Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Web-Crawler Paralelo para Acompanhamento de Registros de Extensão

João Pedro Sadéri da Silva

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientadora
Prof.a Dr.a Carla Maria Chagas e Cavalcante Koike

Brasília
2023

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Web-Crawler Paralelo para Acompanhamento de Registros de Extensão

João Pedro Sadéri da Silva

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof.a Dr.a Carla Maria Chagas e Cavalcante Koike (Orientadora)
CIC/UnB

Prof. Dr. Daniel de Paula Porto
CIC/UnB

Prof. Dr. Marcelo Marotta
CIC/UnB

do Bacharelado em Ciência da Computação

Brasília, 21 de dezembro de 2023

Dedicatória

Esse trabalho de conclusão de curso dedico em especial aos meus pais, Carlos César da Silva e Cleide Aparecida Sadéri da Silva, que me apoiaram em todo o meu caminho acadêmico. Dedico também aos meus familiares e aos meus amigos que me acompanharam até aqui.

Agradecimentos

Meus agradecimentos são destinados aos professores do departamento de Ciência da Computação, aos professores do Instituto de Exatas, aos coordenadores e aos membros do Decanto de Extensão da Universidade de Brasília. Em especial, quero agradecer a minha orientadora, Prof. Dra. Carla Maria Chagas e Cavalcante Koike, por todo acompanhamento fornecido ao longo dos últimos anos.

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

Esse trabalho trata do desenvolvimento do Web Crawler responsável por obter os dados da plataforma SIGAA para acompanhamento dos registros de extensão na Universidade de Brasília, bem como das técnicas de programação concorrente e paralelas empregadas para acelerar a obtenção desses dados. O algoritmo responsável por realizar o cálculo de indicadores e os tratamentos realizados para reduzir os arquivos usados são também descritos.

Palavras-chave: extensão universitária, indexador web, indexador web concorrente, coleta de dados da web, coleta de dados da web concorrente

Abstract

This work deals with the development of the Web Crawler responsible for obtaining data from the SIGAA platform to monitor extension records at the University of Brasília, as well as the techniques of concurrent and parallel programming employed to accelerate the retrieval of this data. The algorithm responsible for calculating indicators and the treatments performed to reduce the used files are also described.

Keywords: university extension, web crawler, concurrent web crawler, web scraping, concurrent web scraping

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

API Application Programming Interface.

DEX Decanato de Extensão.

JSON JavaScript Object Notation.

MEC Ministério da Educação.

SARUE Sistema de Acompanhamento dos Registros Universitários de Extensão.

Semuni Semana Universitária.

SIEX Sistema de Extensão.

SIGAA Sistema Integrado de Gestão de Atividades Acadêmicas.

STI/UnB Secretaria de Tecnologia da Informação da UnB.

TCU Tribunal de Contas da União.

UnB Universidade de Brasília.

Nomenclaturas

$T_{i\text{ano}}$ Tempo de execução sequencial no ano i

$T_{i\text{quadrimestre}}$ Tempo de execução sequencial no quadrimestre i

$T_{i\text{semestre}}$ Tempo de execução sequencial no semestre i

$T_{i\text{trimestre}}$ Tempo de execução sequencial no trimestre i

T_{tpa} Tempo total por ano

T_{tpq} Tempo total por quadrimestre

T_{tps} Tempo total por semestre

T_{tpt} Tempo total por trimestre

Capítulo 1

Introdução

De acordo com a resolução Nº 7, de 18 de dezembro de 2018 publicada no Diário Oficial da União pelo Ministério da Educação (MEC) [1], a atividades extensão na educação superior pode ser definida como atividade que se integra à matriz curricular e à organização da pesquisa, constituindo-se em processo interdisciplinar, político e educacional, cultural, científico, tecnológico, que promove a integração transformadora entre as instituições de ensino e os outros setores da sociedade, por meio da produção e da aplicação do conhecimento, em articulações permanente com um ensino e a pesquisa.

Ainda com base na resolução publicada pelo MEC [1], ela apresentada em seu Artigo 4º, que *"As atividades de extensão devem compor, no mínimo, 10% (dez por cento) do total da carga horária curricular estudantil dos cursos de graduação, as quais deverão fazer parte da matriz curricular dos cursos;"*, após essa publicação, a UnB deverá atualizar as matrizes curriculares de cada curso para integrar a extensão a partir do ano de 2023 conforme página do Decanato de Extensão (DEX) [2].

A missão do DEX, exibida na página [3], é contribuir para a democratizar as relações entre a Universidade e a sociedade em busca do desenvolvimento sustentável. Para isso, existe a importância de obter indicadores sobre a extensão universitária além de poder realizar análises com base nesses indicadores.

O DEX é responsável por calcular e divulgar indicadores de extensão, os quais são métricas ou medidas utilizadas para avaliar e acompanhar o impacto e a efetividade das atividades realizadas pela universidade, assim como auxiliar na tomada de decisões em relação a extensão.

Após a instalação do Sistema Integrado de Gestão de Atividades Acadêmicas (SIGAA) pela Universidade de Brasília (UnB), as ações de extensão realizadas pela Universidade obtiveram um espaço para serem exibidas e isso permitiu aos seus usuários a possibilidade de se candidatar e se inscrever em diversas ações de extensão.

O sistema SIGAA no ponto de vista da extensão, além de possibilitar a inscrição e a gestão dos participantes, ele automatiza a geração dos certificados aos participantes de ações e fornece alguns relatórios sobre as ações realizadas.

1.1 Definição do Problema

Apesar de ser uma plataforma inovadora muito eficiente para gestão de projetos, eventos e ações, o sistema não se adapta rapidamente aos indicadores e relatórios necessários. Mesmo com a iniciativa de promover a manipulação da equipe e dos participantes, o sistema apresenta algumas limitações em relação aos dados que podem ser obtidos através dele.

Em parceria com DEX, foi proposto elaborar uma ferramenta que permitisse aos usuários a observação das ações realizadas na extensão de uma forma simplificada, através do uso de gráficos baseados em indicadores, utilizando como base os dados contidos no sistema SIGAA.

Os dados contidos na plataforma SIGAA são suficientes para o cálculo de alguns indicadores de extensão. Em função da ausência de uma Application Programming Interface (API) que forneça os dados contidos na plataforma para que os indicadores possam ser diretamente calculados, foi implementado um *web crawler* com a finalidade de navegar pelas páginas de extensão e obter os dados nelas cadastrados.

Esse trabalho relata o desenvolvimento do *web crawler* responsável por obter os dados da plataforma SIGAA, bem como das técnicas de programação concorrente e paralelas empregadas para acelerar a obtenção desses dados. Ainda nesse documento é descrito o algoritmo responsável por realizar o cálculo dos indicadores e os tratamentos realizados para reduzir os arquivos usados por outras partes do sistema.

1.2 Objetivos

1.2.1 Gerais

Este trabalho trata do desenvolvimento do *back-end* da ferramenta que obtém os dados de ações de extensão do sistema SIGAA e calcula os indicadores de extensão.

É desejado que os usuários dessa ferramenta possam, por meio da análise dos indicadores fornecidos, decidir sobre a aplicação de verbas e investimentos em determinadas áreas, assim como interpretar com base na progressão temporal, a quantidade e a qualidade das ações realizadas pela universidade.

A ferramenta também visa fornecer indicadores auditáveis aos órgãos de controle competentes.

1.2.2 Específicos

Além do objetivo geral, abaixo sequem alguns objetivos específicos no desenvolvimento dessa ferramenta.

Elicitação de requisitos

Um dos objetivos específicos consiste em realizar a elicitação dos requisitos do projeto. No início do projeto, o grupo de desenvolvimento deve identificar o problema para só então propor soluções. A identificação do problema é obtida através de reuniões com os *stakeholders*, que descrevem as dificuldades na obtenção e cálculo dos indicadores de extensão universitária.

São utilizados roteiros para entrevistas com o intuito de verificar a necessidade de cada *stakeholder*. Esse roteiro também visa permitir a identificação de questões técnicas do projeto.

Durante as entrevistas foi possível o acesso ao SIGAA por meio das credenciais de diferentes perfis do DEX. Por meio do acesso com diferentes credenciais, foi possível observar que os dados brutos não podem ser recuperados do SIGAA, demonstrando a necessidade de outra abordagem. Durante as entrevistas também são definidos quais indicadores devem ser calculados pela ferramenta. Cada entrevista deve ver também verificar qual a frequência do indicado indicador para então fornecer uma estimativa da frequência inspirada de utilização daquele indicador.

Ainda na elicitação de requisitos do sistema, existe outra parte voltada as questões técnicas desse projeto, que consistem em obter as ferramentas e linguagens que devem ser usadas para obter os dados e calcular os indicadores.

Coleta das informações de extensão realizada pela universidade

Esse objetivo é voltado a implementação de uma solução que permita obter os dados de extensão cadastrados no sistema SIGAA.

Testes na plataforma do sistema SIGAA e conversas com os *stakeholders* permitem identificar a falta de uma API pública para a coleta dos dados.

Dessa forma, esse objetivo foca em listar possibilidades para a obtenção dos dados. Como a possibilidades de utilizar um *web crawler* ou um *web scraping* capaz de acessar as páginas de extensão e obter os dados cadastrados na plataforma SIGAA.

Tratamento de credenciais utilizadas pelo sistema

Ainda com objetivo de obter os dados cadastrados na plataforma, existe a necessidade de analisar a autenticação do sistema SIGAA. Para o acesso do *web crawler* às páginas de extensão, é necessário que a instância do navegador esteja autenticada.

Logo, esse objetivo é obter e utilizar uma credencial do sistema SIGAA no script desenvolvido. Além de obter essa credencial, é necessário armazenar e garantir o funcionamento a partir dessa credencial.

Existem diversos perfis de usuário da plataforma SIGAA, por isso, cada tipo de credencial pode resultar em um esquema diferente de autenticação dos usuários e de caminhos diferentes para obter os dados. Ainda nesse objetivo, existe também a necessidade de implementar uma solução que funcione para qualquer tipo de credencial fornecida.

Verificação de integridade e totalidade dos dados

Uma vez que a ferramenta conseguiu obter os dados desejados, é necessário realizar verificações e tratamentos para garantir que os dados obtidos estejam em acordo com os cadastrados na plataforma.

Além disso, é necessário implementar soluções de verificação de falhas ou mudanças no sistema SIGAA. Essas soluções podem ser voltadas a manutenção, a possibilidade de usar o algoritmo a partir de qualquer máquina ou até mesmo o uso de testes unitários para verificar as páginas de extensão.

Criação de uma planilha com os dados do SIGAA

Esse objetivo é voltado a suprir deficiências do sistema SIGAA. Com os dados obtidos pelo *back-end* do sistema SARUE, é possível converter esses dados para uma planilha, permitindo que os usuários possam visualizar todas as ações atualmente cadastradas. Além disso, por meio da manipulação dessa planilha, os usuários podem calcular novos indicadores de extensão que ainda não são contemplados pela interface gráfica.

Avaliação e eventual redução do tempo de execução da ferramenta

Dependendo da forma de acesso aos dados, o tempo de execução da ferramenta pode ser extenso. Um dos principais objetivos para o funcionamento desse sistema é permitir que os dados sejam obtidos de forma mais rápida.

Para isso, após uma implementação inicial e a verificação de que os dados poderiam ser obtidos por meio de um *web crawler*, existiram pesquisas e teste para melhorar o tempo para obtenção dos dados. Esse objetivo visa ou uso de técnicas de programação paralela e concorrentes para permitir o funcionamento simultâneo do script.

Cálculo dos indicadores

A partir dos dados obtidos, os indicadores de extensão precisam ser calculados. Dos vários indicadores a serem calculados, muitos deles possuem diferentes divisões em função do tempo. O cálculo visa por meio de diferentes períodos de tempo, dividir os indicadores calculados, permitindo que interface gráfica possa gerar tabelas e gráficos com os indicadores calculados.

Ainda nessa fase, um dos objetivos é remover as possíveis repetições contidas nos indicadores. Com base nos indicadores de âmbito anual, é preciso fornecer os dados de forma mensal para criação de visualizações mensais e ao mesmo tempo é tratar os dados anuais de forma separada. Isso se deve a possibilidade de ocorrer repetições nos dados mensais em relação aos anuais.

Comunicação entre as frentes do projeto

Com o objetivo de exibir os indicadores obtidos por meio dos dados da plataforma SIGAA é necessário que os dados calculados por esse sistema possam ser enviados com facilidade a outro sistema. Com isso, esse objetivo é dado pela necessidade que os dados obtidos possam ser armazenados em diferentes tipos de arquivos de acordo com a necessidade de ambos os sistemas.

1.3 Estrutura do trabalho

O Capítulo 2 desse documento aborda as referências que embasam esse trabalho. Isso inclui a identificação das deficiências reportadas no sistema SIGAA, o uso de benchmarks para estimar o tempo em máquinas diferentes, a distinção entre web crawling e web scraping, bem como técnicas para acelerar o web crawling com o uso de programação paralela e concorrente. Ainda nesse capítulo são descritas as principais ferramentas e matérias utilizadas para o desenvolvimento do projeto.

No Capítulo 3, é detalhado o processo de desenvolvimento do projeto. Ele compreende a descrição dos diferentes Crawlers utilizados e apresenta as abordagens paralela e concorrente para acelerar o Crawler na obtenção de dados. Nesse capítulo também são descritas as soluções empregadas para a manutenção do sistema e os testes unitários desenvolvidos. Por fim, é descrita a metodologia empregada para o cálculo e verificação dos indicadores.

O Capítulo 4 apresenta os resultados, com destaque para uma comparação dos tempos obtidos pelas diferentes abordagens utilizadas para acelerar a obtenção dos dados do sistema SIGAA. Nesse capítulo também é descrita a metodologia usada para comparar as diferentes maquinas utilizadas e elicitada as principais variáveis para diferenças nos

dados obtidos. Por fim, esse capítulo oferece uma visão sobre a eficácia das abordagens utilizadas no sistema.

Por fim, o Capítulo 5 recapitula os principais pontos discutidos nessa monografia, além de sugerir possíveis direções para trabalhos futuros nessa área.

Capítulo 2

Referencial Teórico

Nesse capítulo é exposto os principais artigos ou livros usados como referencias para o desenvolvimento do projeto. Ainda é exibida as principais ferramentas e linguagens de programação usadas no desenvolvimento.

2.1 Processos de extensão

O processo de realizar uma ação de extensão vai além do cadastro e da execução dessa ação. Existem etapas como cadastrar e registrar a frequências dos participantes, gerar os certificados para a equipe organizadora e para os participantes.

Todas essas etapas para uma ação de extensão podem resultar na necessidade de utilizar mais de um sistema. O sistema SIGAA, de acordo com a perspectiva exibida no artigo [4], permite que etapas como o cadastro, as inscrições, mudanças na gestão da equipe organizadora e geração dos certificados sejam realizadas por um único sistema.

Além disso, com o uso do sistema SIGAA tratando da curricularização da extensão, os créditos obtidos em ações de extensão passam a serem integrados a graduação. Logo após a emissão dos certificados, os créditos referentes a eles já são destinados aos discentes.

2.2 Sistemas de extensão

O artigo Gestão da Política de Extensão Na UnB: Desafios e Possibilidades [5], demonstra diversos problemas no uso do sistema SIGAA voltado a extensão universitária. Além disso, faz referencias ao antigo SIEX que se trata do antigo Sistema de Extensão da UnB.

No artigo é demonstrado uma pesquisa realizada sobre as principais dificuldades na utilização do sistema SIGAA voltado ao módulo de extensão universitária.

O documento fala sobre os principais pontos analisados para verificar a ineficiência do sistema SIGAA na capacidade de suprir questões burocráticas quanto a prestação de contas ou a destinação de orçamento para ações de extensão.

Com base em trechos de entrevistas mostradas no artigo, é possível verificar que mesmo após a mudança ao sistema SIGAA, o módulo de extensão não era capaz de suprir todas as necessidades dos usuários do sistema. Por exemplo, verificar de forma conjunta, qual a origem do financiamento destinada a uma ação de extensão. O SIGAA apesar de possuir essa informação, o local de coleta por informações sobre o status de uma ação e a origem financeira dessa mesma ação não possuem relação entre si.

Ao mesmo tempo, a mudança ao sistema SIGAA possibilitou que a eficácia, relevância e impacto social das ações de extensão da UNB pudessem ser avaliadas através do relatório final ligado aquela ação especifica.

O artigo mostra uma análise realizada sobre a quantidade de discente e docentes envolvidos em ações de extensão realizadas pela UnB. Esses dados eram fornecidos pela antigo SIEX. Atualmente, com o uso do SIGAA o quantitativo desses dados não faz parte da exibição pública fornecida pelo sistema.

2.3 Desempenho de Sistemas de Software

O uso de benchmarks para comparação entre máquinas diferentes utiliza benchmarks representativos [6]. A abordagem mostrada no livro trata da execução de um conjunto de benchmarks representativos. Em outras palavras, de escolher um conjunto de programas ou algoritmos que representem carga de trabalho reais que estimule o uso dos recursos computacionais destinados à sua aplicação.

Detalhes sobre a medição do desempenho de máquinas diferentes, de como obter e verificar métricas de desempenho, como tempo de execução, contagem de instruções ou instruções por ciclo. A abordagem de comparação com base no tempo de execução é utilizada no Capítulo 4 na seção 4.7.

Além disso, por meio desse livro é possível listar diversos motivos para variações obtidas nos resultados. Isso se deve a impossibilidade de seguir de forma constante todos os passos descritos para a comparação de máquinas diferentes, mas serve como comparação para os resultados apresentados.

Os testes apresentados no Capítulo 4 dessa monografia foram obtidos através da execução de um mesmo algoritmo em diversas máquinas. Com base na técnica de benchmark apresentada por essa referencia, foi realizado um comparativo através de um benchmark sintético executado em todas as máquinas.

2.4 Indicadores de extensão universitária

Ao longo da última década, a extensão universitária emergiu como um tema de crescente importância. O relatório [7] não apenas delineia a criação de um grupo de pesquisa focado nesses indicadores, mas também explora detalhadamente a metodologia empregada, apresentando conjuntos de resultados com base nos dados coletados.

O relatório destaca a importância da definição cuidadosa dos indicadores de gestão e avaliação. Além disso, ressalta a complexidade inerente à criação de um planejamento abrangente que atenda a necessidade de uso em diferentes universidades. Consequentemente, propõe o estabelecimento de uma base de referência para tais indicadores, enfatizando que cada organização deve ajustá-los conforme suas particularidades singulares.

A referencia Política Nacional de Extensão Universitária [8], aborda sobre os objetivos pactuados ao longo da existência do FORPROEX. Nesse documento, são tratados temas como a extensão universitária como um processo acadêmico definido e efetivado, como dimensão relevante a atuação universitária, como parte da solução de grandes problemas sociais do país, entre outros. Com base nessa referencia, é possível verificar a importância do uso de indicadores para verificar e avaliar a extensão universitária.

2.5 *Web crawling* e *web scraping*

Em especial, a referencia dada pelo livro [9] apresenta abordagens sobre a obtenção dos dados de uma página por meio de Crawlers e também do uso da biblioteca Selenium com Python3. Essa referencia também trata sobre a diferenciação entre *web scraping* para *web crawling*.

A diferença de *web scraping* para *web crawling*, segundo [9], está justamenteno nível de conhecimento necessário sobre a página pesquisada. No caso do *web scraping*, ele é mais voltado a obter páginas específicas da web. O *web scraping* navega diretamente a uma página e obtém informações específicas já conhecidas daquela página. Já o *web crawling* é voltado de forma automática a navegar pela página sem necessariamente conhecer todos os caminhos daquela página.

