Associação Propagadora Esdeva
Centro Universitário Academia – UniAcademia
Curso de Engenharia de Software
Trabalho de Iniciação Científica – Artigo

Uma plataforma para coleta e análise de dados do GitHub

Cláudio Oliveira[1], Natan Fernandes[1]
Centro Universitário Academia, Juiz de Fora, MG

Tassio Ferenzini Martins Sirqueira[2]
Centro Universitário Academia, Juiz de Fora, MG

Linha de Pesquisa: Engenharia de Software

RESUMO

Os repositórios de código-fonte são fontes ricas da história da evolução de projetos de software e podem ser acessados e estudados abertamente. Quando queremos estudar algum fenômeno dentro da engenharia de software, precisamos de muitos dados históricos, e os projetos de código aberto são uma excelente oportunidade para explorar esse universo. Com a análise de dados históricos contidos em repositórios de código-fonte, é possível detectar e identificar padrões que podem explicar a evolução dos sistemas de software. Neste trabalho, desenvolvemos uma ferramenta para coletar, analisar e exportar dados de repositórios de código-fonte. Os objetivos da ferramenta são consultar dados diretamente do GitHub via API e apresentá-los por meio de uma interface clara e objetiva para pesquisadores e gerentes de projeto. Um exemplo passo a passo de como usar a ferramenta funciona será apresentado para demonstração. Considerando o repositório de código-fonte como base histórica, através da mineração de dados, é possível extrair informações úteis que auxiliam no processo de manutenção e evolução de software, consequentemente seu gerenciamento ao longo do ciclo de vida.

Palavras-chave: Código fonte. GitHub. Mineração de Repositórios.

ABSTRACT

Source code repositories are rich sources of the evolution history of software projects and can be openly accessed and studied. When we want to study some phenomenon within software engineering, we need much historical data, and open-source projects are an excellent opportunity to explore this universe. With the analysis of historical data contained in source code repositories, it is possible to detect and identify

[1] Discente do Curso de Engenharia de Software do Centro Universitário Academia – UniAcademia.
[2] Docente do Curso de Engenharia de Software do Centro Universitário Academia – UniAcademia. Orientador.

patterns that can explain the evolution of software systems. In this work, we developed a tool to collect, analyze and export data from source code repositories. The goals of the tool are to query data directly from GitHub via API and present it through a clear and objective interface to researchers and project managers. A step-by-step example of how to use the tool will be presented to demonstrate how the platform works. Considering the source code repository as a historical base, through data mining, it is possible to extract useful information that helps in the process of maintenance and evolution of software, consequently its management throughout the life cycle.

Keywords: Source code. GitHub. Repository Mining.

1 INTRODUÇÃO

A engenharia de software (ES) pode utilizar informações armazenadas nos repositórios de código fonte para se entender padrões de desenvolvimento, comportamento de programadores, bem como a dinâmica das equipes que colaboram com os projetos. Uma grande quantidade de dados é produzida durante o desenvolvimento de software e os repositórios de código fonte são uma fonte rica de informação.

A mineração de dados na engenharia de software tem emergido, oferecendo formas de interpretar a quantidade abundante de dados e, consequentemente, auxiliar na resolução de diversos problemas que envolvem o desenvolvimento de software (HASSAN e XIE, 2010). Conforme Junior e Nassif (2013), apesar do grande volume de dados, a descoberta de indicadores em projetos de software continuam a ser difíceis de gerenciar.

A ES trata, dentre outros temas, dos processos, métodos, técnicas e ferramentas utilizados no desenvolvimento de software (PRESSMAN, 2016). Por meio da Engenharia de Software é possível aumentar a confiança, a facilidade de manutenção, a chance de aceitação do produto, uma vez que esta preza por uma abordagem sistemática e organizada para seu desenvolvimento (SOMMERVILLE, 2011). O objetivo deste trabalho é construir uma ferramenta para coletar e extrair dados, para cálculo de métricas para os repositórios de código fonte hospedados no GitHub[3], provendo a descoberta de tendências em um projeto de software, a partir dos dados minerados.

A mineração dos dados também pode ser utilizada para relacionar os acontecimentos a aspectos de evolução, considerando a temporalidade em que os fatos acontecem, a temporalidade é importante, pois estabelece formas diferentes de análises.

