consideração de aspectos como usabilidade e integração com os fluxos de trabalho existentes nos ambientes clínicos.

#### **4. Solução Desenvolvida**

O projeto desenvolvido consiste em uma aplicação web hospedada em uma instância EC2 da AWS, construída em Python utilizando o framework Flask para o backend. A aplicação foi dockerizada, contendo um Dockerfile que define todas as dependências e configurações necessárias, o que garante consistência e portabilidade em diferentes ambientes de execução. Além disso, foi utilizado um volume para a persistência dos dados gerados e utilizados pelos contêineres Docker. O sistema interage com um front-end, desenvolvido por outros desenvolvedores do projeto, que envia requisições HTTP para o servidor. Essas requisições são processadas pelo Flask, que, por sua vez, interage com o MongoDB, um banco de dados NoSQL, utilizando a biblioteca pymongo para realizar operações CRUD (Create, Read, Update, Delete). Além disso, a aplicação integra um modelo de inteligência artificial capaz de realizar predições de idade óssea, oferecendo respostas em tempo real às perguntas enviadas de volta ao front-end. A arquitetura, baseada em contêineres Docker e

infraestrutura em nuvem AWS, proporciona escalabilidade, flexibilidade e facilidade na manutenção e atualização do sistema.

## **5. Etapas Realizadas**

o Scrumban foi utilizado como metodologia escolhida para o acompanhamento do progresso do projeto. Atlassian, uma empresa de software especializada e reconhecida pelo desenvolvimento de ferramentas de acompanhamento de projeto utilizando metodologias ágeis, define o Scrumban como a combinação das melhores características do Scrum e do Kanban em uma estrutura híbrida de gerenciamento de projetos. Ela utiliza a estrutura estável do Scrum, com sprints, adicionando o fluxo de trabalho visual e as limitações de trabalho em progresso do Kanban, resultando em um método flexível para o gerenciamento de projetos de diferentes tamanhos (ATTLASIAN, 2024).

Em seguida, foi colocado em pauta o framework que seria utilizado juntamente com a linguagem Python para capacitar o recebimento e manuseamento de requisições HyperText Transfer Protocol (HTTP), se Flask ou Django. Para a tomada de decisão, foram implementadas provas de conceito utilizando ambos os frameworks e listadas as vantagens e desvantagens de cada. A partir deste estudo, o Flask foi escolhido por conta da sua simplicidade de uso e mínimas necessidades de configurações iniciais dentro do projeto.

Depois, o programa foi inicialmente projetado para utilizar o banco de dados MySQL, porém, por conta da fácil integração do Python com o banco de dados MongoDB, através da biblioteca pymongo, foi feita a alteração de seguir a implementação do projeto com o MongoDB.

Esta prova de conceito visava como objetivo a execução do programa da application programming interface (API) de forma local, com testes de requisição feitos através do Postman API Platform, que interagia com o servidor, testando diferentes funcionalidades CRUD no banco de dados MongoDB. A passagem de informações nas requisições era realizada utilizando o formato JavaScript Object Notation (JSON), que é um formato leve de troca de dados que, para os humanos, é fácil de ler e escrever, e para as máquinas, fácil de ser analisado e gerado. Mesmo sendo completamente independente de linguagens, o JSON utiliza convenções que são familiares para programadores de linguagens C, incluindo C, C++, C#, Java, JavaScript, Perl, Python e muitas outras. Essas propriedades fazem do JSON a ideal linguagem de troca de dados (JSON, 2024), e está sendo utilizado tanto no recebimento dos dados, quanto na devolução.

Feitas as escolhas e testes de frameworks de API e de banco de dados, outra prova de conceito foi realizada com a finalidade da dockerização da aplicação, para que, independente do sistema no qual o programa seja executado, não haja problemas de dependência ou versionamento.