Esse projeto possui características de *web scraping* e de crawler. O sistema requer informações detalhadas das páginas que são pesquisadas, o que é característica de um *web scraping*. Ao mesmo tempo, o sistema é capaz de filtrar as ações de extensão e navegar por cada ação, o que é característica de um crawler.

Para a implementação de um *web scraping* e crawler bibliotecas externas podem ser

utilizadas. Existem bibliotecas como a urlib©[1], Requests©[2], Scrapy©[3] e Selenium©[4]. Todas elas possuem documentação oficial e materiais de apoio, para o desenvolvimento, o projeto utiliza a documentação oficial de cada uma para escolher qual ferramenta utilizar.

2.6 Programação Paralela e Concorrente em Crawler

Algumas referencias tratam sobre métodos de programação paralela e concorrentes em *web crawler*s. É o caso do livro [9], nele existe um capítulo que trata justamente sobre o uso da biblioteca voltadas a programação paralela como a concurrent.futures e de funções como a ThreadPoolExecutor usada nesse projeto. O exemplo descrito nessa referencia não utiliza a biblioteca Selenium usada nesse projeto. Por isso, diversas modificações foram empregadas com a finalidade de satisfazer as necessidades desse projeto.

Essa referencia também trata do uso de funções como a ProcessPoolExecutor, no inicio desse projeto, existiu uma tentativa do uso dessa função, entretanto por ela tratar da execução de processos em paralelo, ela utiliza áreas de memória destinas a um processo. O que gerava maior sobrecarga sobre os recursos do sistema da máquina.

O uso do paralelismo através dessas bibliotecas e do Selenium é justificado por meio da documentação existente em cada biblioteca. Além disso, a simplicidade de integração ao algoritmo existente foi levada em consideração. Outro fator é ligado a simplicidade em empregar e controlar a programação concorrente por meio de bibliotecas atualizadas do Python3.

Neste trabalho foram implementadas duas versões de execução paralela.

2.7 Ferramentas e Materiais utilizados

Os principais materiais utilizados no projeto são linguagens de programação e suas respectivas bibliotecas. Essas bibliotecas podem ser nativas da linguagem ou pacotes posteriormente instalados. A escolha dos materiais utilizados está relacionada a facilidade de implementação, verificação e legibilidade de uma linguagem.

2.7.1 Python3

No início do projeto, todos os desenvolvedores tinham conhecimento prévio em algumas linguagens. Pensando na possibilidade de revisões de código ou que melhorias pudessem

[1]https://docs.python.org/3/library/urllib.html
[2]https://pypi.org/project/requests/
[3]https://scrapy.org
[4]https://www.selenium.dev/pt-br/

ser realizadas por outros membros da equipe, uma linguagem de fácil entendimento deveria ser utilizada.

O uso da linguagem Python3 [10] foi motivado por uma série de razões convincentes no contexto do desenvolvimento de um *web crawler* e na geração de um arquivo JSON para o sistema descrito neste artigo.

A escolha do Python3 foi impulsionada pela sua notável simplicidade e facilidade de implementação. O Python3 é conhecido por sua sintaxe limpa e legível, o que torna o código do crawler mais fácil de ser entendido e mantido. Além disso, a vasta comunidade de desenvolvedores Python3 oferece um suporte valioso, tornando mais simples a resolução de problemas e a busca por soluções para desafios específicos da implementação de um crawler.

Outro motivo crucial para a escolha do Python3 é a riqueza de bibliotecas e frameworks disponíveis para tarefas relacionadas à *web scraping* e à análise de dados. Existem diversas opções de uso de bibliotecas para simular a navegação dentro de uma página web.

Além disso, o Python3 oferece um tempo de execução aceitável para o sistema em questão, mesmo sendo uma linguagem interpretada. Sua performance é suficiente para lidar com as tarefas de *web scraping* e manipulação de dados necessárias.

Outro benefício notável do Python3 é a forma eficiente como mantém as bibliotecas nativas atualizadas e simplifica a gestão de pacotes com o comando `pip`[11]. Com apenas um comando, é possível instalar ou atualizar qualquer pacote necessário, o que é particularmente útil ao trabalhar com bibliotecas de terceiros que aprimoram a funcionalidade do crawler ou auxiliam na criação do JSON final.

Ainda relacionado ao Python3, o seu gestor de pacotes `pip` é o responsável por satisfazer todas as demais dependências do projeto. No Capítulo 3 é explicado o funcionamento do módulo de requerimentos, que através do `pip`, realiza a verificação, instalação ou atualização das dependências relacionadas a esse projeto.

2.7.2 Selenium

Para simular a navegação pela página de extensão além de garantir que o sistema possa ser utilizado em futuras atualizações do site ou da linguagem, foi decidido o a utilização da biblioteca Selenium [12] no Python3. Apesar de não ser uma biblioteca nativa da linguagem, o gestor de pacotes do Python3, o `pip`[11], consegue instalar e atualizar a biblioteca.

O Selenium fornece um conjunto de comandos que visam simular ações em um navegador, como a interação entre um usuário e uma página na web.

Através de endereços ou posição de um dado em um site, os comandos da biblioteca podem realizar ações de entrada de dados, interação com dados da página, além da leitura

dos dados dessa página. Como exemplo disso, temos a ação de acessar uma página através de uma URL, localizar elementos como o campo de inserção de credenciais e simular o clique no botão de submeter esses dados. Através disso, é possível autenticar-se em uma página na web.

O Selenium apresenta uma excelente documentação oficial e extraoficial. Além de ser uma biblioteca com frequentes atualizações em função de mudanças nos drivers de navegadores e uma comunidade muito ativa em reportar ou solucionar problemas de implementação.

Essa biblioteca também fornece suporte a múltiplos navegadores, o que permite a execução em máquinas e sistemas operacionais diferentes. Outro fator relacionado é a compatibilidade com a automação de testes e ferramentas de testes unitários. Diversos testes unitários são voltados ao uso do Selenium para verificar o sistema SIGAA ou para verificar o sistema SARUE.

2.7.3 Python-Dotenv

A python-dotenv [13] foi utilizada para gerir a credencial dos usuários na plataforma. Por questões de segurança e versionamento de códigos, as credenciais de um usuário nunca são armazenadas dentro do próprio algoritmo, mas em um arquivo com extensão .env. Para isso, a biblioteca python-dotenv fornece um conjunto de funções para a correta obtenção ou salvamento dessas credenciais.

Por sua facilidade de configuração, essa biblioteca permite que com apenas um comando seja obtido o login ou senha de usuário armazenada no arquivo .env. O processo de armazenar uma nova credencial segue o mesmo princípio.

2.7.4 TQDM

Para fornecer uma visualização do que está acontecendo na execução do algoritmo, diversas técnicas podem ser utilizadas. Exibir textos no terminal, ou uma quantidade de instruções executadas pode aprimorar a experiencia do usuário e demonstrar a execução do algoritmo.

A biblioteca Tqdm [14] quando usada no Python3 possibilita aos usuários visualizar o progresso de um loop. Ela exibe em forma de uma barra de progresso informações sobre a quantidade de elementos visitados. No projeto ela foi utilizada múltiplas vezes para exibir a progressão de captura de dados.

Sua integração a um loop existente é facilitada, visto que a biblioteca fornece uma função para uso fora de um loop. Além disso, ela fornece o tempo decorrido para a execução de um loop e a estimativa de tempo para a conclusão dessa execução.

2.7.5 JSON

O arquivo resultante da execução pode adotar diversos formatos. Para a utilização posterior e possibilitar que manipulações possam ser realizadas, é importante escolher um tipo de arquivo que facilite a leitura e o tratamento dos dados.

O JavaScript Object Notation (JSON) [15] é um arquivo que fornece uma estrutura de dados simples e com alta legibilidade. Seus arquivos são marcados com a extensão .json. O Python3 fornece diversas bibliotecas nativas para importar ou exportar os dados para um JSON mantendo acentuação ou pontuação desses dados.

Outro fator para o uso desse formato é que ele independe da linguagem de programação utilizada no algoritmo. Ele permite que seus dados sejam importados para outras linguagens sem a necessidade de muitos tratamentos aos dados. Além disso, apresenta um formato leve, permitindo que múltiplos dados sejam salvos em um arquivo de tamanho pequeno e apresenta facilidade ao ser enviado a um *front-end*.

Esse formato é muito difundido no uso em navegadores web por seu suporte nativo e também em APIs web, o que garante continuidade do projeto em caso de mudanças para outras soluções de *back-end* ou *front-end*.

2.7.6 Pandas

A biblioteca Pandas [16] é muito conhecida por seu uso em análise de dados no Python3. No projeto, ela está sendo usada para a leitura de JSON e sua conversão para um DataFrame. Os dados contidos no JSON podem ser utilizados como base de dados para criar um arquivo do tipo Excel, entretanto, para isso, é preciso realizar a transposição desse DataFrame. Essa transposição é a troca entre linhas e colunas, transformando em colunas e linhas. Essa operação é voltada ao arquivo de saída do tipo Excel, ela é melhor explicada no Capítulo 3. Além disso, a biblioteca Pandas também é usada para salvar o DataFrame transposto para um arquivo do tipo Excel.

O próximo capítulo trata do desenvolvimento do *back-end* do sistema SARUE.

Capítulo 3

Desenvolvimento

Após a identificação dos principais requisitos para o desenvolvimento e a escolha dos softwares e bibliotecas a serem utilizados, versões preliminares do sistema foram desenvolvidas.

Inicialmente, o sistema tinha como intuito obter apenas o código e o nome das ações cadastradas no ano de 2020. Essa parte do sistema servia como um teste para verificar se era possível obter alguns dados através da biblioteca escolhida.

Após essa fase, novas funcionalidades ao sistema foram incorporadas, como a ação de abrir todas as informações sobre uma ação de extensão e obter parte dos dados dela. Ao mesmo tempo, foi implementada a opção de filtrar as ações na página de listar ações.

No decorrer da implementação, constatou-se que o tempo necessário para acessar toda página de extensão e obter todos os dados de forma sequencial era muito longo. Por isso, parte do desenvolvimento foi interrompido com o intuito de desenvolver formas para acelerar a obtenção dos dados. O Capítulo 4, apresenta os valores de tempo de execução iniciais como parte dos resultados.

O sistema SARUE pode ser dividido em duas fases de desenvolvimento: a primeira é um *back-end*, que é responsável por obter todos os dados de extensão do SIGAA e realizar um cálculo ou pré-cálculo dos indicadores; a segunda é um *front-end* voltado a exibir de forma gráfica os indicadores calculados.

A etapa do *front-end* e das estratégias de gestão empregadas no desenvolvimento do sistema SARUE não são descritas nesse documento e podem ser encontradas em [17].

Esse documento foca no desenvolvimento do *back-end*, nos cálculos dos indicadores e na avaliação e melhoria do desempenho.

O diagrama de funcionamento do *back-end* do sistema SARUE, mostrado na Figura 3.1, trata desde a obtenção da credencial do usuário, até o arquivo com indicadores já calculados. É possível verificar o uso de três diferentes Crawlers, explicados a seguir. Essa divisão se deve principalmente às diferentes possibilidades de uso do sistema SARUE

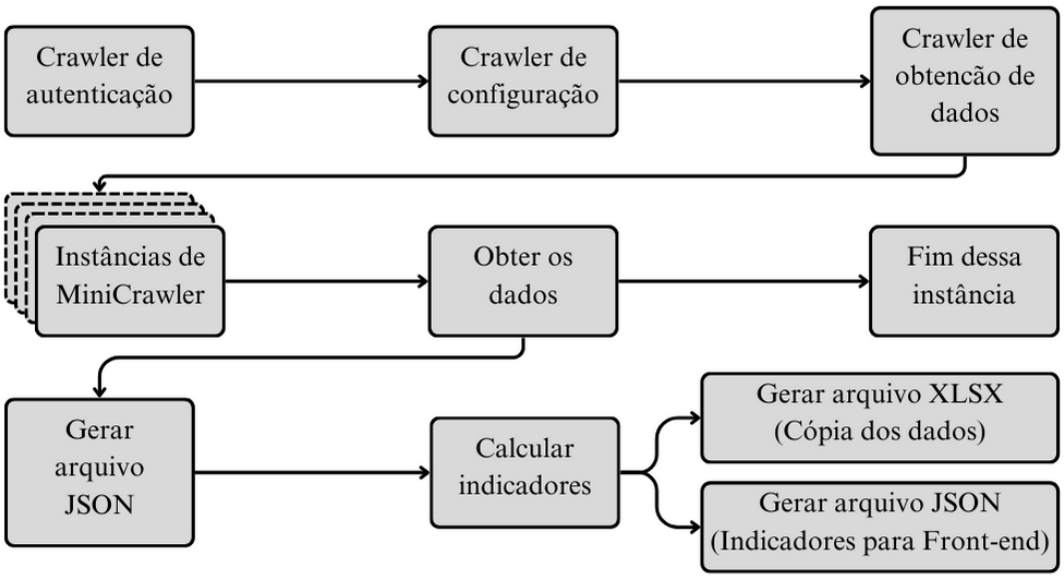


Figura 3.1: Diagrama do fluxo de execução do backend do SARUE.

implementadas. A fase de levantamento de requisitos não gerou um consenso sobre como o sistema seria utilizado e por isso, diferentes usos foram considerados e implementados.

3.1 Locais de funcionamento do sistema

O sistema SARUE pode ser executado como um sistema local na máquina do usuário ou a partir de outro dispositivo. No caso da execução local ela pode ser realizada sobre demanda ou sobre agendamentos da execução do script. No caso da execução a partir de outro dispositivo, seu funcionamento é realizado sobre agendamento.

No caso da execução local, a primeira possibilidade é uma atualização sobre demanda, onde o usuário deve iniciar a aplicação, inserir sua credencial por meio da instância de autenticação, esperar o script de obtenção de dados e então visualizar esses indicadores; a segunda opção é deixar essa credencial armazenada no sistema e então realizar o agendamento da execução do script. Uma vez que existe uma credencial salva, é possível configurar o algoritmo para executar o script e o cálculo dos indicadores sem a necessidade de interferência de um usuário. Nesse caso, os arquivos resultantes do algoritmo estão sempre atualizados, possibilitando ao usuário iniciar a visualização de forma imediata.

Existe também a possibilidade do uso do sistema sem acesso à credencial desse usuário. Durante uma reunião com a Secretaria de Tecnologia da Informação da UnB, foi levantada a possibilidade de usar uma credencial vinculada a um responsável pelo sistema SARUE. Em outras palavras, ocorreu a sugestão de poder utilizar uma credencial que fornecesse

acesso ao sistema SIGAA, mas não pertença diretamente a um usuário. Essa possibilidade foi cogitada principalmente no uso do sistema SARUE em um outro dispositivo.

Tratando do uso em outro dispositivo, seu funcionamento é realizado mantendo o *back-end* em outra máquina, a qual é responsável por iniciar o script e gerar os arquivos resultantes dessa execução. O ideal para o uso nessa situação é através do armazenamento de uma credencial do usuário dentro do sistema. Isso permite que o script seja executado de forma agendada, fazendo com que os dados sempre estejam disponíveis e atualizados.

Após a execução, independentemente do tipo de credencial inserida, o sistema sempre gera dois arquivos que podem ser incorporados ao *front-end*.

3.2 Gerenciamento de dependências

Como mostrado no capítulo anterior, esse sistema depende de algumas bibliotecas não nativas da linguagem para seu funcionamento. Para facilitar as instalações e atualizações, o sistema conta com um algoritmo voltado a resolver essas dependências.

Por meio de conhecimento das bibliotecas utilizadas, o sistema verifica e atualiza o gestor de pacotes `pip`[11] do Python3[10]. Após esse procedimento, por meio do `pip` o algoritmo realiza a instalação ou atualização das bibliotecas usadas no projeto.

Seu funcionamento é realizado antes da importação de qualquer biblioteca ao algoritmo, garantindo que no momento da execução do script de obtenção de dados, essas bibliotecas estão disponíveis e atualizadas.

3.3 Funcionamento do Crawler de autenticação

O sistema SARUE possui um script voltado exclusivamente a autenticar ou obter a credencial de um usuário. Esse script utiliza uma instância do navegador visível ao usuário para verificar se ele possui uma credencial válida para ser utilizada no sistema.

O Crawler de autenticação do sistema pode ser executado de três modos diferentes de execução. A primeira delas é iniciando uma instância do navegador que permite ao usuário se autenticar no sistema SIGAA e então esse script captura a credencial do usuário para o uso durante a execução do sistema SARUE. Essa credencial é descartada após o término da execução. A segunda forma é empregada para a atualização da credencial salva no sistema. Como o sistema permite que uma credencial seja salva, esse módulo de autenticação é utilizado para adicionar ou atualizar a credencial salva. Esse script inicia uma instância do navegador de forma visível ao usuário para que ele possa inserir suas credenciais como se estivesse se autenticando na plataforma SIGAA. O script, nesse caso, atua realizando a captura da credencial desse usuário e, caso essa credencial seja

válida, ela é armazenada. Por fim, a última forma de execução é através da verificação da existência de uma credencial salva no sistema e o carregamento dessa para variáveis locais dentro da execução do sistema.

Essas três implementações foram mantidas no projeto final, pois, através delas, é possível executar o sistema SARUE de forma local ou por meio de um servidor.

3.4 Funcionamento do Crawler de configuração

Existem diferentes tipos de usuário no sistema SIGAA. Um usuário pode ter o perfil de discente, docente, técnico, coordenador de extensão, diretor, chefe, entre outros. Como o script de obtenção dos dados executa a partir do perfil do usuário autenticado, isto é, com base na credencial obtida pelo Crawler de autenticação, cada usuário possui diferentes filtros na página de "consultar ações de extensão", no site do SIGAA. A Figura 3.2 ilustra essa diferença com dois perfis diferentes.

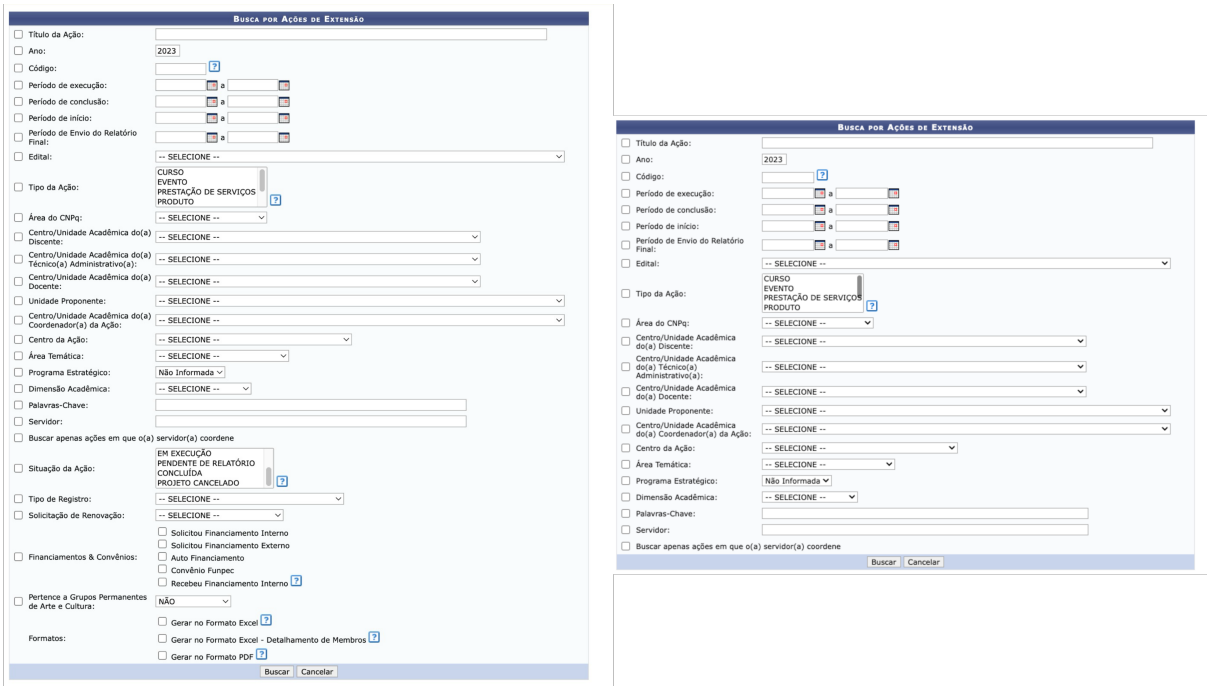


Figura 3.2: Comparativo entre página de listar ações de extensão no perfil de coordenador de extensão e discente.