[3] GtHub. Disponível em: <https://github.com/>. Acessado em 12 de dezembro de 2021.

A temporalidade envolve dados que são obtidos no passado, no presente e que podem gerar dados no futuro.

Conforme Nascimento (2017), dados do GitHub permitem que analisemos informações técnicas e sociais a respeito do código, tornando-se o grande diferencial para os repositórios centralizados e sites de hospedagem de projetos de software como o SourceForge[4].

Esse trabalho é continuação do trabalho desenvolvido por Oliveira, Tostes e Sirqueira (2021). Além dessa introdução, na seção 2 apresentaremos o referencial teórico do trabalho, já a seção 3 apresenta os trabalhos relacionados. A seção 4 explora a API do GitHub utilizada pela ferramenta. A seção 5 demostramos a utilização da ferramenta por meio de um caso de uso. Por fim, a seção 6 apresenta as limitações do trabalho, os próximos passos e as considerações finais.

2 REFERENCIAL TEÓRICO

A mineração de repositórios de código fonte auxilia a validar as leis de evolução de software, além de auxiliar os gerentes de projetos a entenderem a dinâmica do sistema mantido, assim como a do grupo mantenedor. A seguir abordaremos sobre repositórios de código fonte, sobre o GitHub e mineração de repositórios.

2.1 Repositórios de Código Fonte

Há uma variedade de ferramentas de hospedagem de repositórios de código fonte disponíveis na web. Contudo, é importante compreender que os serviços de hospedagem de repositórios e os sistemas de controle de versão são duas entidades separadas.

Os sistemas de controle de versão são os utilitários usados para gerenciar as mudanças no ciclo de vida do desenvolvimento de software, gravando as alterações que ocorrem nos arquivos de código-fonte, sendo exemplos o Git[5] e o SVN[6]. Já Os serviços de hospedagem de repositório são aplicações da web que envolvem e melhoram o sistema de controle de versão, sendo exemplos o GitLab[7] e o GitHub.

O GitHub é mais popular do que o GitLab na comunidade de desenvolvedores e por isso, foi adotado como repositório para mineração.

2.2 Plataforma GitHub

[4] SourceForge. Disponível em: <https://sourceforge.net/>. Acessado em 12 de dezembro de 2021.
[5] Git. Disponível em: <https://git-scm.com/>. Acessado em 13 de dezembro de 2021.
[6] SVN. Disponível em: <https://subversion.apache.org/>. Acessado em 13 de dezembro de 2021.
[7] GitLab. Disponível em: <https://about.gitlab.com/>. Acessado em 13 de dezembro de 2021.

O GitHub foi um dos pioneiros em hospedagem de repositórios Git, lançado em 2008, onde qualquer usuário que possua cadastro na plataforma pode contribuir com projetos privados ou *Open Source*. Possui grandes projetos hospedados, como por exemplo, *WordPress* e o *kernel* do *GNU*/*Linux*.

Em Julho de 2018 o GitHub foi comprado pela Microsoft com valores estimados em cerca de US$ 7,5 bilhões. De Acordo com o próprio GitHub, existem 31 milhões de desenvolvedores e mais de 2,1 milhões de empresas e organizações que o GitHub.

Com um viés colaborativo, a plataforma GitHub permite que o programador compartilhe blocos de códigos, comentar e alterar arquivos em repositórios de outros programadores, para corrigir ou incluir novas funcionalidades (COSTA e PONCIANO, 2018).

2.3 Mineração de Repositórios de Software

O crescente desenvolvimento das comunidades software livre tem proporcionado a evolução da área de estudo definida como MRS, que inclui atividades de análise, mineração e recuperação de dados, possibilitando investigação estatística dos repositórios de software armazenados em ambientes distribuídos (COSTA e PONCIANO, 2018).

Os repositórios do GitHub oferecerem dados históricos de evolução de códigos fontes, as contribuições dos programadores e marcos no desenvolvimento do software, que permitem a investigação e posterior extração de informações para reconhecimento de tendências e padrões.

Esses recursos foram explorados na ferramenta desenvolvida “GIT Viewer” e serão apresentados na seção 5, abordando as telas da ferramenta e os dados que foram obtidos do GitHub durante a análise de um repositório.