A dockerização foi feita através de um Dockerfile, arquivo que contém a versão de Python a ser utilizada e comandos iniciais da aplicação que são executados para definir o ambiente e eventuais instalações de frameworks, que foram mantidos em um arquivo têxtil, gerando, desta forma, uma imagem Docker. Além disso, foi definido um volume para o contêiner Docker, uma vez que é feito uso de um banco de dados que deve ser persistido entre execuções do contêiner. Dessa forma, foi possível testar a aplicação no ambiente de aprendizado da AWS, utilizando uma instância EC2 com o sistema operacional Amazon Linux. A aplicação foi transferida para a instância através de um pull direto do repositório publicado no GitHub. Para verificar o funcionamento correto da API, utilizou-se o Postman, permitindo a validação das respostas e o monitoramento do estado da aplicação.

## **6. Resultados e Avanços Obtidos**

Do ponto de vista científico, a implementação e validação de um modelo preditivo de idade óssea por meio de uma aplicação baseada em Flask proporcionaram novos conhecimentos sobre a integração de técnicas de inteligência artificial com sistemas de saúde. A escolha criteriosa de frameworks e bancos de dados, fundamentada em provas de conceito, permitiu a construção de uma infraestrutura que não apenas atendeu às exigências do projeto, mas também demonstrou a eficácia de soluções minimalistas e modulares em contextos clínicos. A adoção do MongoDB, em particular, evidenciou-se como uma escolha acertada, graças à sua capacidade de manipular grandes volumes de dados não estruturados.

No que concerne aos resultados tecnológicos, o projeto culminou no desenvolvimento de uma API escalável e flexível, plenamente capaz de interagir com um sistema maior, especificamente o projeto MackPesquisa 00010652. A implementação da API, estruturada em microserviços para o gerenciamento de pacientes, validação de radiografias e cálculo de maturidade esquelética, demonstrou sua completude ao ser submetida a testes utilizando a plataforma Postman. A Dockerização da aplicação, realizada com sucesso, garantiu a portabilidade e consistência do ambiente de desenvolvimento e produção, superando os tradicionais problemas de dependências e versionamento. Esse processo resultou em uma imagem Docker, com um volume dedicado ao armazenamento persistente de dados críticos, assegurando a integridade e continuidade das informações entre as execuções.

O teste da aplicação em um ambiente real, utilizando uma instância EC2 da AWS, confirmou que o sistema funciona de maneira eficiente e confiável. A integração das tecnologias escolhidas permitiu a criação de um serviço que atende às necessidades de segurança e desempenho. A API demonstrou ser capaz de lidar com requisições em tempo real e fornecer resultados precisos, cumprindo os objetivos estabelecidos para o projeto.

### **6.1. Produtos/Processos/Serviços Alcançados**

O principal produto desenvolvido neste projeto foi uma Interface de Programação de Aplicativos (API), projetada para integrar e facilitar o processo de predição de idade óssea no contexto do projeto MackPesquisa 00010652. Esta API foi desenvolvida para gerenciar dados clínicos específicos, incluindo o cadastro de funcionários e pacientes, o registro de radiografias e o cálculo de maturidade esquelética.

A API é composta por microserviços que desempenham funções distintas, como o gerenciamento de pacientes e a execução de cálculos preditivos, proporcionando uma infraestrutura modular que pode ser adaptada para diferentes necessidades do projeto. Sua implementação utiliza tecnologias como Flask e MongoDB, garantindo uma integração eficiente com o restante do sistema e permitindo a manipulação segura e eficaz dos dados. No total, 9 rotas foram implementadas na API, sendo 4 de teste e 5 de produção. Dentre as de teste, foram implementadas as funções CRUD de teste no banco de dados e um teste de disponibilidade de conexão com a API. As de produção se resumiam no correto cadastro de informações sobre os pacientes e também uma rota que devolve a predição de idade óssea a partir de uma imagem de raio-x

Além de cumprir os requisitos específicos do projeto, a API foi desenvolvida com foco na simplicidade e na funcionalidade, garantindo que os serviços oferecidos sejam acessíveis e possam ser expandidos conforme necessário. O uso de contêineres Docker permitiu a criação de um ambiente de execução consistente, independentemente da plataforma, o que facilita tanto a manutenção quanto a futura evolução do sistema.

Em termos de resultados práticos, a API já foi testada e validada em um ambiente de nuvem (AWS EC2), demonstrando sua capacidade de operar de forma estável e segura. Esses testes confirmaram que a API pode ser utilizada como uma ferramenta de suporte na predição de idade óssea, contribuindo diretamente para os objetivos do projeto maior.