No início do projeto, para verificar a quantidade de ações contidas em uma página, o script obtinha o valor de ações exibidas da página. Na Figura 3.3 é possível verificar esse valor.

Entretanto, por dificuldades voltadas à biblioteca Selenium, usada para simular a navegação pela página, procurar um elemento que pode não existir resultava em uma

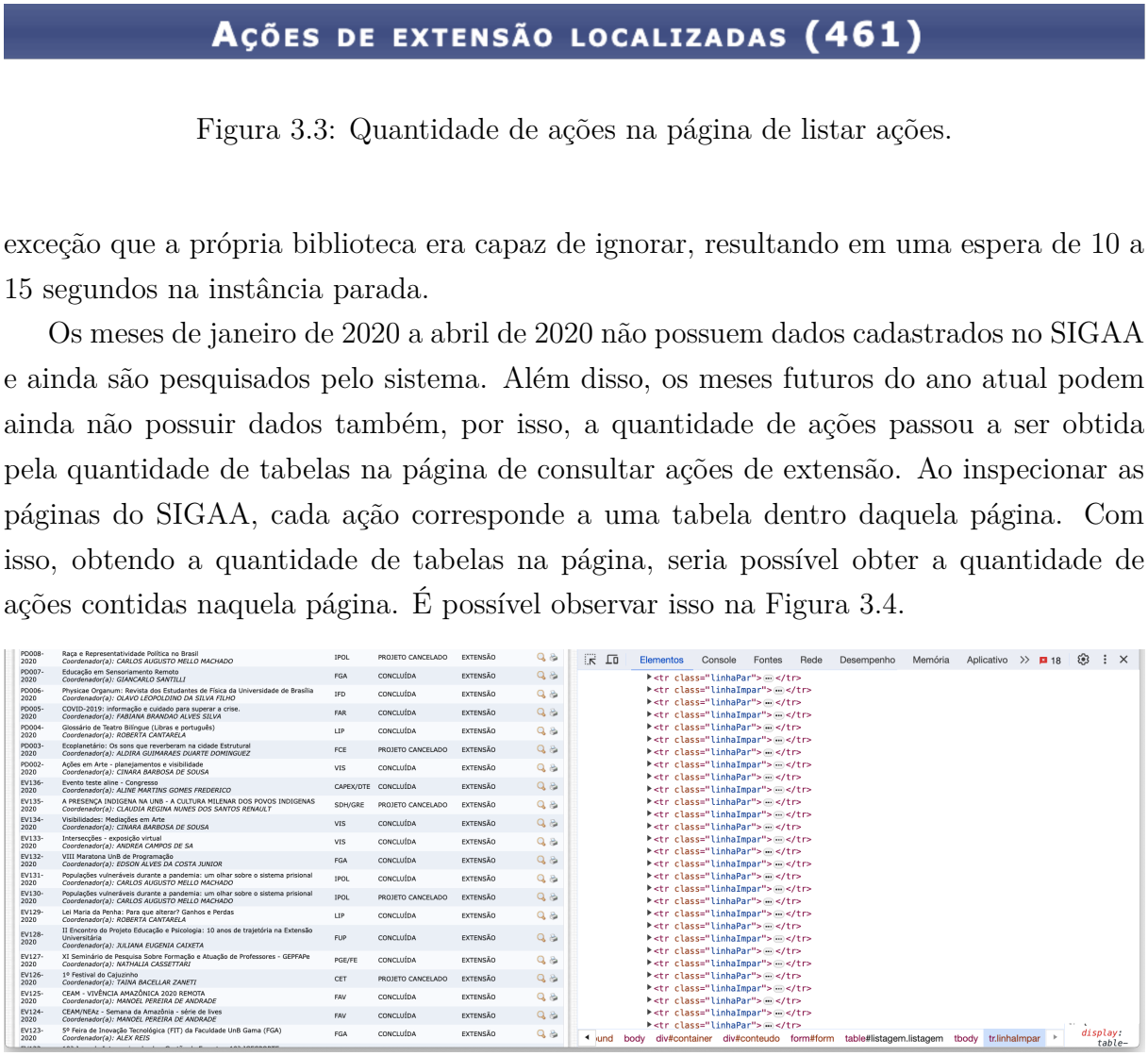


Figura 3.3: Quantidade de ações na página de listar ações.

exceção que a própria biblioteca era capaz de ignorar, resultando em uma espera de 10 a 15 segundos na instância parada.

Os meses de janeiro de 2020 a abril de 2020 não possuem dados cadastrados no SIGAA e ainda são pesquisados pelo sistema. Além disso, os meses futuros do ano atual podem ainda não possuir dados também, por isso, a quantidade de ações passou a ser obtida pela quantidade de tabelas na página de consultar ações de extensão. Ao inspecionar as páginas do SIGAA, cada ação corresponde a uma tabela dentro daquela página. Com isso, obtendo a quantidade de tabelas na página, seria possível obter a quantidade de ações contidas naquela página. É possível observar isso na Figura 3.4.

Figura 3.4: Exibição do DevTools da página de exibir ações de extensão.

A quantidade de tabelas obtidas pelo algoritmo do script é dado por:

$$quantidade\_total = quantidade\_filtros + quantidade\_acoes \tag{3.1}$$

Logo, para obter apenas a quantidade de ações contidas em uma página, é necessário saber a quantidade de filtros existentes para aquele usuário. Como são muitos perfis de usuários e esses filtros podem variar, realizar uma pesquisa e contar a quantidade de filtros não é uma opção viável. Dessa forma, o Crawler de configuração se faz necessário. Esse script executa antes do início do Crawler de obtenção de dados e realiza a pesquisa de ações em um mês vazio, como o mês de janeiro de 2020 que não possui ações cadastradas. Através da quantidade de tabelas contidas nessa página é obtido um *offset* que é a quantidade de tabelas que são filtros na página de extensão. Esse *offset* é utilizado no Crawler

de obtenção de dados para obter a quantidade de ações em uma página. Com esse *offset* calculado, é possível obter a quantidade de ações em uma página para qualquer perfil do sistema SIGAA.

3.5 Funcionamento do Crawler de obtenção de dados

O Crawler de obtenção de dados é a parte mais importante e extensa do funcionamento do script, uma vez que tem como intuito obter as informações de cada ação cadastrada no sistema SIGAA.

Antes de apresentar o funcionamento do Crawler de obtenção de dados, é necessário explicar algumas particularidades do sistema SIGAA. A página de listar ações de extensão exibe as ações solicitadas pela busca, mas, por limitações técnicas, impede a exibição de todas as ações cadastradas na plataforma, quando são solicitadas mais de 1000 ações ou quando não foram especificados filtros para a busca. As Figura 3.5 e Figura 3.6 respectivamente mostram esses erros.

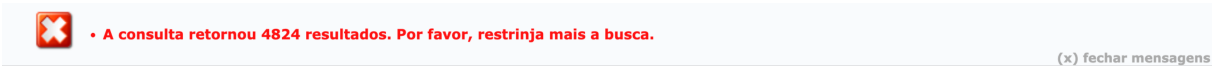


Figura 3.5: Mensagem de erro SIGAA - Retornou muitos resultados.

Figura 3.6: Mensagem de erro SIGAA - Nenhum filtro aplicado.

No momento de execução desse relatório, existiam mais de 8500 ações cadastradas na plataforma. Dessa forma, para obter todos os dados, foram acrescentadas, ao sistema, opções de selecionar os filtros contidos na página de extensão do SIGAA.

Inicialmente, foi adicionado o filtro por data prevista de início de ações. Esse filtro foi escolhido porque toda ação, independentemente do status de execução dela, seja cancelada, aprovada com ou sem recursos, concluída ou pendente de relatórios, possui essa data. Dessa forma, realizar divisões com base na data de início da ação garantiria a completude dos dados obtidos e que uma mesma ação não fosse pesquisada mais de uma vez, evitando repetições e pesquisas desnecessárias.

Na filtragem baseada na data de início, o objetivo do script era obter todos os dados de um determinado ano. Assim, para obter todos os dados de um ano, eram realizadas pesquisas à página do SIGAA, onde cada uma era destinada a obter os dados de um mês em específico.

Durante o desenvolvimento do projeto, essa solução era suficiente para obter os dados de todas as ações cadastradas. Todos os meses que possuíam ações, a sua quantidade não era maior que 1000, limite estabelecido pela plataforma SIGAA. Entretanto, em função do edital da Semana Universitária de 2023, que aconteceu entre os dias 25 e 29 de setembro de 2023, mais de 1000 ações foram cadastradas com data de início nessa semana. A primeira solução proposta foi subdividir o mês de setembro em partes menores, tratando por semanas ao invés de um mês. Mesmo assim, essa solução não seria viável. A próxima solução proposta foi voltada ao tipo de ação, em que o script deveria realizar a filtragem por curso, evento, prestação de serviço, produto, programa ou projeto. Ainda assim, por uma particularidade do edital da Semuni, das 1162 ações cadastradas nessa semana, 1117 ações eram eventos.

Por isso, ainda com o intuito de escolher uma filtragem que garantisse ou até mesmo reduzisse a quantidade de repetições em uma única semana, foi escolhido o filtro Área do CNPq. Esse campo também tem cadastro obrigatório no cadastro de uma ação e é exclusivo por ação. Em outras palavras, uma ação só pode ter uma área do CNPq cadastrada. A partir do momento em que foi implementada a filtragem por área do CNPq, o script foi capaz de realizar divisões de cada mês em outras nove subcategorias.

O uso da filtragem pela área do CNPq não é usado em todos os meses. A grande maioria dos meses cadastrados no sistema SIGAA possui menos de 1000 ações por mês. O uso da filtragem por mês e pela área do CNPq é obrigatório para o mês de setembro de 2023 e foi usado também em outros meses. A escolha desses meses foi condicionada para permitir que instâncias livres pudessem ajudar a concluir esses meses mais rapidamente. Em outras palavras, meses que possuem mais de 500 ações podem ser filtrados apenas pela data de início, mas, por questões de performance, foram adicionadas ambas as filtragens.

3.6 Funcionamento do MiniCrawler de obtenção de dados

O MiniCrawler é o nome dado a uma das muitas possíveis instâncias do que seria o Crawler de obtenção de dados. Esse funcionamento trata-se da divisão do Crawler de obtenção de dados em instâncias de MiniCrawlers.

No Capítulo 4, serão exibidas comparações que justificam o uso de diversas instâncias do script para acelerar a obtenção dos dados.

Foram desenvolvidas e mantidas no projeto duas formas de paralelizar a execução de instâncias de um MiniCrawler. Em uma delas, chamada paralela, cada mês deve iniciar uma nova instância, realizar a autenticação, navegar a página de extensão, filtrar e obter esses dados. Dessa forma, o Crawler de obtenção de dados deveria apenas sincronizar

as instâncias e impedir o funcionamento de todas ao mesmo tempo. A outra é uma abordagem também paralela, mas que utiliza técnicas de programação concorrente, e por isso chamada de MiniCrawler Concorrente. Nela, as N instâncias do navegador são inicializadas e mantidas abertas para futuras pesquisas. Essa abordagem utiliza o Crawler de obtenção de dados para gerir as instâncias disponíveis e seu funcionamento.

3.6.1 MiniCrawler Paralelo

O funcionamento do MiniCrawler paralelo pode ser visto na Figura 3.7. Essa figura fornece uma abstração do funcionamento dessa parte do script. Por padrão, toda instância é iniciada, autenticada e então realiza a navegação até a página de listar ações. Nesse momento é realizada a filtragem das ações e cada ação é copiada uma a uma. Após isso, essa instância é encerrada e, com base no máximo de ações simultâneas, outra instância pode iniciar o processo de procura.

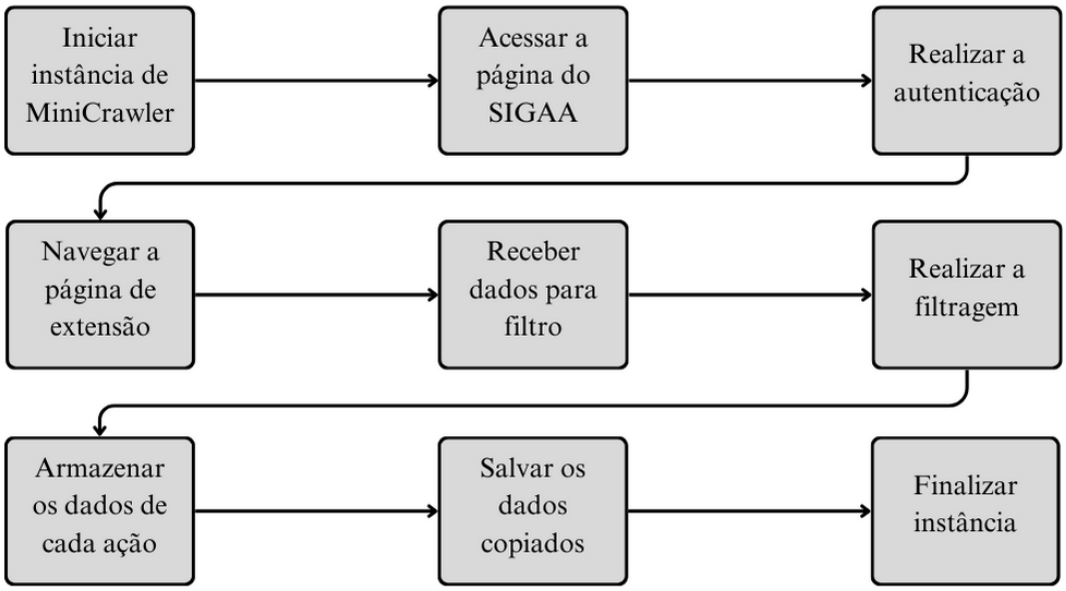


Figura 3.7: Diagrama do fluxo de execução de MiniCrawler paralelo.

Apesar de ser uma solução não muito sofisticada do ponto de vista de memória ou de repetições de instruções, como autenticar uma instância e navegar a página de listar ações, essa solução mostrou um caminho muito eficiente para reduzir o tempo de execução do script.

Para decidir quais instâncias realizariam determinadas buscas com base nos meses, foram implementadas divisões baseadas em conjunto de meses. Com isso, determinada instância deveria pesquisar um conjunto predefinido de meses.

Inicialmente foi incorporada a divisão anual, e cada instância era responsável por obter os dados de um determinado ano. Essa solução tinha um tempo de execução muito próximo ao necessário para obter os dados do maior ano, visto que instâncias que realizassem a procura em anos que possuem uma quantidade menor de ações cadastradas tendiam a terminar antes do que as que possuem maiores quantidades de ações.

A Equação 3.2 exibe de forma aproximada o tempo de execução na abordagem paralela usando divisão por ano,

$$T_{tpa} \approx \max_{i=1}^{n} T_{i\text{ano}}, \tag{3.2}$$

onde o tempo total de execução dado por T_{tpa} é estimado de forma aproximada pelo tempo de execução do maior ano. Vale ressaltar que essa equação é baseada na disponibilidade de uma instância para cada período. Ou seja, na existência de três anos a serem pesquisados, pelo menos três instâncias devem ser utilizadas.

Posteriormente, o projeto recebeu a possibilidade de realizar divisões semestrais. Nesse caso, cada instância ficou responsável por obter os dados de um determinado semestre. Parecido como na divisão anual, essa solução apresentava um tempo de execução muito próximo ao tempo de execução linear do semestre com mais ações cadastradas.

Ainda como uma aproximação, a Equação 3.3 demonstra o tempo de execução com essa divisão,

$$T_{tps} \approx \max_{i=1}^{n} T_{i\text{semestre}}, \tag{3.3}$$

na qual o tempo total dado por T_{tps} é estimado através do tempo de execução do maior semestre.

Ao mesmo tempo, para possibilitar o uso de mais instâncias e melhorar a divisão dos meses, foram introduzidas ao algoritmo duas novas possibilidades: a primeira delas era com foco em divisões trimestrais e a segunda, com base em quadrimestres. Essas soluções foram implementadas juntas, pois usando a solução trimestral, um ano seria dividido em quatro partes.

Com isso, pensando em uma atualização grande de dados, como por exemplo, de 2020 a 2023, sendo um intervalo de quatro anos, existiriam dezesseis partes que deveriam ser pesquisadas e, com base em testes, uma máquina só é capaz de lidar com no máximo doze instâncias simultâneas. Dessa forma, o tempo de execução com o uso da divisão quadrimestral, e tendo instâncias o suficiente para permitir que uma instância pudesse lidar com apenas um quadrimestre de um ano, teríamos a aproximação dada por

$$T_{tpq} \approx \max_{i=1}^{n} T_{i\text{quadrimestre}}, \tag{3.4}$$

onde o tempo total `T_tpq` é resultado do tempo de execução linear do quadrimestre com mais ações cadastradas.

Já na execução trimestral, considerando um conjunto de divisões menor ou igual à quantidade de instâncias disponíveis, teríamos a equação

$$T_{tpt} \approx \max_{i=1}^{n} T_{i\text{trimestre}}, \tag{3.5}$$

na qual `T_tpt`, por aproximação, é dada pelo tempo de execução do trimestre com mais ações cadastradas.

Entretanto, pensando em um conjunto com quatro anos na divisão trimestral e principalmente sem a possibilidade de utilizar dezesseis instâncias, o tempo de execução não pode ser perfeitamente estimado, visto que depende de quais instâncias realizariam a obtenção das divisões restantes.

Para a implementação de forma paralela, foi utilizada a biblioteca *concurrent.futures*, que possibilita o uso de funções como a *ThreadPoolExecutor*. Através dela é possível realizar um *loop* de forma paralela.

Listagem 3.1: MiniCrawler paralelo

```
instances = {}
with ThreadPoolExecutor(max_workers=MAX_THREADS) as executor:
    for year in range(START_YEAR, END_YEAR+1):
        for i in range(3):
            quarter_instance = MiniCrawler()
            instances[str(year)+"_"+str(i)] = quarter_instance
            executor.submit(instances[str(year)+"_"+str(i)].run, self.perfil,
                username, password, type_search, year, None, i+1)
```

Com base no código 3.1, assumindo as variáveis START_YEAR e END_YEAR como 2020 e 2023 respectivamente, o trecho de código responsável por iniciar uma instância de um navegador e executá-lo com base nos parâmetros a ele fornecidos, é realizado 3 vezes por ano, logo, 12 vezes no período de 2020 a 2023.