3 TRABALHOS RELACIONADOS

Em Robbes (2007), abordou que sistema de controle de versão apesar de úteis para avaliar a evolução de um software, as informações que eles contêm são limitadas de várias maneiras. Neste trabalho foi proposto um repositório de informações alternativo para armazenar as mudanças nos repositórios.

No trabalho de Poncin, Serebrenik e Van Den Brand (2011), desenvolveu-se o FRASR (*Framework for Analyzing Software Repositories*), aumentando o framework de

mineração de processos ProM. Isto é, aplicaram técnicas de *Process Mining*, para mineração de repositórios de software.

Já Ali, Guéhéneuc e Antoniol (2012), apresentaram o Trustrace, uma abordagem de recuperação de rastreabilidade baseada em confiança, para mineração de repositórios de software, baseado em técnicas de recuperação da informação.

O MetricMiner (SOKOL, ANICHE e GEROSA, 2013), é uma aplicação web que ajuda em algumas etapas da mineração de repositórios de software, como cálculo de métricas, extração de dados e inferência estatística.

No trabalho de Kalliamvakou et al. (2014), os pesquisadores exploraram as informações armazenadas nos logs de eventos do GitHub, buscando entender como seus usuários utilizam o site para colaborar no software. Além disso, forneceram um conjunto de recomendações para abordar os dados no GitHub.

Por fim, no trabalho de Costa e Ponciano (2018), explorou a interação entre os programadores dentro do GitHub, onde a análise dos perfis permitiu compreender a influência dos comportamentos dos programadores com os níveis de sucesso de seus repositórios.

4 GitHub API

O termo API significa Application Program Interfaces, que os desenvolvedores usam para acessar ferramentas da web ou informações na nuvem. A API do GitHub permite usar e interagir com o GitHub, criando e gerenciando repositórios, *branches*, problemas, solicitações de *pull* e etc. Algumas solicitações são abertas enquanto para outras ações, é necessário fornecer um token autenticado.

A URL base para APIs do GitHub é https://api.github.com/. Neste projeto exploramos 4 *endpoint* da API, onde informa-se o usuário e o repositório que se deseja

analisar. Utilizando como exemplo o usuário WordPress e o seu repositório WordPress, temos as seguintes chamadas a API:

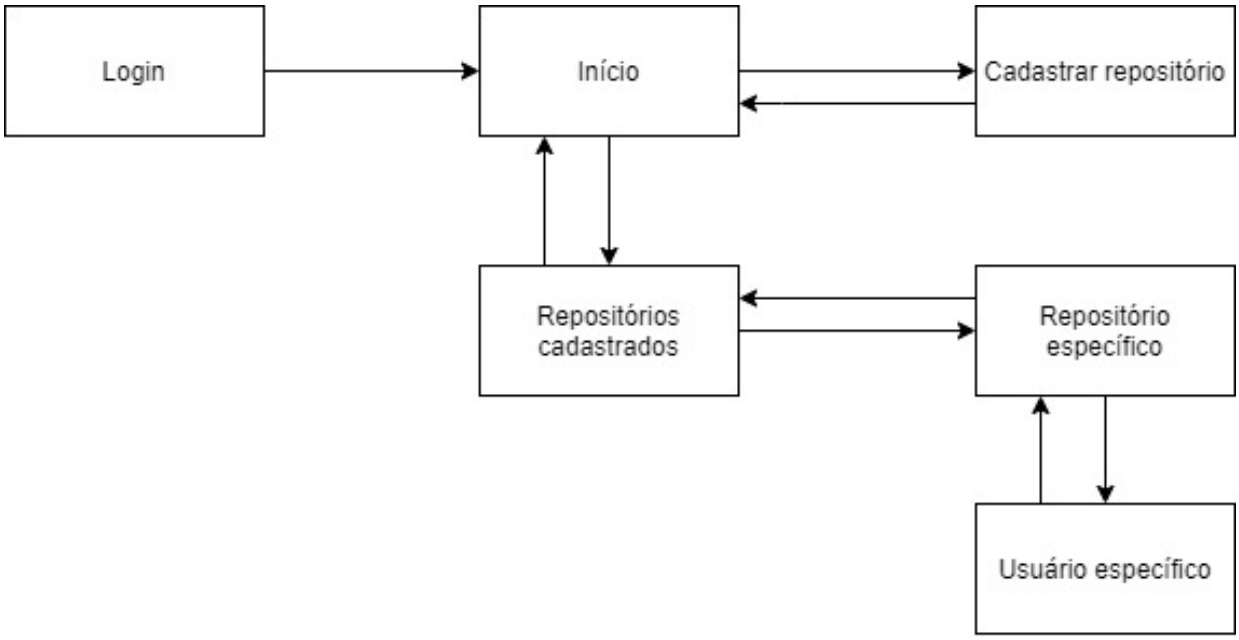
1. Repositórios: https://api.github.com/repos/WordPress/WordPress
2. *Tags* do repositório: https://api.github.com/repos/WordPress/WordPress/tags?per_page=100&page=6
3. Usuários do repositório: https://api.github.com/repos/WordPress/WordPress/contributors?per_page=100&page=1
4. Informações de um usuário específico: https://api.github.com/users/SergeyBiryukov