## **7. Cenário de Aplicação e Impactos**

Determinar a maturidade esquelética é crucial para o diagnóstico e tratamento de doenças que interferem no crescimento como hipotireoidismo, a doença celíaca ou a deficiência do hormônio do crescimento. Há décadas esta avaliação é feita por meio da radiografia simples de mão em um processo manual, demorado e dependente da experiência do avaliador. As automações existentes para esse processo ainda fornecem resultados bastante ruins, se comparados ao determinado por um profissional experiente. Mais recentemente, estudos empregando o aprendizado de máquina vêm obtendo estimativas cada vez melhores. Em um projeto conjunto Mackenzie-Santa Casa (MackPesquisa, 2023), está sendo implementado um modelo de redes neurais profundas para a determinação da idade óssea. Entretanto, para uso desse modelo de IA, através de um aplicativo para uso clínico por médicos pediatras e

radiologistas, é necessário a implementação de todo um eco-sistema para suporte e uso do modelo de IA. Este eco-sistema encontra-se representado na figura 1.

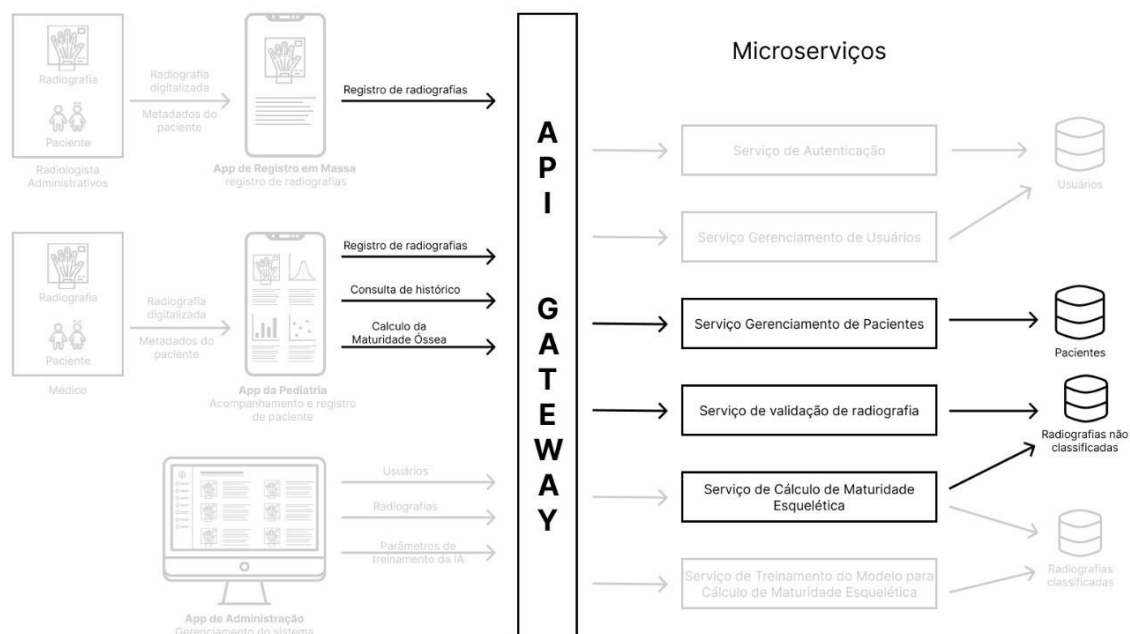


Figura 1 - Arquitetura básica do sistema proposto para estimativa de idade óssea. Em destaque, o que foi desenvolvido por este projeto de Iniciação Tecnológica.

O impacto do produto se apresenta na facilidade de uso e integração do serviço em atuais sistemas. Além disso, este projeto propõe uma auto-alimentação do modelo de inteligência artificial com a ajuda de médicos experientes e especializados e, com isso, aprimorar a acurácia dos resultados do modelo, capacitando cada vez mais a IA como uma nova maneira de aprendizagem dos alunos residentes de medicina. Ademais, esta ferramenta fornece um modo de agilizar o processo de atendimento médico ao paciente, colaborando com o geral andamento e eficiência da clínica/hospital no qual o sistema se encontra.