O uso de *ThreadPoolExecutor* antes dos laços de repetição permite que mais de uma execução desse laço possa ser realizada ao mesmo tempo. Vale ressaltar que essa implementação inicializa uma instância do navegador por período de procura dos dados. Sendo assim, são inicializadas 12 instâncias do navegador mesmo que a quantidade máxima de

trabalhadores seja interior. Logo, essa implementação utiliza mais recursos de memória do computador utilizada quando em comparação à execução sequencial.

3.6.2 MiniCrawler Concorrente

Buscando aprimorar as estratégias de segmentação temporal adotadas pelo MiniCrawler Paralelo, surgiu a ideia de realizar uma divisão com base nos meses. No entanto, usar a abordagem anterior não se mostrou ideal para gerenciar ou coordenar a execução simultânea de múltiplas instâncias.

Neste ponto, introduzimos o que estamos chamando de MiniCrawler Concorrente. Essa abordagem assemelha-se muito ao MiniCrawler Paralelo, mas, para a pesquisa de um período de tempo, a execução deve competir pelo uso de uma instância. Ele é tratado assim pois apesar de ser majoritariamente paralelo, o sistema deve concorrer para o uso de uma instância do navegador. Seu funcionamento pode ser dividido em várias etapas conforme a Figura 3.8.

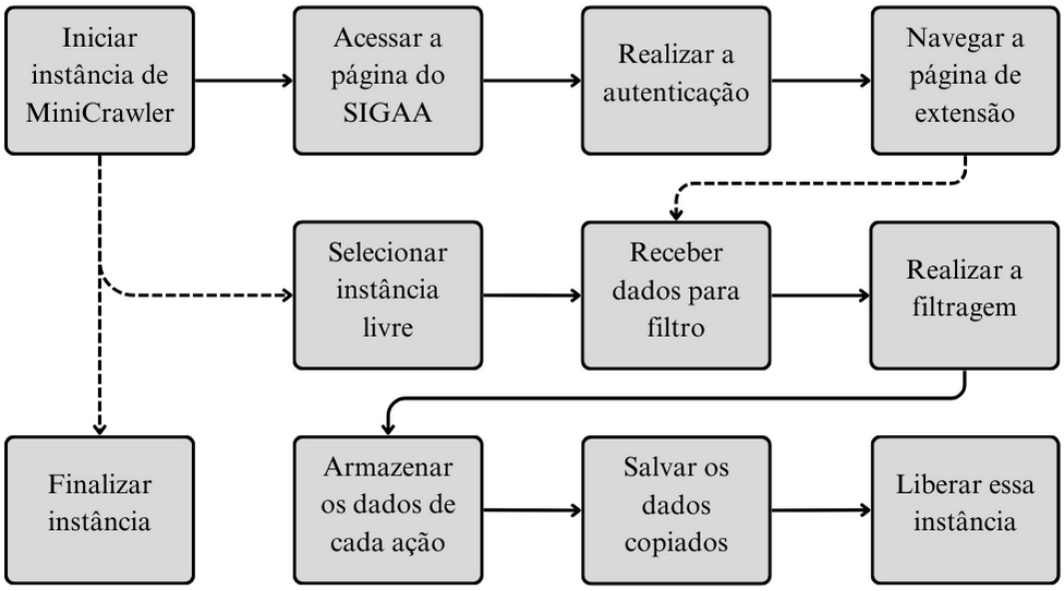


Figura 3.8: Diagrama do fluxo de execução de MiniCrawler concorrente.

Na Figura 3.8 é apresentado o trajeto percorrido pelo script. Primeiramente, é criada uma lista com elementos. Esses elementos são gerados pela concatenação do ano e do mês. No entanto, se esse mês estiver numa lista de meses a serem divididos, a concatenação incluirá não apenas o ano e o mês, mas também a área do CNPq a ser pesquisada por uma instância do MiniCrawler.

Listagem 3.2: Criação da lista para o MiniCrawler concorrente

```
lista = []
for year de START_YEAR até END_YEAR (inclusive):
    for month de FIRST_MONTH_OF_YEAR até LAST_MONTH_OF_YEAR (inclusive):
        year_month = concatenar(str(year), '/', str(month))
        month_year = concatenar(str(month), '/', str(year))

        se month_year está em SPECIAL_DATE então:
            para cada cnpq em AREA_CNPq faca:
                adicionar a lista a concatenacão de year_month, '/', cnpq
        senão:
            adicionar a lista a concatenacão de year_month
```

Esta lista é gerada através do algoritmo 3.2. Assumindo os valores das variáveis START_YEAR e END_YEAR como 2020 e 2022 respectivamente, teríamos inicialmente 36 elementos na lista. Para otimizar a pesquisa de meses específicos, durante esse intervalo, dois meses empregam a divisão por CNPq, o que resulta em um total de 52 filtros a serem aplicados durante este período de pesquisa.

Listagem 3.3: Inicialização das N instâncias do MiniCrawler concorrente

```
Criar um ThreadPoolExecutor com um número máximo de threads igual a MAX_THREADS
Criar uma lista chamada "instances"

Para cada _ no intervalo de 0 a MAX_THREADS (não incluso):
    Adicionar uma nova instancia de MiniCrawlerConcurrent com os argumentos
        self.username e self.password a lista "instances"

Exibir uma barra de progresso com a descricão desc, utilizando o tqdm, com o formato
    '{'Loggin in'} - {elapsed} {bar} {n_fmt}/{total_fmt} - {percentage:.0f}%', e um
    número de colunas igual a SIZE_TERMINAL
```

Após a definição da lista, são iniciadas, de forma paralela, N instâncias do MiniCrawler, conforme algoritmo mostrado por 3.3, permitindo que várias instâncias possam ser criadas ao mesmo tempo. No início de cada instância, cada uma delas passa por um processo de aceitar os *cookies* da página, pela autenticação, por meio da credencial obtida e pela navegação até a página de extensão. Após concluir o processo de inicialização dessas instâncias, elas são mantidas abertas para a próxima fase.

Listagem 3.4: MiniCrawler concorrente

```
função get_instance_with_wait():
    enquanto o comprimento de "instances" for igual a 0:
        aguarde 1 segundo
    instance = remover o primeiro elemento de "instances"
    retornar instance
```

```
6
7  função run_instance(username, password, year_month_cnpq):
8      lista_year_month_cnpq = dividir year_month_cnpq por '/'
9      year, month = converter os primeiros dois elementos de lista_year_month_cnpq para
           inteiros
10     cnpq = lista_year_month_cnpq[2] se o comprimento de lista_year_month_cnpq for
           igual a 3, caso contrário, None
11
12     instance = chamar a função get_instance_with_wait()
13     instance.run(self.offset, year, month, cnpq)
14     adicionar instance ao final de "instances"
15     retornar year_month_cnpq
```

O algoritmo 3.4 engloba duas funções interconectadas que coordenam a execução concorrente de operações. A função denominada `get_instance_with_wait`, desempenha o papel crucial de aguardar até que uma lista denominada "instances"contenha pelo menos uma instância antes de proceder para retirar e retornar a primeira instância disponível. Isso assegura que as operações subsequentes possam ser conduzidas de forma assíncrona, otimizando o aproveitamento de recursos.

A segunda função, denominada `run_instance`, desempenha um papel fundamental na execução das operações propriamente ditas. Ela recebe diversos parâmetros, incluindo um nome de usuário, senha, e uma string que representa ano, mês e, opcionalmente, um parâmetro adicional. Essa função faz uso da primeira função, `get_instance_with_wait`, para garantir que uma instância esteja pronta para a execução. Em seguida, realiza uma operação específica sobre a instância obtida e, finalmente, reintegra a instância de volta à lista "instances".

Nessa segunda função vale ressaltar atribuição de valor a variável `cnpq`. Com base no comprimento da variável `lista_year_month_cnpq`, se existem 3 elementos após a separação por "/", então o último elemento é a área do CNPq ao qual deve ser aplicado o filtro. Caso contrário, `None` deve ser usado como o valor da variável `cnpq`.

Listagem 3.5: MiniCrawler concorrente

```
1  Criar um ThreadPoolExecutor com um número máximo de threads igual a MAX_THREADS
2  Criar um dicionário chamado "futures"
3
4  Para cada year_month_cnpq em lista:
5      Adicionar uma nova tarefa ao executor para chamar a funcao run_instance com os
           argumentos self.username, self.password, e year_month_cnpq
6      Associar a tarefa ao year_month_cnpq no dicionário "futures"
7
8  Para cada future em tarefas concluídas (as_completed(futures)):
```

```
    year_month_cnpq = obter o valor associado ao future no dicionário "futures"
    Tentar obter o resultado do future
    Se ocorrer uma excecao, adicionar uma entrada ao log indicando o year_month_cnpq e
        a excecao
    Senão, adicionar uma entrada ao log indicando que o year_month_cnpq foi processado
        com sucesso

Para cada instance em instances:
    Chamar a função quit() na instance
```

Esta nova etapa envolve a seleção de um elemento na lista e a execução simultânea da função *run_instance* por meio da `ThreadPoolExecutor` da biblioteca `concurrent.futures`, conforme algoritmo 3.5. Enquanto a função para executar uma instância está em andamento, o elemento da lista é convertido no filtro a ser aplicado. O script então seleciona ou aguarda uma instância disponível, envia as condições de filtragem e espera a conclusão dessa ação, de forma a verificar a resposta da instância e determinar se os dados foram corretamente obtidos.

Cada instância é mantida ativa até que não existam elementos não pesquisados na lista. Por exemplo, supondo que existam seis instâncias ativas livres e somente cinco elementos na lista, cada instância irá pesquisar um elemento da lista e instância restante será encerrada.

Para isso, o controle sobre o funcionamento e a sincronização dessas instâncias são realizados por uma classe que define os elementos da lista, a ordem que devem ser executados e as instâncias disponíveis.

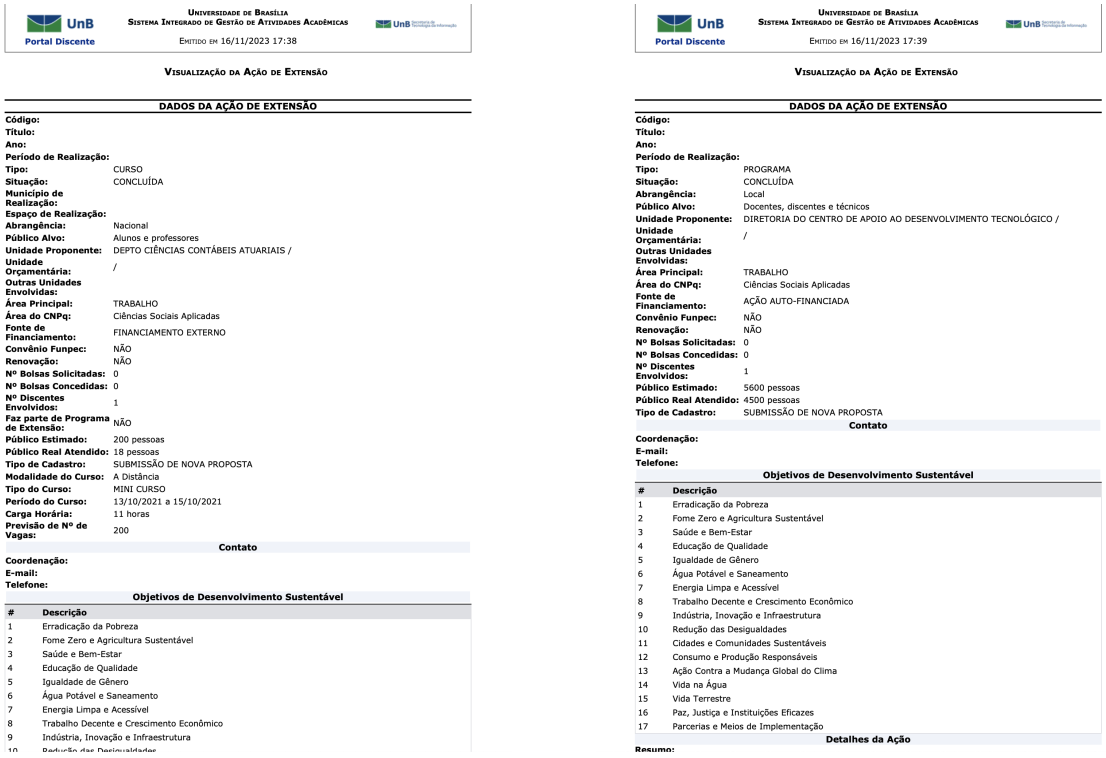
Além da `ThreadPoolExecutor`, outras funções da biblioteca `concurrent.futures` foram utilizadas. Como por exemplo, a função `as_completed`, que informa os elementos da lista que já foram visitados. Através desses elementos, é possível verificar cada um deles por meio da função `.result()`, obtendo as exceções geradas por cada elemento ou se foram executadas corretamente.

No caso da execução nesse modo, estimar uma função que se aproxime do tempo de execução não é fácil. Os tempos obtidos por esse método de execução são analisados no Capítulo 4. Mesmo assim, a tarefa realizada por cada instância pode não ser a mesma em execuções sucessivas. Por exemplo, no caso da execução através de um MiniCrawler paralelo, adotando o método de divisão de tempo trimestral, é possível determinar qual instância realiza a pesquisa em determinado mês. No caso do uso do MiniCrawler concorrente, para pesquisar um determinado mês, o sistema obtém uma instância não predefinida, na qual o script concorre pelo uso de uma instância. A instância usada na pesquisa de um determinado mês no MiniCrawler concorrente não pode ser determinada. Por isso, não é possível estimar ou supor uma equação nesse método com base em execução

sequencial.

3.7 Obter os dados de extensão

Para obter o conteúdo das páginas de visualizar ações de extensão, foram realizadas pequenas alterações para cada tipo de ação. Existem campos e endereços *XPATHS* cadastrados nessas páginas que são padrões a qualquer tipo de ação, entretanto, existem campos que são específicos de determinado tipo. Na Figura 3.9 é possível visualizar a comparação entre as páginas de visualização de ação de extensão de um curso, a esquerda,e de um programa, a direita. Ainda é possível identificar alguns campos que não são constantes para todas as ações.



UnB Portal Discente

Universidade de Brasília
Sistema Integrado de Gestão de Atividades Acadêmicas

Emitido em 16/11/2023 17:38

Visualização da Ação de Extensão

DADOS DA AÇÃO DE EXTENSÃO

Código:	
Título:	
Ano:	
Período de Realização:	
Tipo:	CURSO
Situação:	CONCLUÍDA
Município de Realização:	
Espaço de Realização:	
Abrangência:	Nacional
Público Alvo:	Alunos e professores
Unidade Proponente:	DEPTO CIÊNCIAS CONTÁBEIS ATUARIAIS /
Unidade Orçamentária:	/
Outras Unidades Envolvidas:	
Área Principal:	TRABALHO
Área do CNPq:	Ciências Sociais Aplicadas
Fonte de Financiamento:	FINANCIAMENTO EXTERNO
Convênio Funpec:	NÃO
Renovação:	NÃO
Nº Bolsas Solicitadas:	0
Nº Bolsas Concedidas:	0
Nº Discentes Envolvidos:	1
Faz parte de Programa de Extensão:	NÃO
Público Estimado:	200 pessoas
Público Real Atendido:	18 pessoas
Tipo de Cadastro:	SUBMISSÃO DE NOVA PROPOSTA
Modalidade do Curso:	A Distância
Tipo do Curso:	MINI CURSO
Período do Curso:	13/10/2021 a 15/10/2021
Carga Horária:	11 horas
Previsão de Nº de Vagas:	200

Contato

Coordenação:	
E-mail:	
Telefone:	

Objetivos de Desenvolvimento Sustentável

#	Descrição
1	Erradicação da Pobreza
2	Fome Zero e Agricultura Sustentável
3	Saúde e Bem-Estar
4	Educação de Qualidade
5	Igualdade de Gênero
6	Água Potável e Saneamento
7	Energia Limpa e Acessível
8	Trabalho Decente e Crescimento Econômico
9	Indústria, Inovação e Infraestrutura
10	Redução das Desigualdades

UnB Portal Discente

Universidade de Brasília
Sistema Integrado de Gestão de Atividades Acadêmicas

Emitido em 16/11/2023 17:39

Visualização da Ação de Extensão

DADOS DA AÇÃO DE EXTENSÃO

Código:	
Título:	
Ano:	
Período de Realização:	
Tipo:	PROGRAMA
Situação:	CONCLUÍDA
Abrangência:	Local
Público Alvo:	Docentes, discentes e técnicos
Unidade Proponente:	DIRETORIA DO CENTRO DE APOIO AO DESENVOLVIMENTO TECNOLÓGICO /
Unidade Orçamentária:	/
Outras Unidades Envolvidas:	
Área Principal:	TRABALHO
Área do CNPq:	Ciências Sociais Aplicadas
Fonte de Financiamento:	AÇÃO AUTO-FINANCIADA
Convênio Funpec:	NÃO
Renovação:	NÃO
Nº Bolsas Solicitadas:	0
Nº Bolsas Concedidas:	0
Nº Discentes Envolvidos:	1
Público Estimado:	5600 pessoas
Público Real Atendido:	4500 pessoas
Tipo de Cadastro:	SUBMISSÃO DE NOVA PROPOSTA

Contato

Coordenação:	
E-mail:	
Telefone:	

Objetivos de Desenvolvimento Sustentável

#	Descrição
1	Erradicação da Pobreza
2	Fome Zero e Agricultura Sustentável
3	Saúde e Bem-Estar
4	Educação de Qualidade
5	Igualdade de Gênero
6	Água Potável e Saneamento
7	Energia Limpa e Acessível
8	Trabalho Decente e Crescimento Econômico
9	Indústria, Inovação e Infraestrutura
10	Redução das Desigualdades
11	Cidades e Comunidades Sustentáveis
12	Consumo e Produção Responsáveis
13	Ação Contra a Mudança Global do Clima
14	Vida na Água
15	Vida Terrestre
16	Paz, Justiça e Instituições Eficazes
17	Parcerias e Meios de Implementação

Detalhes da Ação

Resumo:

Figura 3.9: Comparativo entre a página de visualização de um curso e de um programa.

Para obter todos os dados da ação cadastrada nessa página, o sistema realiza pesquisa com base na posição em que um elemento se encontra na tela. Assim como na página de listar ações de extensão, essas páginas ao inspecionar os elementos da página, possuem os dados armazenados em tabelas pares e ímpares com os dados. Por isso, as funções responsáveis por armazenar os dados de cada página são baseadas no tipo de ação que está sendo pesquisada e, logo em seguida, em qual posição determinado dado deve ser encontrado.

Essa solução requer informações detalhadas sobre o que é esperado em cada página de cada tipo de ação. Mesmo assim, todos os elementos obtidos pelo script são condizentes com os elementos cadastrados na plataforma SIGAA.

3.8 Manter dados salvos no sistema

Como dito anteriormente, o desenvolvimento de soluções paralelas e concorrentes foi implementado com o objetivo de reduzir o tempo de execução do script. Além dessas técnicas, foi inserida no sistema gestão de dados com o objetivo de evitar que ações já concluídas fossem pesquisadas em todas execuções.

Para a implementação do script, o perfil de discente foi utilizado, que não permite a filtragem pelo status da ação.

Assim, para reduzir a quantidade de elementos pesquisados em cada execução, o sistema é capaz de executar o script com duas configurações diferentes. A primeira delas é a que visa obter os dados mais atuais do sistema. Essa configuração é voltada a realizar pesquisas no ano atual ou em períodos próximos a data atual. A outra forma de execução é voltada a realizar atualizações dos dados cadastrados no sistema. Apesar da maioria dessas ações não sofrer alterações, algumas delas podem ter seu status atualizado, seja por causa do cadastro de relatório ou por finalizar sua execução. Os resultados dessa execução do script são comparados com os dados armazenados no sistema e, se necessário, o sistema SARUE atualiza esses dados e gera uma nova base de dados. Essa base de dados pode ser atualizada com menos frequência que os dados que são obtidos pela execução padrão do sistema, pois a frequência de atualização desses dados é menor do que os dados do período atual.