Cada chamada é explorada em uma tela diferente pela ferramenta desenvolvida.

5 DEMONSTRAÇÃO DA PLATAFORMA

A plataforma desenvolvida segue o fluxo presente na Figura 1. Cada etapa é representada por uma tela diferente dentro da ferramenta e abordam o gerenciamento e visualização de dados dos repositórios, assim como a sua exportação.

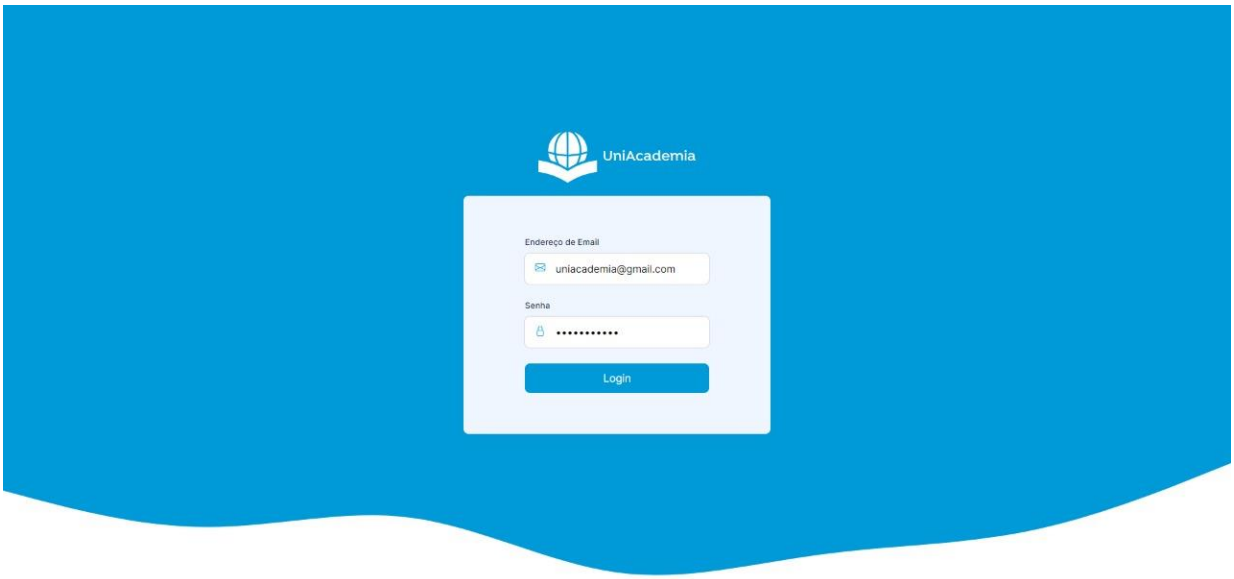
Figura 1: Fluxo de execução da ferramenta.



Fonte: Elaboração própria.

Conforme apresenta a Figura 2, ao abrir a ferramenta, deve-se autenticar na ferramenta para recuperar os repositórios já cadastrados. Caso não tenha cadastro é necessário realizá-lo no primeiro acesso.