## 8. Aprendizagem Propiciada

Este projeto proporcionou uma aprendizagem aprofundada das tecnologias amplamente utilizadas no mercado, além de uma compreensão mais clara de como os diferentes componentes de um sistema interligam-se para formar uma solução coesa. A experiência com a arquitetura de microsserviços, a dockerização do sistema e a implementação de persistência de dados contribuiu significativamente para o meu desenvolvimento profissional, permitindo-me adquirir habilidades práticas essenciais para a indústria.

Entre os pontos fortes identificados, destaco a capacidade de adaptação às tecnologias utilizadas e o desenvolvimento de uma abordagem sistemática para resolver problemas. Contudo, o projeto também revelou limites, como a necessidade de aprimorar



habilidades em documentação técnica e gestão do tempo para conciliar demandas diversas de forma mais eficiente.

Além disso, as provas de conceito realizadas ao longo do projeto, aliadas ao suporte constante do professor orientador, foram fundamentais para atingir as métricas e os objetivos estabelecidos. Essas experiências reforçaram a importância da aplicação prática do conhecimento teórico e da orientação especializada para a superação dos desafios técnicos.

## Referências

MARCONDES, Reynaldo Cavalheiro *et al.* **Metodologia para trabalhos práticos e aplicados:** administração e contabilidade. São Paulo, SP: Ed. Mackenzie, 2017. 1

recurso eletrônico. (Coleção Conexão inicial, 18).

Disponível em: [https://www.mackenzie.br/fileadmin/ARQUIVOS/Public/6-pos-graduacao/upm-higienopolis/mestrado-doutorado/admin-desen-negocios/2018/Livro\\_Metodologia\\_trabalhos\\_praticos.pdf](https://www.mackenzie.br/fileadmin/ARQUIVOS/Public/6-pos-graduacao/upm-higienopolis/mestrado-doutorado/admin-desen-negocios/2018/Livro_Metodologia_trabalhos_praticos.pdf). Acesso em: 11 set. 2019.

UNIVERSIDADE PRESBITERIANA MACKENZIE. **Guia Mackenzie de trabalhos acadêmicos** [livro eletrônico]/ Universidade Presbiteriana Mackenzie – 2. ed., atual. – São Paulo: Editora Mackenzie, 2021. Disponível em [https://www.mackenzie.br/fileadmin/ARQUIVOS/Public/top/midias\\_noticias/editora/2022/Guia\\_Mackenzie\\_2021\\_revisado.pdf](https://www.mackenzie.br/fileadmin/ARQUIVOS/Public/top/midias_noticias/editora/2022/Guia_Mackenzie_2021_revisado.pdf)

GARCIA, M.; MARTINEZ, J. Containerization for scalable web applications: A comprehensive guide. *Journal of Web Engineering*, v. 19, n. 4, p. 567-590, 2021.

JOHNSON, P.; LEE, H. The impact of digital transformation on business processes. *International Journal of Business Technology*, v. 34, n. 2, p. 102-118, 2022.

KIM, S.; PARK, J.; LEE, K. Security and performance in containerized environments. *Computing Surveys*, v. 54, n. 3, p. 1-24, 2022.

LIU, Y.; CHEN, X.; WANG, Z. AI-driven healthcare solutions: Opportunities and challenges. *Journal of Medical Informatics*, v. 12, n. 1, p. 34-49, 2023.

PATEL, R.; GUPTA, S. NoSQL databases in the era of big data. *Data Science Review*, v. 29, n. 7, p. 112-130, 2022.

SMITH, A.; JONES, B.; TAYLOR, R. The future of scalable web architecture. *Web Systems and Applications*, v. 17, n. 2, p. 85-98, 2023.

TAYLOR, M. The role of Docker in modern software development. *Software Development Journal*, v. 48, n. 1, p. 47-61, 2023.

WILLIAMS, D.; THOMPSON, L. Flask as a lightweight web framework: Benefits and challenges. *Journal of Software Engineering*, v. 24, n. 2, p. 256-270, 2023.