Para obter todos os dados cadastrados, o sistema SARUE realiza a junção dos dados armazenados com a execução padrão do algoritmo, obtentendo dados que, para o ano atual, refletem o que está atualmente cadastrado e, para anos anteriores, refletem dados condizentes com a última execução de atualização da base de dados.

Usando como referência os anos de 2020 a 2022 como base dos dados, esse tratamento permite que mais de 4500 ações estejam armazenadas no sistema SARUE, reduzindo a quantidade de ações que devem ser frequentemente pesquisadas.

3.9 Outros tratamentos

3.9.1 Executar instâncias de forma invisível ao usuário

Os navegadores mais populares, como o Google Chrome[18], Microsoft Edge[19] e Mozilla Firefox[20] permitem que instâncias geradas por um script sejam executadas de forma invisível ao usuário. Essa solução permite a execução do algoritmo em segundo plano na máquina do usuário.

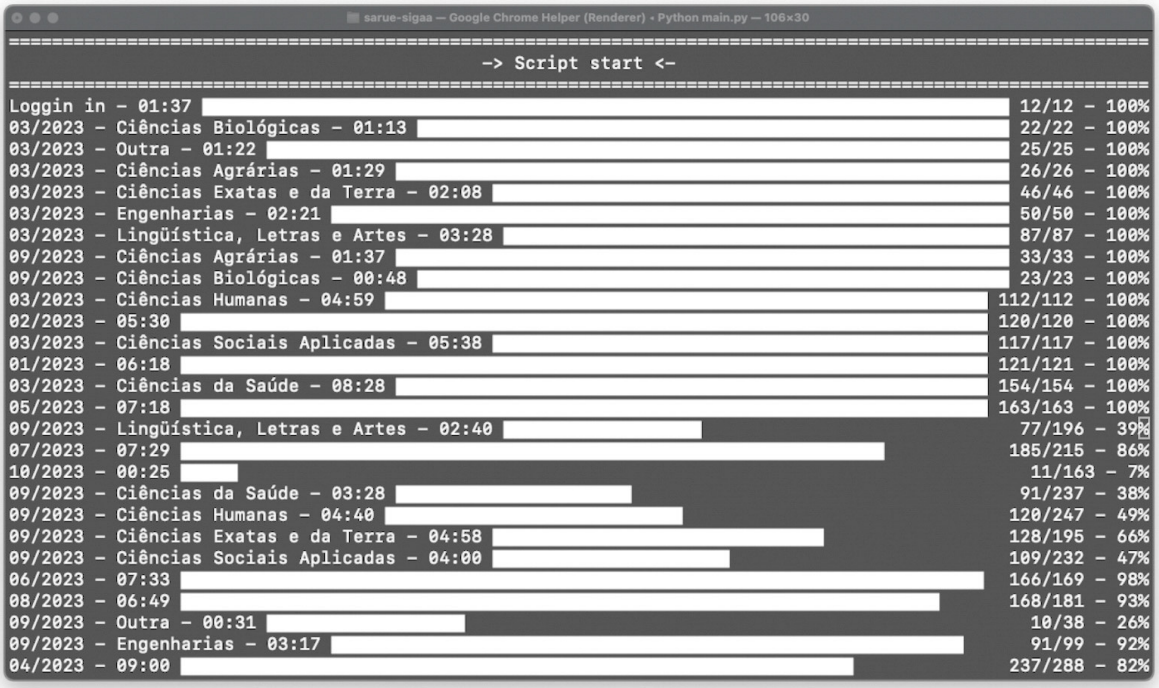


Figura 3.10: Mensagem no terminal exibidas ao usuário.

Para exibir ao usuário o progresso da obtenção dos dados, foi utilizada a biblioteca Tqdm. Essa biblioteca fornece uma barra de progresso exibindo os itens de um loop, onde cada loop é responsável por uma filtragem de arquivos. A Figura 3.10 exibe como é visível ao usuário o progresso do algoritmo de obtenção dos dados.

3.9.2 Uso de diferentes navegadores

Uma das principais vantagens da utilização da biblioteca Selenium é poder utilizar o algoritmo a partir de vários navegadores instalados na máquina do usuário. Nos testes realizados, o navegador Google Chrome[18] foi o que apresentou melhores tempos para a obtenção de dados. Os navegadores Microsoft Edge[19] e o Mozilla Firefox[20] também apresentaram bons tempos para a obtenção dos dados.

Na utilização de dispositivos com sistema operacional macOS, para o correto funcionamento do script, é necessária a instalação de um outro navegador. Por questões de segurança e privacidade dos dados, o navegador padrão desses dispositivos Safari[21] impede a inserção de dados em uma instância controlada por um algoritmo. A Figura 3.11 ilustra o erro indicado ao usuário.

Figura 3.11: Alerta do navegador sobre automação de navegação no Safari.

Além disso, o uso do navegador Safari[21] impede o algoritmo de ser executado sem a interface gráfica, exibindo todas as instâncias ao usuário durante a execução do script.

3.9.3 Obter todas as ações canceladas

Todo código de uma ação de extensão aprovada é composto por no mínimo dez caracteres. Os dois primeiros dígitos identificam o tipo daquela ação. Os números seguintes e que antecedem o caractere '-' representam o número ou a sequência dessa ação. Por fim, os últimos quatro dígitos representam o ano da ação. Por exemplo, o código da ação PJ002-2021 informa que a ação é um projeto, pois inicia com PJ, que foi a segunda ação do tipo projeto, por isso o 002, e que foi criada no ano de 2021.

O arquivo de saída JSON utiliza o código de uma ação como chave primária para armazenar os dados contidos dessa ação, de forma a facilitar a pesquisa por um elemento

único. A princípio, o código de uma ação deve ser único, entretanto, ações canceladas são mantidas no sistema SIGAA, e não possuem um código único.

Para solucionar isso e garantir que todas as ações fossem obtidas pelo script, quando encontra alguma ação que possui os caracteres 'xxx' no código, o sistema salva como chave primária a concatenação entre o código e o nome da ação, garantindo que as ações canceladas também façam parte do resultado do script. Essa solução resulta em chaves primárias como a Equação 3.6.

$$PJxxx - 2021\_Nome\_da\_ação\_de\_extensão \tag{3.6}$$

3.9.4 Redução do arquivo enviado ao *front-end*

O arquivo resultante da execução do script de obtenção de dados é baseado em uma cópia de todos os dados contidos no sistema SIGAA. Esse arquivo é constituído por um JSON onde existem diversos elementos salvos e a chave primária é dada pelo código de uma ação.

Como são obtidos dados como membros da equipe, e demais informações sobre o projeto, esse arquivo costuma ter um tamanho significativo, visto que são muitas ações já cadastradas no sistema e existe uma tendencia de existirem ainda mais ações ao longo do tempo.

Para reduzir a quantidade de dados enviados ao *front-end*, o cálculo dos indicadores foi realizado ainda no *back-end*. O objetivo era reduzir a quantidade de dados e, com isso, o tamanho do arquivo enviado. Dessa forma, o arquivo com os indicadores já calculados costuma ter 3,5% do tamanho do arquivo gerado pelo script. Isso, levando em consideração o arquivo com os indicadores com 1,01 MBytes e o arquivo baseado na cópia do sistema SIGAA com 29,17 MBytes.

Além de facilitar o envio do arquivo ao *front-end*, isso também permitiu que o *front-end* fosse apenas uma exibição dos dados já calculados, deixando todo o cálculo no *back-end*.

3.9.5 Planilha baseada nos dados do SIGAA

Ao mesmo tempo em que os indicadores foram calculados para reduzir a quantidade de informações armazenadas, existia um problema com os dados fornecidos ao *front-end*. Esses dados só eram suficientes para calcular os indicadores preestabelecidos. Em outras palavras, mesmo com a capacidade de obter todos os dados do SIGAA, o sistema só seria capaz de mostrar os indicadores que ele fosse capaz de calcular.

Naquele momento, a melhor alternativa seria fornecer todos os dados aos usuários para que esses pudessem realizar uma manipulação ou cálculo de novos indicadores. Para isso,

o ideal não seria fornecer o JSON usado no cálculo dos atuais indicadores, mas, sim, em fornecer uma planilha com os dados desse JSON.

Existe, sim, a opção de importação de dados a partir de um JSON na maioria dos aplicativos de terceiros. Entretanto, por conveniência e vontade de utilizar os softwares institucionalizados, o sistema passou a fornecer um arquivo do tipo Excel, com a extensão XLSX. Esse arquivo é gerado via software e mantém todos os dados do JSON transpostos, visto que, em análises, era mais interessante ver as ações por linhas e o campo de cada ação por colunas.

Vale ressaltar que esse arquivo do tipo Excel não é usado diretamente no *front-end*, apesar de existir uma opção de baixá-lo por lá. Ele é voltado principalmente para conferências dos indicadores ou de análises de dados não calculados pelos indicadores.

3.10 Manutenção do sistema

Pensando em facilitar a manutenção do sistema SARUE, diversas configurações do sistema foram movidas para um conjunto de arquivos em uma pasta para verificar e corrigir problemas no algoritmo. Esse sistema tem muita dependência sobre as páginas do sistema SIGAA. Então, a maioria das estratégias de manutenção são voltadas para correções em relação a mudanças do SIGAA. Além disso, o algoritmo do sistema conta com documentação sobre a implementação e rodas desenvolvidas.

3.10.1 Documentação do projeto

Para a documentação do projeto, foi essencial estabelecer uma estrutura clara e abrangente que abordasse todos os aspectos relevantes do desenvolvimento. Isso inclui a descrição detalhada das especificações técnicas, fluxo do algoritmo e decisões de empregadas no script.

3.10.2 Arquivos de configuração

Como dito anteriormente, os dados contidos nas páginas de extensão são obtidos através da abertura de cada página e da leitura dos dados nela contidos. Os dados obtidos baseiam-se nas posições em que estão na página e suas quantidades dependem do tipo de ação que está sendo pesquisada.

Páginas e endereços do SIGAA

Para solucionar possíveis mudanças na página, todos os XPATHS usados pelo algoritmo foram movidos para arquivos dentro de uma pasta de configurações. Essa mudança permite

alterar os respectivos endereços de um dado sem a necessidade de modificar o algoritmo. Isso garante que nenhuma lógica ou rota estabelecida pelo algoritmo possa ser quebrada.

Através desses arquivos de configurações, também é possível solucionar atualizações quanto às pesquisas. No caso da introdução de uma nova área do CNPq ao sistema SIGAA, o script precisa ser atualizado para pesquisá-la. Esses arquivos permitem que essa nova área seja contemplada apenas inserindo um novo elemento de texto com o seu nome na lista de possíveis áreas do CNPq.

As páginas e URLs usadas pelo sistema SIGAA também são salvas nesses arquivos de configuração. O objetivo disso é permitir que os responsáveis pela manutenção possam localizar com facilidade o elemento que precisa ser alterado. Além disso, essa estratégia permite que esse sistema possa ser testado em diferentes sistemas SIGAAs. Se for de interesse de um desenvolvedor, ele pode alterar as páginas e endereços XPATHS para coincidir com suas páginas do SIGAA e possivelmente utilizar esse algoritmo para obter os dados do SIGAA de outra universidade.

Configurações do script

Além de possibilitar alterações rápidas no que se refere a configurações voltadas ao SIGAA, também foram introduzidas configurações voltadas ao script. Elas têm como principais funções controlar o comportamento do script. Nessas configurações, é possível selecionar o número máximo de instâncias simultâneas e se a execução será visível ou não ao usuário e, ainda, qual o tipo de pesquisa a ser realizada. Existem também informações quanto ao ano de início e fim que devem ser pesquisados, além de uma lista com meses em que deve ser realizada a divisão pela área do CNPq.

As bibliotecas utilizadas também fazem parte de um arquivo nessa pasta. Essa estratégia permite ao sistema verificar, instalar ou atualizar cada biblioteca antes do início do algoritmo, bem como que ele seja executado tendo dependência apenas do Python3[10] e do gestor de pacotes `Pip`[11].

Quanto aos nomes de arquivos de entrada ou saída do script, eles também podem ser alterados por meio dessa pasta. Nela é possível editar o nome do arquivo usado na base de dados anterior, bem como o caminho até ela. Também é possível alterar o nome ou caminho de cada indicador calculado.

Existem também macros definidas ao sistema que podem ser alteradas nessa pasta. Permitindo que o tempo de espera entre módulos do script sejam configuradas de acordo com a escolha do usuário. É possível também alterar as mensagens exibidas pelo terminal durante a execução.

3.11 Testes implementados

Dado que o sistema SARUE necessita conhecer com detalhes o sistema SIGAA, os testes implementados devem verificar dois tipos de requerimentos diferentes.

O primeiro tipo verifica o algoritmo do script, as principais funcionalidades necessárias ao sistema SARUE. Nele, podemos destacar os seguintes testes exibidos pela Tabela 3.1.

Tabela 3.1: Testes unitários do sistema SARUE.

Nome do teste no algoritmo	Objetivo do teste
`testLibraries`	Verificação de presença e importação das bibliotecas necessárias.
`testLoginMultipleInstances`	Verificação se a máquina é capaz de iniciar N instâncias do navegador.
`testGetCredentialsFromEnv`	Verificação da existência de uma credencial armazenada no sistema SARUE.

O teste unitário de verificar a presença e importação das bibliotecas necessárias visa principalmente verificar se as dependências do SARUE estão instaladas e atualizadas em relação às versões mais recentes disponíveis.

O teste de desempenho de verificar se a máquina é capaz de iniciar N instâncias é voltado a máquinas com recursos computacionais de memória limitados. Nessas máquinas, pode ser interessante reduzir de forma manual a quantidade máxima de instâncias utilizadas para obter os dados.

Quanto a verificação de uma credencial armazenada no sistema SARUE, esse teste de integração serve principalmente para evitar que o usuário tenha que pesquisar a credencial de forma manual. Como o sistema não executa o Crawler de autenticação caso exista uma credencial salva, através desse teste, o usuário pode verificar se existe uma credencial armazenada.

O segundo tipo de teste é voltado principalmente para verificar se o SARUE é capaz de se comunicar com a página SIGAA. Os testes implementados devem verificar informações conforme a Tabela 3.2.

A verificação da validade das páginas do sistema SIGAA procura alterações como mudanças de rotas ou endereços no sistema SIGAA. Como o Crawler navega pelas páginas no SIGAA, ele precisa de conhecimento sobre os endereços que devem ser pesquisados. Esse teste verifica se as URLs salvas no SARUE condizem com o que é esperado no sistema SIGAA.

Na verificação da existência de elementos para a autenticação, esse teste visa principalmente verificar mudanças no código fonte da página de autenticação. Para realizar a autenticação, é necessário inserir as credenciais nos campos de login e senha. Para isso, o

Tabela 3.2: Testes unitários do sistema SARUE analisando o SIGAA.

Nome do teste no algoritmo	Objetivo do teste
`testMainPage` `testLoginPage`	Verificação se as páginas do sistema SIGAA salvas são válidas
`testUsername` `testPassword`	Existência de elementos para a autenticação
`testLoginInstance`	Verificação se a credencial salva é válida para autenticação no sistema SIGAA
`testGetOffset`	Possibilidade de obter o offset pelo Crawler de configuração
`testGetActions`	Verificação da quantidade de ações obtidas em um mês específico.

teste verifica se *username* e *password* são campos de entrada na página de autenticação do sistema SIGAA.

O teste de integração de verificação se a credencial salva é válida para autenticação no sistema SIGAA atua em conjunto com o teste de verificar se existe uma credencial armazenada no sistema SARUE. Por meio desses dois testes, é possível verificar se a credencial armazenada existe e se ela é válida para a autenticação. Esse teste é realizado por meio da tentativa de autenticação do sistema.

O teste de possibilidade de obter o *offset* pelo Crawler de configuração, tem como intuito verificar se através de um mês vazio é possível obter um valor numérico e condizente com um valor esperado para o *offset*.

A verificação da quantidade de ações obtidas em um mês específico é dada por um mês em que a quantidade de ações seja conhecida. O teste consiste em realizar a filtragem por mês, e obter os dados de um mês inteiro para então verificar se a quantidade de dados obtidos é condizente com a quantidade de dados cadastrados para aquele mês. Esse teste demora um tempo considerável em relação aos demais, mas, por meio dele, é possível verificar de forma geral o funcionamento do sistema.

3.12 Limitações do sistema

O sistema SARUE apresenta algumas limitações para seu correto funcionamento. Essas limitações podem envolver questões ligadas à disponibilidade e à autenticação do sistema SIGAA, o qual é utilizado com base de dados, e também limitações referentes às bibliotecas ou aos navegadores utilizados. A seguir serão descritas as principais limitações do sistema.

3.12.1 Disponibilidade do sistema SIGAA

Existem algumas limitações ligadas exclusivamente ao sistema SIGAA. A primeira delas é a disponibilidade do sistema SIGAA. Como esse sistema é hospedado dentro da infraestrutura da Universidade, ele pode sofrer com instabilidades ou até mesmo desligamentos. Em períodos de matrícula, o sistema pode sofrer com indisponibilidade total por congestionamento. Nesses períodos, o sistema recebe tantas requisições simultâneas que apresenta demoras ou até mesmo cancelamento de respostas de requisições. Nessas épocas, o uso do sistema SARUE pode ser comprometido em decorrência da indisponibilidade de utilizar a autenticação pelo sistema SIGGA, bem como não poder realizar novas obtenções ou atualizações dos dados.

Vale ressaltar que a indisponibilidade do sistema pode ser parcial. Muitas vezes algum módulo do sistema está sofrendo com uma instabilidade enquanto outros módulos funcionam corretamente.

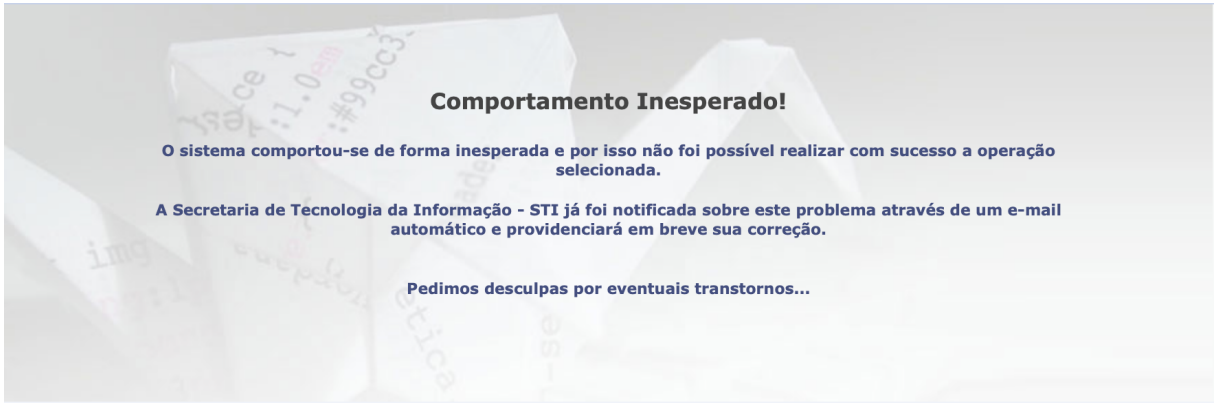


Figura 3.12: Mensagem de comportamento inesperado do sistema SIGAA.