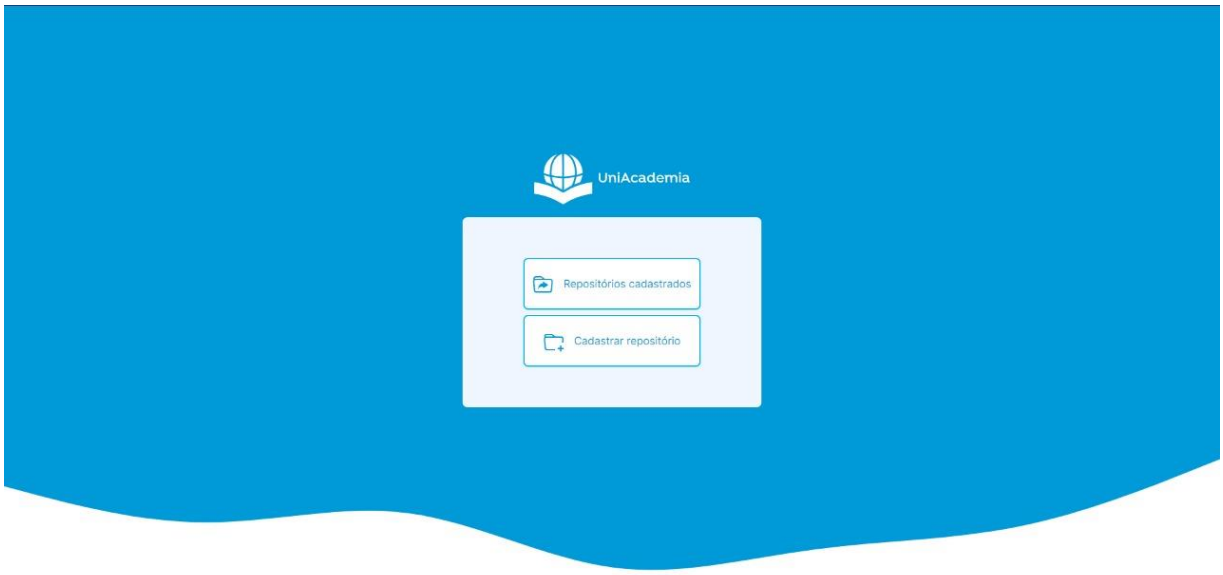
Figura 2: Definição do repositório na ferramenta.



Fonte: Elaboração própria.

Após autenticado, são exibidas duas opções (conforme Figura 3). A primeira opção é listar os repositórios cadastrados, ou como segunda opção, iniciar o cadastro de um novo.

Figura 3: Tela inicial da ferramenta.



Fonte: Elaboração própria.

Ao solicitar o cadastro de um repositório para análise, é exibido a tela da Figura 4, onde deve-se informar o usuário a qual o repositório pertence e seu respectivo nome.

Figura 4: Cadastro de repositório.

Fonte: Elaboração própria.

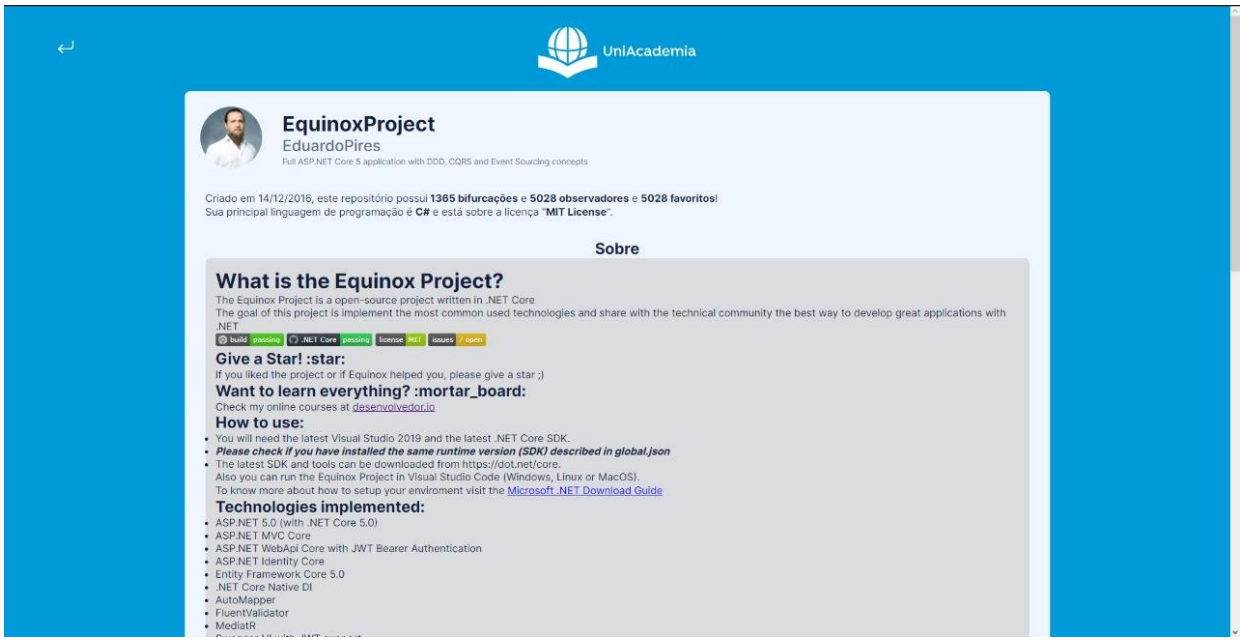
Com os repositórios cadastrados, pode-se listá-los e ver cada uma de suas informações básicas, conforme Figura 5.

Figura 5: Repositórios cadastrados.

Fonte: Elaboração própria.

Ao selecionar determinado repositório, os detalhes são expandidos e as informações passam para o modo detalhado, conforme a Figura 6. Além de consultar os detalhes do repositório, esta tela permite que os dados sejam exportados em JSON (*JavaScript Object Notation*) ou CSV (*Comma-separated values*).

Figura 6: Detalhes do repositório.



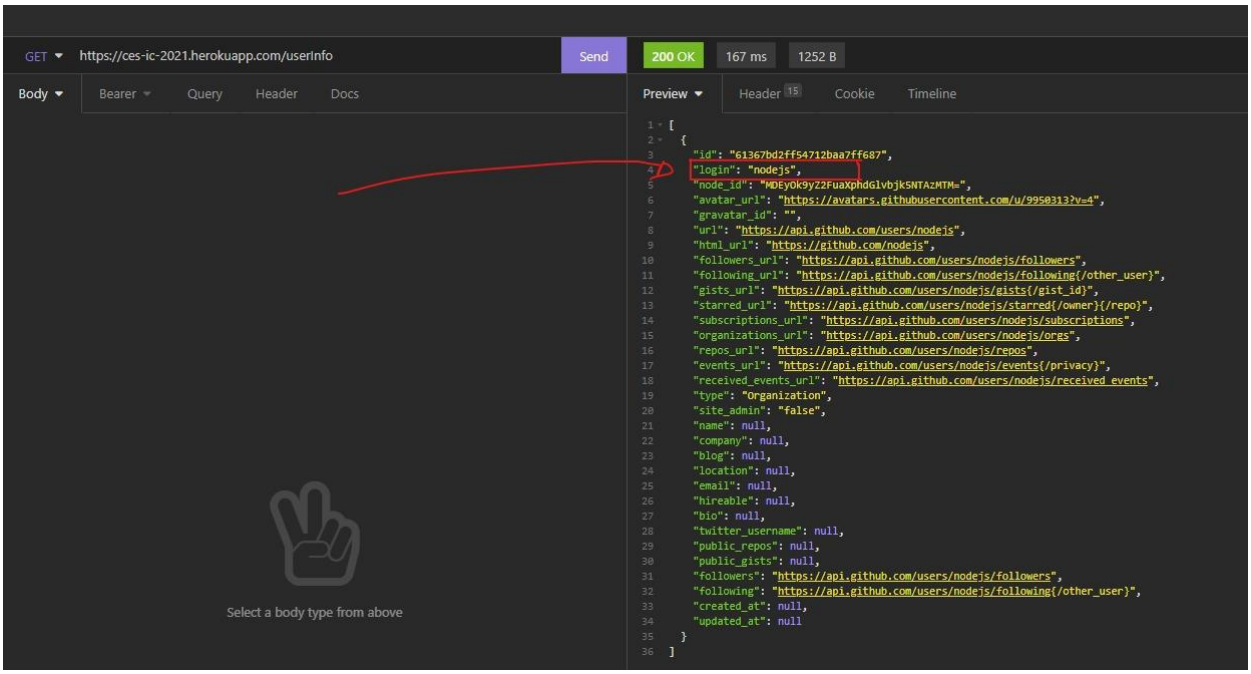
Fonte: Elaboração própria.