Quando a listagem de ações dentro da plataforma está indisponível, o site retorna a mensagem de erro da Figura 3.12.

O sistema, ao detectar que existe uma indisponibilidade na plataforma de obtenção dos dados, retorna uma mensagem de erro ao usuário informando isso.

3.12.2 Necessidade de uma credencial válida

Como o sistema realiza uma pesquisa dos dados contidos no sistema SIGAA, ele necessita acessar a página de listar ações de extensão. E para tanto, requer uma credencial válida do sistema SIGAA.

3.12.3 Dependência de bibliotecas

O fato do sistema SARUE depender de bibliotecas para seu funcionamento, implica que as dependências dessas bibliotecas são também limitações do sistema. Modificações e atualizações dessas ferramentas, assim como sua correta instalação influenciam diretamente o funcionamento do sistema. Apesar de testes extensivos quanto ao funcionamento em diferentes máquinas, o script do SARUE está sujeito ao não funcionamento dependendo de bibliotecas externas.

3.13 Indicadores

Os indicadores são um conjunto de dados que visam possibilitar a avaliação do desempenho operacional das instituições, de acordo com [22]. Além disso, por meio do uso de indicadores é possível analisar de forma precisa as informações, facilitando a extração de *insights* relevantes a partir dos dados disponíveis.

3.13.1 Obtenção dos indicadores

No decorrer das entrevistas de elicitação de requisitos, foi disponibilizado ao grupo do projeto uma lista de indicadores que o DEX gostaria que fossem calculados. Apesar da lista de indicadores fornecida especificar como realizar o cálculo dos indicadores, foram necessárias adaptações para calcular alguns indicadores. Alguns indicadores solicitados não podem ser calculados com base nos dados obtidos apenas pela plataforma SIGAA.

Os indicadores solicitados são divididos principalmente com base na sua dimensão de avaliação. Existem indicadores voltados a Política de Gestão, Infraestrutura, Plano Acadêmico, Relação Universidade-Sociedade e Produção Acadêmica. Dos 46 indicadores solicitados, 8 deles foram fornecidos por meio de um acórdão do TCU[22]. Esse acórdão apresenta indicadores que devem ser enviados em uma determinada época, visto que são utilizados na prestação de contas ao referido Tribunal.

Esses indicadores de acordo com [22] precisam atender aos seguintes requisitos: possibilitar apuração operacional, possuir atributos de comparabilidade e ter capacidade de representar de forma confiável aspectos da realidade acadêmica.

De maneira prioritária, foram considerados na implementação os indicadores voltados ao TCU. Entretanto, alguns deles, apesar de serem indicadores de extensão, dependem de dados que não estão disponibilizados no módulo de extensão da plataforma SIGAA da UnB. Outros indicadores utilizam dados do SIGAA em parte do seu cálculo.

Foram obtidos todos os indicadores que poderiam ser calculados unicamente com os dados contidos na plataforma SIGAA. Como dentro da plataforma não é possível obter

a quantidade total de discentes, docentes e técnicos, o sistema SARUE fornece os indicadores de percentual ou índices de forma parcial, disponibilizando apenas a quantidade presente nos dados de extensão. O usuário final, com conhecimento da quantidade total dos discentes, docentes e externos, pode então realizar o cálculo percentual.

3.13.2 Indicadores calculados

No desenvolvimento do *back-end* foram calculados os indicadores contidos na Tabela 3.3. Os nomes referentes aos indicadores não condizem com os indicadores propriamente ditos. Isso se deve a necessidade de armazenar nomes pequenos e de fácil entendimento nos arquivos enviados ao *front-end*.

Tabela 3.3: Descrição dos indicadores calculados no *back-end*.

Descrição do indicador	**Nome arquivo JSON**

Quantidades de ações	`quantidade`
	`quantidade_anual`
	`quantidade_anual_tipo`
	`quantidade_mensal`
	`quantidade_mensal_tipo`

Status das ações	`status_acoes`
	`status_acoes_anual`
	`status_acoes_anual_tipo`
	`status_acoes_mensal`
	`status_acoes_mensal_tipo`

Objetivos contemplados	`objetivos_contemplados`
	`objetivos_contemplados_anual`
	`objetivos_contemplados_anual_tipo`
	`objetivos_contemplados_mensal`
	`objetivos_contemplados_mensal_tipo`

Envolvidos como membros	`quantidade_envolvidos`
	`quantidade_envolvidos_anual`
	`quantidade_envolvidos_anual_tipo`
	`quantidade_envolvidos_mensal`
	`quantidade_envolvidos_mensal_tipo`

Informações completas	`info`
	`info_anual`
	`info_anual_tipo`
	`info_mensal`
	`info_mensal_tipo`

Alguns indicadores solicitados não puderam ser calculados de forma completa, por isso, eles são tratados a partir dos dados que puderam ser obtidos do sistema SIGAA.

Em relação ao *front-end* do sistema SARUE, os indicadores lá exibidos podem usar mais de uma fonte de dados fornecida pelos arquivos do *back-end*. Exemplo disso, são as tabelas que exibem os valores por mês e por ano. Na Tabela 3.4 são mostrados os indicadores exibidos no *front-end*, bem como a origem dos dados, a dimensão de avaliação deles e o requerente desses indicadores.

Tabela 3.4: Descrição dos indicadores exibidos no *front-end*.

Nome do indicador no *front-end*	**Origem dos dados**	**Dimensões de avaliação**	**Requerente**
Ações institucionalizadas por ano	`quantidade_anual` `quantidade_mensal`	Política de Gestão	TCU
Ações cadastradas por categoria	`quantidade_anual_tipo` `quantidade_mensal_tipo`	Plano Acadêmico	TCU
Ações por ODS	`objetivos_contemplados_anual` `objetivos_contemplados_mensal`	Relação Universidade-Sociedade	TCU
Pessoas envolvidas por ano	`quantidade_envolvidos_anual`	Plano Acadêmico	TCU
Público real atingido por ano	`info_anual` `info_mensal`	Relação Universidade-Sociedade	TCU
Estudantes extensionistas por projeto	`info_anual`	Plano Acadêmico	Indicadores Acadêmicos
Ações de extensão por unidade	`info`	Plano Acadêmico	Indicadores Acadêmicos
Ações com financiamento externo x autofinanciadas	`info_anual`	Política de Gestão	Indicadores Acadêmicos
Ações institucionalizadas no SIGAA em relação ao ano anterior	`status_acoes_anual`	Relação Universidade-Sociedade	Indicadores Acadêmicos

A quantidade de arquivos que é fornecida ao *front-end* é maior que o necessário para a exibição gráfica dos indicadores. Com isso, em caso da necessidade de exibir um novo indicador, e tendo a garantia que os dados contidos no sistema SIGAA serão suficientes para esse novo indicador, existe a chance de poder adiciona-lo sem a necessidade de alterar o *back-end*.

3.13.3 Cálculo dos indicadores

Os indicadores são calculados com base no arquivo de saída resultante da execução do script. A execução do algoritmo responsável pelo cálculo dos indicadores pode ser realizada através de um comando do usuário ou como acontece na maioria das vezes, é realizada após a execução do script.

O algoritmo foi implementado utilizando dicionários do Python3. Inicialmente o arquivo JSON gerado pelo script é carregado a memória pelo algoritmo. A estratégia de primeiro salvar o arquivo resultante da execução do script de obtenção de dados e então reabri-lo foi empregada para garantir que o algoritmo de obtenção de dados ou de cálculo de indicadores pudessem ser executados de forma independentes.

Após carregar em um dicionário os dados obtidos, o algoritmo realiza diversas funções para calcular diversos indicadores. Cada função, de forma reduzida, consiste em iterações sobre esses elementos e em contar a quantidade através de um novo dicionário no caso de elementos repetidos, ou então por meio de listas para elementos com repetições. Por exemplo, para verificar os membros de equipe, cada membro de equipe é adicionado a uma lista sem repetições, então, com base no escopo de pesquisa, é informada a quantidade de membros e seus papéis no indicador final.

Cada função dos indicadores retorna de cinco novos dicionários com os valores calculados. Esses cinco são utilizados na divisão dos indicadores calculados, pois, permitem fornecer os dados dos indicadores sem repetições.

Após isso, cada conjunto de indicadores é salvo em um arquivo que contém apenas o valor desse indicador e os conjuntos de quase todos os indicadores calculados é salvo em um outro arquivo que é destinado ao *front-end*.

Nem todos os indicadores calculados são enviados ao *front-end*. Isso se deve a não necessidade deles ou até mesmo requisições que não condizem com o projeto atual. No cálculo dos indicadores no *back-end*, é criada uma tabela com o nome dos membros de cada equipe e seus papéis naquela equipe. Esses dados são usados para remover repetições. Entretanto, não são indicadores propriamente ditos. Nesse caso, esses dados são mantidos em algoritmo, mas não são enviados ao *front-end* do SARUE.

3.13.4 Divisão dos indicadores calculados

A dimensão do cálculo é realizada com base na abrangência de pesquisa que pode ser selecionada. Isto é, existem pequenas diferenças e tratamentos de repetições ao selecionar diferentes escopos de pesquisa. Exemplo disso é a quantidade de discentes que participaram de ações de extensão. Se o cálculo for realizado mês a mês, a somatória dos meses

de um ano pode não compreender os dados referentes a ele, visto que um mesmo discente pode participar de mais de uma ação de extensão ao longo desse mesmo ano.

Nesses casos, são realizados cálculos anuais e mensais dos mesmos conjuntos de dados. Essa estratégia, apesar de onerosa em recursos computacionais, garante que determinados dados sejam tratados quanto à repetição.

Os indicadores calculados no *back-end* apresentam 5 diferentes âmbitos de seleção, sendo eles:

- Sem divisão;
- Por ano;
- Por ano e por tipo de ação;
- Por mês; e
- Por mês e por tipo de ação.

Tratando dos dados sem divisões

A proposta de exibir todos os dados obtidos pelo sistema SARUE veio da necessidade de remover todas as repetições possíveis contidas no sistema. Isso possibilitou exibir dados como a quantidade de discentes e docentes que participaram como membros em ações de extensão em todos os projetos cadastrados no sistema SIGAA.

Além disso, esses valores ajudam o *front-end* a exibir dados rápidos sobre a extensão na UnB, possibilitando mostrar valores como a quantidade de ações realizadas e o tipo de cada uma delas. Ainda é possível, de forma inicial, exibir um histórico sobre a extensão universitária na UnB.

Listagem 3.6: Quantidade de envolvidos como membros da equipe no período de 2020 a 2022

```
{
   "Discente": 11551,
   "Docente": 1851,
   "Externo": 7947,
   "Servidor": 598
}
```

O código 3.6 exibe a quantidade de envolvidos como membros da equipe organizadora por tipo de ação cadastradas no sistema de 2020 a 2022.

Tratando dos dados por ano

Como já dito anteriormente, a existência de repetições nos dados foi a principal justificativa para realizar cálculos anuais ao invés de simplesmente somar os dados mensais de determinado ano.

Existem diversos indicadores que são de abrangência anual. Nesse caso, o tratamento com base no ano fornece de forma mais precisa esses indicadores.

Listagem 3.7: Quantidade de envolvidos por ano como membros da equipe no período de 2020 a 2022

```
{
   "2020": {
      "Discente": 1198,
      "Docente": 468,
      "Externo": 840,
      "Servidor": 65
   },
   "2021": {
      "Discente": 5847,
      "Docente": 1304,
      "Externo": 4257,
      "Servidor": 387
   },
   "2022": {
      "Discente": 8103,
      "Docente": 1490,
      "Externo": 4080,
      "Servidor": 442
   }
}
```

O código 3.7 exibe por ano a quantidade de envolvidos como membros da equipe organizadora por tipo de ação cadastradas no sistema de 2020 a 2022.

De acordo com os dados exibidos no código 3.6 e no código 3.7, somando a quantidade de discentes envolvidos nos anos de 2020 a 2022 de acordo com a divisão por ano, existiriam 15.148 discentes envolvidos no período de 2020 a 2022. Entretanto como exibido nos dados sem divisões, o total de discentes envolvidos é de 11.551 como membros da equipe. Isso se deve a possibilidade de um mesmo discente poder participar de mais de uma ação de extensão.

Para remover essas repetições, os nomes dos discentes são adicionados a uma estrutura de dados que remove as repetições, por isso o uso de dicionários do Python3.

Tratando dos dados por ano e pelo tipo de ação

Existem também indicadores que são anuais e dependem do tipo de ação realizada. Essa abordagem é capaz de fornecer dados como a quantidade anual de uma determinada ação e possivelmente permitir comparações entre diferentes ações.

Listagem 3.8: Quantidade de envolvidos por ano e por tipo de ação como membros da equipe no período de 2020 a 2022

```
{
   "2020": {},
   "2021": {},
   "2022": {
      "CURSO": {
         "Discente": 646,
         "Docente": 324,
         "Externo": 656,
         "Servidor": 72
      },
      "EVENTO": {
         "Discente": 4217,
         "Docente": 1200,
         "Externo": 2357,
         "Servidor": 332
      },
      "PRODUTO": {},
      "PROGRAMA": {},
      "PROJETO": {}
   }
}
```

O código 3.8 exibe por ano e pelo tipo de ação de extensão a quantidade de envolvidos como membros da equipe organizadora por tipo de ação cadastradas no sistema de 2022. Para melhorar a visualização, os dados de 2020 e 2021 e os dados referentes aos produtos, programas e projetos de 2022 foram removidos.

Tratando dos dados por mês

O tratamento por mês ajuda a exibir valores em progressões de tempo ou até mesmo analisar meses de forma individual. Por meio desses indicadores, também é possível preencher gráficos anuais com os valores mensais. Nesse caso, é importante ressaltar que a soma dos valores mensais pode não corresponder ao valor anual.

Listagem 3.9: Quantidade de envolvidos por ano e por mês como membros da equipe no período de 2020 a 2022

```
{
   "2020": {},
   "2021": {},
   "2022": {
      [...]
      "novembro": {
         "Discente": 405,
         "Docente": 174,
         "Externo": 262,
         "Servidor": 65
      },
      "dezembro": {
         "Discente": 265,
         "Docente": 106,
         "Externo": 134,
         "Servidor": 23
      }
   }
}
```

O código 3.9 exibe por ano e por mês a quantidade de envolvidos como membros da equipe organizadora por tipo de ação cadastradas no sistema de 2022. Para melhorar a visualização, os dados de 2020 e 2021 e os dados referentes aos meses de janeiro a outubro de 2022 foram removidos.

Tratando dos dados por mês e pelo tipo de ação

Essa abordagem fornece os valores por meses e pelo tipo de ação que foi cadastrada. Com isso, é possível analisar os dados de um tipo de ação em função do tempo. Além disso, existem indicadores voltados a um tipo de ação específica de determinados meses. Por meio dessa abordagem, o usuário pode ver a progressão de um tipo de ação em meses de diferentes anos, como por exemplo, verificar a quantidade de eventos realizados no mês da Semana Universitária da UnB.

Essa é a divisão em função do tempo mais detalhada aplicada ao sistema. Em caso de necessidade por partes dos stakeholders, é possível adicionar novas subdivisões aos dados.

Listagem 3.10: Quantidade de envolvidos por ano por mês e por tipo de ação como membros da equipe no período de 2020 a 2022

```
{
   "2020": {},
```

```
3     "2021": {},
4     "2022": {
5        [...]
6        "dezembro": {
7           "CURSO": {
8              "Discente": 48,
9              "Docente": 18,
10             "Externo": 2,
11             "Servidor": 9
12          },
13          "EVENTO": {
14             "Discente": 209,
15             "Docente": 81,
16             "Externo": 119,
17             "Servidor": 14
18          },
19          "PRODUTO": {},
20          "PROJETO": {}
21       }
22    }
23 }
```

O código 3.10 exibe por ano, por mês e pelo tipo de ação de extensão a quantidade de envolvidos como membros da equipe organizadora por tipo de ação cadastradas no sistema de 2022. Para melhorar a visualização, os dados de 2020 e 2021, os dados referentes aos meses de janeiro a novembro de 2022 e os referentes aos produtos e projetos de dezembro de 2022 foram removidos.

3.13.5 Ordenação dos indicadores calculados

Antes de salvar os indicadores calculado em arquivos do tipo JSON, são aplicadas funções de ordenação nesses indicadores. Essas funções ajudam a exibir os dados com maior legibilidade e organização. Além disso, essa padronização ajuda a realizar diferentes comparações entre os arquivos calculados.

Com o uso desse método, é possível encontrar os valores de um indicador simplesmente navegando pelo arquivo e vendo o seu valor. Apesar de ser um arquivo parecido com arquivos de texto, ele permite que um valor seja encontrado com facilidade abrindo e fechando chaves do arquivo.

3.14 Considerações finais sobre o desenvolvimento

Nesse capítulo foram exibidas divisões realizadas nos Crawlers, as técnicas utilizadas para obter os dados contidos no sistema SIGAA, bem como os métodos usados para acelerar a obtenção dos dados, garantir os dados e calcular os indicadores propostos.

A versão completa do algoritmo desenvolvido pode ser encontrada em [23].

No próximo capítulo, no Capítulo 4, são exibidos os resultados com base no tempo de execução com a execução sequencial, com uso de MiniCrawlers paralelos e com o uso de MiniCrawlers concorrentes.

Capítulo 4

Resultados

Esse capítulo exibe os resultados obtidos pelos testes, e apresentar variáveis que podem influenciar na execução do script.

Todos os testes comentados a seguir foram executados através do terminal, tendo as configurações ajustadas, selecionadas e empregadas para satisfazer as condições desse teste.

Nos anexos são fornecidas maiores informações sobre os testes realizados, como a descrição dos testes por execução, a quantidade de instâncias utilizadas, o método de divisão dessas instâncias, ano de início e fim pesquisados, quantidade de ações obtidas, horário de início e fim dessa execução e o tempo de execução.

Neste capítulo também é apresentado um teste de benchmark. Esse benchmark tem como base a metodologia apresentada em [6], e através dele é possível estimar o poder computacional de cada máquina utilizada.

4.1 Metodologia dos testes

A metodologia de testes utilizada consiste em executar o algoritmo de forma repetida e utilizar os menores valores obtidos em cada rodada de testes. A execução com o uso de MiniCrawlers paralelos e a execução com o uso de MiniCrawlers concorrentes foram realizadas em sequência buscando obter condições de testes mais próximas. A execução de forma linear, utiliza os melhores valores obtidos na execução paralela ou concorrente com o uso de uma instância.

Para estimar o tempo de execução do script, foram implementadas funções com base na biblioteca `time.h` do Python3. No projeto foi introduzido uma classe na qual é responsável por ao iniciar o script, salvar o horário de início da execução e ao finalizar a execução desse script, armazenar o horário de fim da execução, o tempo decorrido e salvar em um arquivo essas informações.