A exportação dos dados é feita após tratamento dos mesmos pela ferramenta, que os retorna seguindo o formato desejado direto do servidor da ferramenta, conforme Figura 7, onde a exportação dos dados do usuário “nodejs” foi solicitada em JSON.

O *back-end* da aplicação foi desenvolvida com Spring Boot[8], Java[9] 8 e possui como banco de dados o MongoDB[10]. Todo acesso ao *back-end* é via API REST (*Representational State Transfer*) e a ferramenta encontra-se hospedado no Heroku[11].

[8] Spring Boot. Disponível em: <https://spring.io/projects/spring-boot>. Acessado em 13 de dezembro de 2021.
[9] Java. Disponível em: <https://www.java.com/pt-BR/>. Acessado em 13 de dezembro de 2021.
[10] MongoDB. Disponível em: <https://www.mongodb.com/>. Acessado em 13 de dezembro de 2021.
[11] Heroku. Disponível em: <https://www.heroku.com/>. Acessado em 13 de dezembro de 2021.

Figura 7: Exportação dos dados JSON.



Fonte: Elaboração própria.

Já o *front-end* foi construído em Next.js[12], um *framework* React[13] com foco em produção e eficiência criado e mantido pela equipe da Vercel[14], o Next.js busca reunir diversas funcionalidades como renderização hibrida e estática de conteúdo, suporte a *TypeScript*, *pre-fetching*, sistema de rotas, pacotes de funcionalidades e diversos plugins.

O Next.js adiciona várias funcionalidades em cima do React, com uso do Next.js a interface web da ferramenta é adaptativa para dispositivos móveis. A Figura 8 exemplifica a tela inicial para dispositivos móveis, onde tem as opções de listar e cadastrar os repositórios.

[12] Next.js. Disponível em: <https://nextjs.org/>. Acessado em 13 de dezembro de 2021.
[13] React. Disponível em: <https://pt-br.reactjs.org/>. Acessado em 13 de dezembro de 2021.
[14] Vercel. Disponível em: <https://vercel.com/>. Acessado em 13 de dezembro de 2021.

Figura 8: Tela inicial para dispositivos móveis.

Fonte: Elaboração própria.

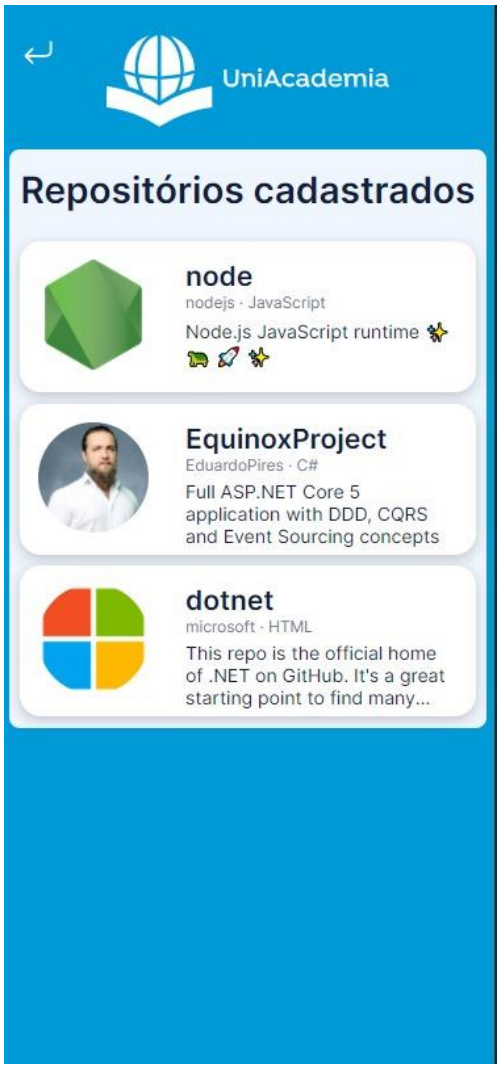
Já a Figura 9, demostra a interface de cadastro de repositório voltada para dispositivos móveis, seguindo o modelo web, deve-se inserir o usuário e o nome do repositório.

Figura 9: Cadastro de repositórios para dispositivos móveis.

Fonte: Elaboração própria.

Por fim, na Figura 10 é apresentada a tela de repositórios adaptada para dispositivos móveis. Por conta da dimensão da tela, determinadas informações são ocultadas de forma a garantir a legibilidade das informações e conforme a tela amplia as informações voltam a serem exibidas.

Figura 10: Lista de repositórios para dispositivos móveis.



Fonte: Elaboração própria.

O código fonte da plataforma está divido em *back-end* disponível em <https://github.com/ces-jf/IC-2021-Back-end> e *front-end* disponível em <https://github.com/ces-jf/IC-2021-Front-end> e um vídeo de demonstração do uso pode ser visto no endereço <https://www.youtube.com/watch?v=bWN4rhbr5q8>.

6 CONSIDERAÇÕES FINAIS

Esse trabalho é resultado de uma iniciação cientifica do UniAcademia e buscou explorar novas tecnologias para a construção da ferramenta GIT Viewer. A arquitetura REST da ferramenta possibilita *Web Services* mais leves e na direção da metodologia ágil, com flexibilidade na escolha do formato na troca de mensagens e no uso da interface.

A API do GitHub permite explorar uma serie de dados, contudo, a falta de uma ferramenta que os consolidem foi a motivação para a construção desta ferramenta. Algumas limitações são com relação a repositórios privados, onde a aplicação depende do token de autorização para acessar as informações por completo e nas exportações, que hoje são por grupo de visão. Pretende-se em versão futuro consolidar todas as visões em uma única exportação, facilitando a análise posterior por ferramentas estatísticas.

Entender a dinamia de do ciclo de vida de um software não é uma tarefa trivial, pois envolve vários elementos, e por conta disso, a MRS é um caminho que permite explorar a partir de dados históricos o comportamento ao longo do tempo de sua evolução, auxiliando a melhorar o processo de desenvolvimento de software e da gestão de equipes.

REFERÊNCIAS

ALI, Nasir; GUÉHÉNEUC, Yann-Gaël; ANTONIOL, Giuliano. **Trustrace: Mining software repositories to improve the accuracy of requirement traceability links**. IEEE Transactions on Software Engineering, v. 39, n. 5, p. 725-741, 2012.

COSTA, Victor; PONCIANO, Lesandro. **Minerando Padrões de Interação de Programadores com Repositórios na Plataforma GitHub**. PUC-Minas. 2018.

HASSAN, Ahmed E.; XIE, Tao. **Mining software engineering data**. In: 2010 ACM/IEEE 32nd International Conference on Software Engineering. IEEE, 2010. p. 503-504.

JUNIOR, Roma; NASSIF, Douglas. **Uma ferramenta para mineração de dados de projetos de software livre e criação de redes sócio-técnicas**. 2013. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

KALLIAMVAKOU, Eirini et al. **The promises and perils of mining GitHub**. In: Proceedings of the 11th working conference on mining software repositories. 2014. p. 92-101.

NASCIMENTO, André Luan Chiquetto. **Mineração de dados aplicada ao controle de prazos e prioridades em projetos de software**. 2017. Trabalho de Conclusão de Curso. Universidade Tecnológica Federal do Paraná.

OLIVEIRA, Cláudio; TOSTES, Rogério; SIRQUEIRA, Tassio Ferenzini Martins. **GIT Viewer: uma plataforma para análise de dados do GitHub**. ANALECTA-Centro Universitário Academia, v. 6, n. 3, 2021.

PONCIN, Wouter; SEREBRENIK, Alexander; VAN DEN BRAND, Mark. **Process mining software repositories**. In: 2011 15th European Conference on Software

Maintenance and Reengineering. IEEE, 2011. p. 5-14.

PRESSMAN, Roger; MAXIM, Bruce. **Engenharia de Software**-8ª Edição. McGraw Hill Brasil, 2016.

ROBBES, Romain. **Mining a change-based software repository**. In: Fourth International Workshop on Mining Software Repositories (MSR'07: ICSE Workshops 2007). IEEE, 2007. p. 15-15.

SOKOL, Francisco Zigmund; ANICHE, Mauricio Finavaro; GEROSA, Marco Aurélio. **MetricMiner: Supporting researchers in mining software repositories**. In: 2013 IEEE 13th International Working Conference on Source Code Analysis and Manipulation (SCAM). IEEE, 2013. p. 142-146.

SOMMERVILLE, Ian et al. **Engenharia de software**.[Sl]. Pearson Education, v. 19, p. 23, 2011.