Listagem 4.1: Classe Timer - Usada para estimar tempo de execução

```
1  Classe Timer:
2      Função __init__():
3          Inicializar self.start_time com o valor atual de tempo
4          Inicializar self.start_ctime com o valor atual de tempo formatado como string
5
6      Função set_end_time():
7          Atribuir a self.end_time o valor atual de tempo
8          Atribuir a self.end_ctime o valor atual de tempo formatado como string
9
10     Função get_elapsed_time():
11         Chamar a função set_end_time()
12         Calcular elapsed_time como (self.end_time - self.start_time)
13         Calcular minutes como a parte inteira de (elapsed_time/60)
14         Calcular seconds como a parte inteira do resto da divisão (elapsed_time, 60)
15
16         Retornar minutes, seconds
17
18     Função print_elapsed_ctime():
19         Desempacotar minutes e seconds chamando a função get_elapsed_time()
20
21         Chamar a função clear_screen()
22         Chamar a função centralize() com o argumento f'{RIGHT_ARROW} Script end
               {LEFT_ARROW}'
23
24         Imprimir o separador
25         Chamar a função centralize() com o argumento f'Quantidade de acoes =
               {get_quantity_of_activities()}'
26         Imprimir o separador
27
28         Imprimir o separador
29         Imprimir f"\tStart time: {self.start_ctime}"
30         Imprimir f"\tEnd time: {self.end_ctime}"
31         Imprimir f"\tElapsed time: {minutes:02}:{seconds:02} minutes"
32         Imprimir o separador
```

O código 4.1 é referente a essa classe de funções. Nele é possível identificar a função responsável por salvar o tempo de início [linhas 2 a 4], bem como a função responsável por obter o tempo de fim [linha 6 a 8] e o tempo decorrido [linhas 10 a 16].

Na Figura 4.1 é possível identificar as variáveis que são geradas pelo algoritmo mostrado no código 4.1 e salvas no arquivo em anexo.

No capítulo anterior foram descritos métodos para acelerar a obtenção dos dados contidos na plataforma SIGAA. As subseções a seguir mostram os valores dos tempos de

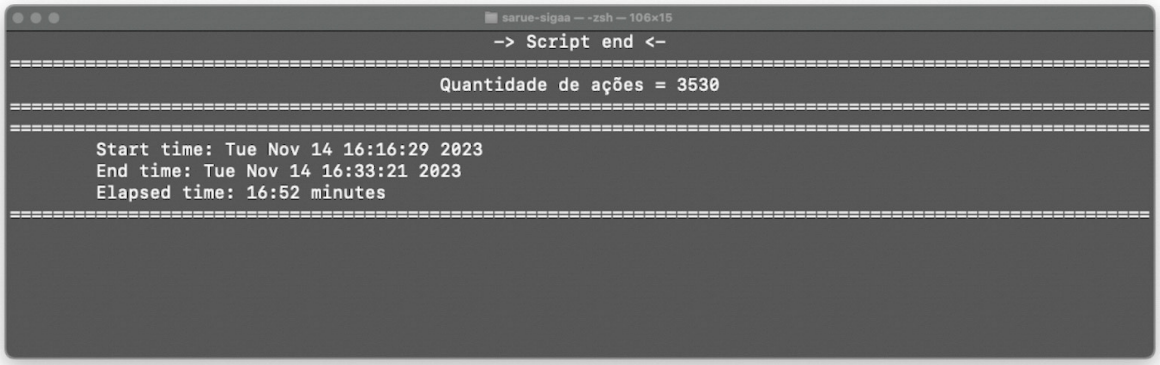


Figura 4.1: Mensagem do terminal exibidas ao usuário ao final da execução.

execução obtidos para cada tipo de execução.

4.2 Descrição dos equipamentos utilizados

Para a realização dos testes, as máquinas descritas nas Tabelas 4.1 a 4.3 foram utilizadas.

Tabela 4.1: Descrição de componentes da Máquina 1.

Máquina 1	
Hardware	Processador: Apple M2 Pro (12-core CPU) Memória RAM: 16GB DDR5 Armazenamento: 1tb SSD
Sistema Operacional	macOS Sonoma Versão 14.0 Arquitetura Arm64

Tabela 4.2: Descrição de componentes da Máquina 2.

Máquina 2	
Hardware	Processador: 10700K (8-core CPU / 16 threads) Memória RAM: 32gb DDR4 Armazenamento: 1tb SSD
Sistema Operacional	Windows 11 Pro Versão 22H2 Arquitetura x64

Uma observação é que entre as máquinas utilizadas a máquina 2 e a máquina 3, descritas pelas Tabelas 4.2 a 4.3 respectivamente, possuem o mesmo hardware, apenas utilizam sistemas operacionais diferentes entre si.

Tabela 4.3: Descrição de componentes da Máquina 3.

Máquina 3	
Hardware	Processador: 10700K (8-core CPU / 16 threads) Memória RAM: 32gb DDR4 Armazenamento: 128gb SSD
Sistema Operacional	Linux Mint Versão 21.2 Cinnamon Edition Arquitetura x64

Todas as máquinas no período de realização dos testes utilizavam as mesmas versões das bibliotecas conforme a Tabela 4.4

Tabela 4.4: Versões das bibliotecas utilizadas em todas as máquinas

Bibliotecas	Python 3.12.0 Pip 23.3.1 Selenium 4.15.2 Python-dotenv 1.0.0 Tqdm 4.66.1 Pandas 2.1.3

4.3 Resultados da execução sequencial

A primeira implementação do projeto foi realizada utilizando uma única instância de um navegador. Os tempos obtidos pelo uso de apenas uma instância é muito próximo ao tempo obtido pelo uso de uma instância com o uso de MiniCrawlers paralelos ou MiniCrawlers concorrentes. Por isso, os valores mostrados a seguir são os menores valores obtidos pela execução paralela ou concorrente nessa máquina com o uso de uma instância.

Tabela 4.5: Execução sequencial.

Máquina 1	119:56
Máquina 2	136:54
Máquina 3	123:40

Com base nos valores contidos na Tabela 4.5 seria possível estimar o tempo de execução com a divisão pelo uso de múltiplas instâncias. Entretanto o resultado com o uso de mais instâncias depende de como a máquina se comporta ao iniciar e sincronizar essas instâncias.

4.4 Resultados com uso de MiniCrawlers paralelos

Essa seção de resultados apresenta os resultados da execução com base no modo paralelo, como descrito na seção MiniCrawler paralelo no Capítulo 3. Inicialmente é importante verificar a quantidade de ações pesquisadas por cada instância de MiniCrawlers. Na Tabela 4.6, é exibida a quantidade de ações por ano no período de 2020 a 2022.

Tabela 4.6: Quantidade de ações por ano.

Ano	Quantidade de ações
2020	461
2021	2028
2022	2336

Com base nessa tabela, é possível identificar que as instâncias responsáveis por pesquisar 2021 e 2022 devem realizar mais pesquisas em relação ao total de 4825 ações no período de 2020 a 2022.

Conforme a Equação 3.2 exibida no Capítulo 3, essa execução deve demorar o mesmo tempo de pesquisar de forma linear, o ano com mais ações cadastradas. Nesse caso, a execução de 2020 a 2022 deve demorar o mesmo tempo que executar de forma linear a procura pelo ano de 2022.

Tabela 4.7: Quantidade de ações por semestre.

Semestre	Quantidade de ações
01/2020	62
02/2020	399
01/2021	701
02/2021	1327
01/2022	1101
02/2022	1235

Utilizando a divisão semestral, conforme Tabela 4.7, é possível identificar 3 instâncias são responsáveis por obter mais dados em relação ao total do período de 2020 a 2022, sendo elas os semestres 02/2021, 01/2022 e 02/2022.

De acordo com a Equação 3.3 exibida no Capítulo 3, é possível supor que o tempo de execução do script com base na divisão semestral deve ser igual ao tempo de execução linear do semestre 02/2021, que possuiu a maior quantidade de ações cadastradas nessa divisão.

Quanto a divisão quadrimestral, a quantidade de ações cadastradas a cada 4 meses é exibida na Tabela 4.8. Com base nessa tabela é possível identificar que 3 instâncias realizam mais trabalho em função da quantidade de ações cadastradas, gerando um tempo de

Tabela 4.8: Quantidade de ações por quadrimestre.

Quadrimestre	Quantidade de ações
01/2020	-
02/2020	208
03/2020	253
01/2021	485
02/2021	421
03/2021	1122
01/2022	1924
02/2022	885
03/2022	527

execução próximo ao tempo de execução linear do quadrimestre com mais ações conforme Equação 3.4 exibida no Capítulo 3.

Tabela 4.9: Quantidade de ações por trimestre.

Trimestre	Quantidade de ações
01/2020	-
02/2020	62
03/2020	253
04/2020	146
01/2021	358
02/2021	343
03/2021	1042
04/2021	285
01/2022	764
02/2022	337
03/2022	1002
04/2022	233

No caso da divisão por trimestres, a quantidade de ações é exibida na Tabela 4.9. Com base nesses valores, é possível identificar que existem 2 períodos de três meses que possuem mais ações cadastradas. Com base na Equação 3.5 exibida no Capítulo 3, é possível assumir que o tempo de execução nesse uso é estimado pelo tempo de execução linear do maior entre esses 2 períodos.

A Tabela 4.10 mostra os resultados obtidos com a execução paralela do algoritmo na máquina 1 descrita pela Tabela 4.1. As colunas dessa tabela mostram a quantidade de instâncias do navegador utilizadas e as linhas dessa tabela mostram o período de divisão utilizado pelas instâncias para obter as ações contidas na página de extensão do sistema SIGAA, sendo o mais rápido no quesito de tempo quando múltiplas instâncias são usadas.

Com base nos tempos exibidos pela Tabela 4.10 é possível verificar que, em geral, uso de maior quantidade de instâncias permite ao algoritmo finalizar sua execução completa

Tabela 4.10: Execução paralela - Máquina 1.

	Instâncias				
Período de divisão	1	3	6	9	12
Anual	124:06	60:47	-	-	-
Semestral	119:56	49:22	49:09	-	-
Quadrimestral	120:41	44:28	30:39	30:23	-
Trimestral	124:27	51:53	32:16	26:26	26:23

mais rápido. Ao mesmo tempo, é possível identificar que em determinados períodos de divisão de tempo, o uso de mais instâncias não trouxe um avanço significativo no tempo de execução. Como por exemplo, na divisão semestral entre 3 e 6 instâncias, na divisão quadrimestral entre 6 e 9 instâncias e também na divisão trimestral com 9 e 12 instâncias.

Esses resultados são principalmente decorrentes da divisão escolhida, mostrada no Capítulo 3, onde o tempo de execução depende das quantidades de ações na subdivisão de tempo. Na divisão semestral, por exemplo, mesmo com o uso de 6 instâncias, 3 delas tinham mais ações a serem pesquisadas, fazendo com que seu tempo de execução fosse maior que o tempo das demais instâncias. Isso pode ser observado pela quantidade de ações semestrais contidas no SIGAA e exibidas na Tabela 4.7.

Tabela 4.11: Execução paralela - Máquina 2.

	Instâncias				
Período de divisão	1	3	6	9	12
Anual	147:39	77:04	-	-	-
Semestral	144:38	60:12	44:47	-	-
Quadrimestral	148:32	57:37	40:28	41:11	-
Trimestral	136:54	65:46	41:38	37:02	36:26

Para os resultados obtidos pela execução na máquina 2 são mostrados na Tabela 4.11. Com base nesses resultados, é possível identificar que alguns tempos de execução que estavam próximos na máquina 1 se mantém em relação a máquina 2, entretanto tempos como na execução com divisão semestral em 3 e 6 instâncias, o uso de mais instâncias possibilitou finalizar a execução mais rápido.

Tabela 4.12: Execução paralela - Máquina 3.

	Instâncias				
Período de divisão	1	3	6	9	12
Anual	131:27	67:30	-	-	-
Semestral	134:13	60:31	42:12	-	-
Quadrimestral	140:43	55:36	43:18	41:25	-
Trimestral	165:03	65:49	44:58	39:07	39:30

Os resultados da máquina 3 são exibidos pela Tabela 4.12. Com base nesses valores é possível identificar que o uso de mais instâncias em geral significa um menor tempo de obtenção dos dados. Apesar disso, na divisão quadrimenstral, o uso de 9 instâncias não proporcionou um tempo melhor que o uso de 6 instâncias. É possível que devido a um possível congestionamento na rede, ou ainda porque uma ou duas instâncias estarem sobrecarregadas de ações a serem pesquisadas.

Em geral, é possível apontar que com o uso de mais instância é possível reduzir o tempo de execução com essas subdivisões temporais.

4.5 Resultados com uso de MiniCrawlers concorrentes

No Capítulo 3 foi apresentado o funcionamento do Crawler concorrente, como explicado na seção MiniCrawler concorrente do Capítulo 3, esse método realiza todas as divisões de tempo de mês a mês, podendo ocorrer também a divisão por meio da área do CNPq de uma ação.

Além de limitar a quantidade de configurações de tempo possíveis, esse método realiza a autenticação das instâncias antes de realizar a filtragem e pesquisa das ações, impedindo que uma instância já autenticada realize filtragens e pesquisas até que todas estejam autenticadas.

Tabela 4.13: Execução concorrente - Máquina 1.

Instâncias	1	3	6	9	12
Tempo de execução	122:41	43:50	26:37	21:22	20:12

A Tabela 4.13 mostra os tempos de execução obtidos pela versão concorrente na máquina 1.

Com base nos tempos exibidos na Tabela 4.13 é possível verificar que o uso de mais instâncias resulta em uma redução do tempo de execução. Além disso, quando comparada em relação aos tempos da execução paralela, os tempos obtidos pela execução concorrente tendem a serem melhores ou relativamente próximos dos melhores tempos obtidos pela execução paralela levando em conta a mesma quantidade de instâncias.

Tabela 4.14: Execução concorrente - Máquina 2.

Instâncias	1	3	6	9	12
Tempo de execução	151:33	57:50	45:43	31:16	29:26

Os dados obtidos pela execução dos testes no modo de execução concorrente na máquina 2 podem ser visualizados na Tabela 4.14

Com base nos tempos apresentados na Tabela 4.14, é possível identificar um aumento no tempo de execução quando comparado aos tempos de execução paralela mostrados na Tabela 4.11 com o uso de 1 ou 6 instâncias. Isso pode ser explicado pela maior complexidade empregada nesse tipo de execução ou pelas limitações de conexão exibida na seção MiniCrawler concorrente do Capítulo 3.

Quando comparada a uma maior quantidade de instâncias, os resultados mostram que essa solução é capaz de obter os dados em menor tempo quando comparada a execução paralela. Por exemplo, como o uso de 9 instâncias do navegador, o melhor tempo obtido pela execução paralela na máquina 2 é de 39 minutos e 07 segundos, enquanto com a execução concorrente, é possível obter os mesmos dados em 31 minutos e 16 segundos.

Tabela 4.15: Execução concorrente - Máquina 3.

Instâncias	1	3	6	9	12
Tempo de execução	123:40	45:12	34:40	21:56	20:47

A Tabela 4.15 exibe os valores obtidos pela execução dos scripts na máquina 3 de forma concorrente. Com base nos valores apresentados, é notável a redução do tempo para a obtenção dos dados com o uso de mais instâncias de navegadores.

Além disso, é possível verificar o impacto de diferentes sistemas operacionais sobre o mesmo hardware. Como dito anteriormente, as máquinas 2 e 3 possuem o mesmo hardware, mas com diferenças no sistema operacional. Quando comparados os tempos exibidos pela Tabela 4.14 e pela Tabela 4.15, é possível identificar uma redução nos tempos de obtenção dos dados na máquina 3, que roda o sistema operacional Linux.

Além dos fatores listados nas limitações desse teste, é possível identificar fatores como a distribuição de recursos computacionais, como a alocação de memória e uso do processador a um processo geridos pelo sistema operacional, que podem influenciar significativamente no desempenho da execução paralela.

4.6 Limitações dos testes

Os resultados apresentados são influenciados por diversos elementos. Variações relacionadas à máquina de teste, à conexão com a internet no momento do teste, entre outros, desempenham um papel significativo. A seguir, são destacados os principais elementos que podem impactar os resultados obtidos.

4.6.1 Elementos relacionados à máquina de teste

Diferentes aspectos da máquina utilizada para os testes podem afetar o tempo de execução. Variações nas arquiteturas, quantidade e velocidade da memória, bem como nos sistemas

operacionais, podem influenciar os resultados. Além disso, softwares antivírus também podem ter efeitos nos resultados.

Quantidade e velocidade da memória

A utilização de memórias com velocidades diversas é um elemento que pode influenciar o tempo de execução do algoritmo. Tanto a velocidade quanto a quantidade de memória disponível podem impactar o funcionamento e a sincronização das instâncias do navegador. Diversos navegadores demandam considerável uso de memória para operar adequadamente. O navegador Chrome, por exemplo, mantém páginas carregadas em cache, resultando em melhor desempenho quando há memória suficiente para atender todas as instâncias ativas. Vale destacar que, nos testes realizados, a mesma versão do navegador foi utilizada em todas as máquinas, havendo diferenças relacionadas à versão específica do Chrome [18] adaptada à arquitetura do processador de cada máquina.

Disponibilidade de recursos computacionais

Além das considerações específicas sobre a memória, a disponibilidade geral de recursos computacionais é um fator crucial. Isso engloba a capacidade de processamento da CPU, a largura de banda da rede, o escalonamento de processos pelo sistema operacional, entre outros.

Um sistema com recursos computacionais mais robustos pode otimizar o tempo de execução do script. Como exemplificado pelo benchmark apresentado neste capítulo, a máquina 1 demonstra maior poder computacional em comparação com as máquinas 2 e 3. Os tempos de execução do script na máquina 1 também são inferiores aos das máquinas 2 e 3, conforme indicado pelas Tabelas 4.10 a 4.12.

Uso de antivírus no navegador

Alguns antivírus adotam a prática de processamento remoto, utilizando servidores para verificar a confiabilidade de um site antes mesmo de recepcionar a resposta do site. Embora essa verificação seja rápida, em situações de requisições consecutivas, o tempo de espera pode resultar em atrasos na execução do script.

4.6.2 Elementos relacionados à conexão de internet

Com base nos testes realizados e nas execuções descartadas, a variável mais impactante para diferenças no tempo de execução está vinculada à conexão com a internet. Esse impacto é perceptível tanto no cliente quanto no servidor.

Conexão no âmbito do cliente

Ao referir-se ao cliente, estamos abordando a máquina que executa o script do sistema SARUE. A velocidade da internet nessa máquina influencia diretamente o tempo de obtenção dos dados. Em resumo, uma baixa velocidade de internet na máquina impacta principalmente nos resultados obtidos por ela.

Conexão no âmbito do servidor

Esta limitação está relacionada aos servidores de hospedagem do sistema SIGAA. Assim, o tempo de execução do script depende também da conexão desses servidores com a internet. Em períodos de alta demanda no sistema SIGAA, o desempenho do algoritmo é comprometido devido à grande quantidade de requisições ao SIGAA.

4.7 Benchmark para comparação entre as máquinas

A execução desse benchmark surge com o objetivo de obter uma métrica objetiva para avaliar o desempenho das máquinas. Além disso, por meio do benchmark é possível obter avaliações em relação aos resultados quando realizados em outras máquinas.

4.7.1 Metodologia da comparação entre as máquinas

A comparação entre as máquinas foi conduzida por meio da execução de um benchmark no navegador. Dentre os benchmarks executados pelo navegador, o escolhido foi o WebXPRT 4 da Principled Technologies [24]. Este benchmark tem como objetivo avaliar o desempenho das máquinas durante a navegação na web. A ferramenta utilizada proporciona a comparação entre as máquinas em diversos navegadores e sistemas operacionais. Além da medição da velocidade de carregamento de páginas, o benchmark também analisa outros aspectos, tais como a eficiência geral na execução de tarefas relacionadas à navegação web. Essa abordagem abrangente permite uma avaliação mais próxima ao funcionamento no navegador do sistema SARUE.

De acordo com a documentação do WebXPRT 4 [25], os valores obtidos neste benchmark refletem o desempenho de um sistema ao realizar uma variedade de tarefas, como aprimoramento de fotos, organização de álbuns usando inteligência artificial, precificação de opções de ações, criptografia de notas e digitalização OCR, gráficos de vendas e tarefas de dever de casa online. Cada teste é repetido sete vezes, e a pontuação geral é calculada com base na média geométrica das razões das pontuações individuais do sistema de teste em relação ao sistema de calibração. No contexto do WebXPRT 4, onde maiores valores indicam melhor desempenho, a consistência e precisão dos resultados são avaliadas

através de métricas como desvio padrão, intervalo de confiança e erro padrão. Valores que se afastam significativamente da média, são identificados e excluídos para garantir resultados confiáveis. Assim, a metodologia adotada assegura uma avaliação robusta e comparativa do desempenho dos sistemas testados.

4.7.2 Resultados do benchmark para as máquinas diferentes

Com o objetivo de reduzir as variações entre os resultados obtidos, odos os testes foram executados no navegador Chrome[18]. Além disso, durante a execução nenhum outro processo foi utilizado em conjunto ao navegador que estava realizando os testes.

Tabela 4.16: Resultado obtido pelo benchmark da Máquina 1.

Máquina 1		
Item de Teste	Valor	Variação
Pontuação Geral	285	7

Para a máquina 1, descrita pela Tabela 4.1, o resultado obtido pelo benchmark é exibido pela Tabela 4.16.

Tabela 4.17: Resultado obtido pelo benchmark da Máquina 2.

Máquina 2		
Item de Teste	Valor	Variação
Pontuação Geral	237	5

No caso da máquina 2, descrita pela Tabela 4.2, o resultado obtido pelo benchmark é exibido pela Tabela 4.17.

Tabela 4.18: Resultado obtido pelo benchmark da Máquina 3.

Máquina 3		
Item de Teste	Valor	Variação
Pontuação Geral	222	6

Por fim, a máquina 3, descrita pela Tabela 4.3, o resultado obtido pelo benchmark é exibido pela Tabela 4.18.

O resultado obtido pela execução do benchmark fornece uma nota para cada parâmetro verificado na execução. O documento completo, exibindo todas as notas e dados obtidos pela execução do benchmark em cada máquina pode ser visualizado no anexo nas Tabelas IV.1 a IV.3.

Ao considerar as notas gerais obtidas por cada máquina, torna-se evidente uma vantagem no desempenho da máquina 1 em relação às demais. Sua pontuação mais alta

sugere um melhor rendimento em relação às métricas ou critérios de avaliação específicos aplicados.

Quando comparados os resultados das máquinas 2 e 3, é visualizado uma vantagem para a máquina 2, o que não era esperado quando comparado aos resultados obtidos pela execução com MiniCrawler concorrente nessas máquinas.

Vale lembrar que os resultados gerados por esse benchmark devem ser levados em consideração apenas para comparar as máquinas sobre determinada aplicação e podem não resultar no desempenho real em outros contextos ou cenários de uso. Cada benchmark é projetado para avaliar um conjunto específico de tarefas e condições, refletindo, assim, a performance relativa das máquinas testadas dentro desse escopo limitado.

4.7.3 Limitações desse benchmark

Apesar desse benchmark fornecer uma avaliação sobre o desempenho de uma máquina no o uso de navegadores em diferentes plataformas, é importante reconhecer algumas limitações inerentes a essa ferramenta. O WebXPRT[24], ao focar em tarefas específicas, como edição de fotos, manipulação de planilhas e navegação web, pode não capturar completamente a diversidade de operações que os usuários realizam em suas máquinas. Além disso, a variação significativa nas configurações de hardware e sistemas operacionais das máquinas avaliadas pode impactar os resultados, uma vez que diferentes arquiteturas e otimizações podem influenciar o desempenho.

Capítulo 5

Conclusão

Diante do exposto nos resultados desse documento é possível concluir que a abordagem utilizada para acelerar a obtenção dos dados é efetiva.

Quanto ao uso de web crawler no SIGAA, o projeto descrito por esse relatório apresenta um sistema capaz de navegar pelo sistema SIGAA e de indicar os locais de mudanças e alterações. Além disso, o *back-end* do sistema SARUE permite contornar mudanças e o uso em diferentes SIGAAs sem a necessidade de alterar a lógica por trás do sistema.

Em função do cálculo dos indicadores, esse relatório apresenta a forma como foram divididos além da metodologia por trás da remoção das repetições contidas nos dados.

Com base nos resultados, é possível verificar que ambas as soluções para acelerar a obtenção dos dados são efetivas quanto a redução de tempo. Ainda, é possível verificar que o uso de MiniCrawlers concorrentes é atualmente a melhor escolha para obter os dados. Visto que, com essa técnica é possível melhorar o aproveitamento das instâncias ativas e permitir que os períodos de tempo sejam melhores aproveitados.

5.1 Trabalhos futuros

Uma possibilidade é a modificação do algoritmo para obter os dados referentes ao financiamento das ações. Apesar de ser um dado presente dentro do sistema SIGAA, ele é um dado que só pode ser obtido por alguns perfis de usuários e não é visto na página de listar ações de extensão. Através desses dados é possível obter de forma detalhada a origem do financiamento desse projeto, podendo então realizar a cálculo de indicadores voltados a origem orçamentária.

No quesito da credencial necessária para o uso do sistema SARUE, o módulo de autenticação desenvolvido pode ser substituído por um módulo de autenticação fornecido pela UnB. Considerando que esse módulo forneça acesso como se estivesse autenticado, seria

possível por meio dele utilizar o sistema SARUE, sem a necessidade de obter a credencial de um usuário e utilizar ela para realizar buscar.

Atualmente os meses que devem ser pesquisados utilizando a divisão por área do CNPq precisam estar cadastrados no sistema. Uma possibilidade é fazer com que o script preveja a quantidade de ações em determinado mês e possa então decidir qual método de pesquisa ele quer utilizar.

Em função do aumento da quantidade de ações cadastradas, é possível que a divisão por mês e por área do CNPq não sejam capazes de contornar a limitação de 1000 ações exibidas no sistema. Dessa forma, pode ser necessário implementar uma nova divisão que evite repetições entre os dados e garanta o funcionamento do sistema.

Uma solução que pode acelerar a execução do script é por meio de descartar as ações concluídas da busca. Supondo o acesso por meio de uma credencial que possa realizar filtragens pelo status da ação, após obter todos os dados e mantê-los salvos no sistema, as próximas execuções podem descartar ações canceladas e concluídas. Isso deve gerar uma quantidade menor de ações a serem pesquisadas e pode melhorar o tempo de execução do script.

Como sugestão em vista dos indicadores, o cálculo dos indicadores depende de um algoritmo dentro do sistema, então, como sugestão fica a possibilidade de fornecer aos usuários um campo onde eles possam adicionar um novo indicador por meio de uma equação ou texto.

A melhor abordagem para a obtenção dos dados contidos na plataforma SIGAA é por meio de uma API. Então a maior sugestão é justamente incentivar o desenvolvimento de uma API em conjunto com o STI/UnB para obter as ações de extensão.

Referências

[1] Araujo Freitas Júnior, Antonio de: *Resolução nº 7, de 18 de dezembro de 2018*. Diário Oficial da República Federativa do Brasil, 2018. https://www.in.gov.br/materia/-/asset_publisher/Kujrw0TZC2Mb/content/id/55877808, Edição: 243, Seção: 1, Página: 49. 1

[2] Ascom, Gabinete da Reitoria: *Extensão será obrigatória no currículo da graduação em 2023*, 2022. https://dex.unb.br/noticias/931-extensao-sera-obrigatoria-no-curriculo-da-graduacao-em-2023#:~:text=Em%20janeiro%20de%202023%2C%20as,graduaÃğÃčo%20em%20todo%20o%20paÃŋs. 1

[3] Extensão, UnB Decanato de: *O decanato de extensão*, 2023. https://dex.unb.br/odecanatodeextensao. 1

[4] Falcão, Luiz Daniel Costa *et al.*: *A institucionalidade da extensão universitária a partir do sigaa: perspectiva dos docentes extensionistas da universidade federal da paraíba*. 2020. 7

[5] Leite, Petra Raissa Lima Pantoja: *Gestão da política de extensão na unb: desafios e possibilidades*. 2020. https://bdm.unb.br/handle/10483/27871. 7

[6] Patterson, D.A. e J.L. Hennessy: *Computer Organization and Design ARM Edition: The Hardware Software Interface*. ISSN. Elsevier Science, 2016, ISBN 9780128018354. https://books.google.com.br/books?id=Pz-XCgAAQBAJ. 8, 48

[7] Maximiliano Júnior, Manoel, Ana Inês Sousa, Dalva Maria de Oliveira Silva, Etevaldo Almeida Silva, Maristela Helena Zimmer Bortolini, Nadege da Silva Dantas e Regina Lúcia Monteiro Henriques: *Indicadores Brasileiros de Extensão Universitária (IBEU)*. EDUFCG, Campina Grande - PB, 2017. http://dspace.sti.ufcg.edu.br:8080/jspui/handle/riufcg/30201. 9

[8] Educação Superior Brasileiras, Fórum de Pró-Reitores das Instituições Públicas de: *Política Nacional de Extensão Universitária*. UFSC, Manaus - AM, 2012. https://proex.ufsc.br/files/2016/04/PolÃŋtica-Nacional-de-ExtensÃčo-UniversitÃąria-e-book.pdf. 9

[9] Jarmul, Katharine e Richard Lawson: *Python Web Scraping*. Packt Publishing Ltd, 2017. 9, 10

[10] Python Software Foundation: *Python 3 Documentation*, 2023. https://docs.python.org/3/. 11, 16, 34

[11] Pypi - Python Package Index: *Pip 3 Documentation*, 2023. https://pypi.org/project/pip/. 11, 16, 34

[12] SeleniumHQ: *Selenium Documentation*, 2023. https://www.selenium.dev/documentation/en/. 11

[13] Oliveira, Sergio: *python-dotenv Documentation*, 2023. https://github.com/theskumar/python-dotenv. 12

[14] Costa-Luis, Casper da: *tqdm Documentation*, 2023. https://github.com/tqdm/tqdm. 12

[15] Python Software Foundation: *JSON Documentation*, 2023. https://docs.python.org/3/library/json.html. 13

[16] al., Wes McKinney et: *pandas Documentation*, 2023. https://pandas.pydata.org/docs/. 13

[17] Vilas Novas Soares, Carlos Gabriel: *Sarue: Sistema de acompanhamento de registros universitários de extensão*. 2023. noprelo. 14

[18] Google: *Google Chrome*. https://www.google.com/chrome/, 2023. Acessado em 17/11/2023. 30, 57, 59

[19] Microsoft Corporation: *Microsoft Edge*. https://www.microsoft.com/edge/, 2023. Acessado em 17/11/2023. 30

[20] Mozilla Foundation: *Mozilla Firefox*. https://www.mozilla.org/firefox/, 2023. Acessado em 17/11/2023. 30

[21] Apple Inc.: *Safari*. https://www.apple.com/safari/, 2023. Acessado em 17/11/2023. 31

[22] Rodrigues, Walton Alencar: *Acórdão 461/2022 - plenário*. TCU Pesquisa Integrada - Acórdão, 2022. https://portal.tcu.gov.br/imprensa/noticias/tcu-avalia-indicadores-de-gestao-e-desempenho-das-universidades-federais.htm, PROCESSO: 026.147/2020-3, NÚMERO DA ATA: 8/2022 - Plenário. 38

[23] Sadéri da Silva, João Pedro: *Github - sarue-sigaa*, 2023. https://github.com/jpsaderi/sarue-sigaa.git. 47

[24] Technologies, Principled: *Webxprt benchmark*, 2023. https://www.principledtechnologies.com/benchmarkxprt/webxprt/, Acessado em 17/11/2023. 58, 60

[25] Technologies, Principled: *Webxprt 4 results calculation and confidence interval*, 2022. https://www.principledtechnologies.com/benchmarkxprt/counter.php?inline=true&redirect=/benchmarkxprt/whitepapers/webxprt/WebXPRT-4-results-calculation.pdf, Acessado em 17/11/2023. 58

Anexo I

Relatório dos tempos obtidos pela máquina 1

```
================================================================================
Instâncias = 12 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sun Nov  5 13:51:27 2023
End time: Sun Nov  5 14:17:50 2023
Elapsed time: 26:23 minutes
================================================================================
Instâncias = 9 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov  6 00:02:07 2023
End time: Mon Nov  6 00:28:34 2023
Elapsed time: 26:26 minutes
================================================================================
Instâncias = 6 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov  6 10:29:27 2023
End time: Mon Nov  6 11:01:44 2023
Elapsed time: 32:16 minutes
================================================================================
Instâncias = 3 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov  6 11:38:45 2023
```

```
End time: Mon Nov  6 12:30:39 2023
Elapsed time: 51:53 minutes
================================================================================
Instâncias = 1 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov  6 14:31:54 2023
End time: Mon Nov  6 16:36:21 2023
Elapsed time: 124:27 minutes
================================================================================

================================================================================
Instâncias = 9 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sun Nov  5 14:27:46 2023
End time: Sun Nov  5 14:58:09 2023
Elapsed time: 30:23 minutes
================================================================================
Instâncias = 6 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov  6 11:07:59 2023
End time: Mon Nov  6 11:38:38 2023
Elapsed time: 30:39 minutes
================================================================================
Instâncias = 3 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov  6 12:32:18 2023
End time: Mon Nov  6 13:16:47 2023
Elapsed time: 44:28 minutes
================================================================================
Instâncias = 1 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov  6 17:56:36 2023
End time: Mon Nov  6 19:57:17 2023
Elapsed time: 120:41 minutes
================================================================================
```

```
================================================================================
Instâncias = 6 - PARALLEL - SEMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sun Nov  5 18:24:20 2023
End time: Sun Nov  5 19:13:30 2023
Elapsed time: 49:09 minutes
================================================================================
Instâncias = 3 - PARALLEL - SEMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov  6 13:16:55 2023
End time: Mon Nov  6 14:06:17 2023
Elapsed time: 49:22 minutes
================================================================================
Instâncias = 1 - PARALLEL - SEMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov  6 22:38:54 2023
End time: Tue Nov  7 00:38:51 2023
Elapsed time: 119:56 minutes
================================================================================

================================================================================
Instâncias = 3 - PARALLEL - YEAR
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sun Nov  5 22:40:25 2023
End time: Sun Nov  5 23:41:13 2023
Elapsed time: 60:47 minutes
================================================================================
Instâncias = 1 - PARALLEL - YEAR
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Tue Nov  7 19:16:03 2023
End time: Tue Nov  7 21:20:09 2023
Elapsed time: 124:06 minutes
================================================================================
```

```
================================================================================
Instâncias = 12 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Tue Nov  7 21:36:46 2023
End time: Tue Nov  7 21:56:58 2023
Elapsed time: 20:12 minutes
================================================================================
Instâncias = 9 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Tue Nov  7 22:08:05 2023
End time: Tue Nov  7 22:29:28 2023
Elapsed time: 21:22 minutes
================================================================================
Instâncias = 6 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Tue Nov  7 22:44:59 2023
End time: Tue Nov  7 23:11:37 2023
Elapsed time: 26:37 minutes
================================================================================
Instâncias = 3 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Tue Nov  7 23:11:54 2023
End time: Tue Nov  7 23:55:45 2023
Elapsed time: 43:50 minutes
================================================================================
Instâncias = 1 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Wed Nov  8 09:44:18 2023
End time: Wed Nov  8 11:46:59 2023
Elapsed time: 122:41 minutes
================================================================================
```

Anexo II

Relatório dos tempos obtidos pela máquina 2

```
================================================================================
Instâncias = 12 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Wed Nov  8 14:34:38 2023
End time: Wed Nov  8 15:11:05 2023
Elapsed time: 36:26 minutes
================================================================================
Instâncias = 9 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Wed Nov  8 15:11:14 2023
End time: Wed Nov  8 15:48:16 2023
Elapsed time: 37:02 minutes
================================================================================
Instâncias = 6 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Wed Nov  8 16:04:40 2023
End time: Wed Nov  8 16:46:19 2023
Elapsed time: 41:38 minutes
================================================================================
Instâncias = 3 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Wed Nov  8 16:46:28 2023
```

```
End time: Wed Nov  8 17:52:14 2023
Elapsed time: 65:46 minutes
================================================================================
Instâncias = 1 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Wed Nov  8 19:19:27 2023
End time: Wed Nov  8 21:36:22 2023
Elapsed time: 136:54 minutes
================================================================================

================================================================================
Instâncias = 9 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Fri Nov 10 11:28:39 2023
End time: Fri Nov 10 12:09:51 2023
Elapsed time: 41:11 minutes
================================================================================
Instâncias = 6 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Fri Nov 10 12:10:00 2023
End time: Fri Nov 10 12:50:29 2023
Elapsed time: 40:28 minutes
================================================================================
Instâncias = 3 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Fri Nov 10 12:50:38 2023
End time: Fri Nov 10 13:48:15 2023
Elapsed time: 57:37 minutes
================================================================================
Instâncias = 1 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Fri Nov 10 17:21:09 2023
End time: Fri Nov 10 19:49:41 2023
Elapsed time: 148:32 minutes
================================================================================
```

```
================================================================================
Instâncias = 6 - PARALLEL - SEMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Fri Nov 10 19:57:57 2023
End time: Fri Nov 10 20:42:44 2023
Elapsed time: 44:47 minutes
================================================================================
Instâncias = 3 - PARALLEL - SEMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Fri Nov 10 20:42:53 2023
End time: Fri Nov 10 21:43:06 2023
Elapsed time: 60:12 minutes
================================================================================
Instâncias = 1 - PARALLEL - SEMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sat Nov 11 11:54:24 2023
End time: Sat Nov 11 14:19:03 2023
Elapsed time: 144:38 minutes
================================================================================

================================================================================
Instâncias = 3 - PARALLEL - YEAR
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sat Nov 11 14:19:12 2023
End time: Sat Nov 11 15:36:16 2023
Elapsed time: 77:04 minutes
================================================================================
Instâncias = 1 - PARALLEL - YEAR
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sat Nov 11 15:36:25 2023
End time: Sat Nov 11 18:04:04 2023
Elapsed time: 147:39 minutes
================================================================================
```

```
================================================================================
Instâncias = 12 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov 13 19:03:04 2023
End time: Mon Nov 13 19:30:41 2023
Elapsed time: 27:36 minutes
================================================================================
Instâncias = 9 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov 13 19:30:59 2023
End time: Mon Nov 13 20:00:35 2023
Elapsed time: 29:35 minutes
================================================================================
Instâncias = 6 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov 13 20:00:53 2023
End time: Mon Nov 13 20:36:33 2023
Elapsed time: 35:40 minutes
================================================================================
Instâncias = 3 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov 13 21:47:43 2023
End time: Mon Nov 13 22:45:11 2023
Elapsed time: 57:27 minutes
================================================================================
Instâncias = 1 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Mon Nov 13 22:49:10 2023
End time: Tue Nov 14 01:17:45 2023
Elapsed time: 148:35 minutes
================================================================================
```

Anexo III

Relatório dos tempos obtidos pela máquina 3

```
================================================================================
Instâncias = 12 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Tue Nov 14 20:05:19 2023
End time: Tue Nov 14 20:44:49 2023
Elapsed time: 39:30 minutes
================================================================================
Instâncias = 9 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Wed Nov 15 12:37:17 2023
End time: Wed Nov 15 13:16:24 2023
Elapsed time: 39:07 minutes
================================================================================
Instâncias = 6 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Wed Nov 15 14:48:10 2023
End time: Wed Nov 15 15:33:08 2023
Elapsed time: 44:58 minutes
================================================================================
Instâncias = 3 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Wed Nov 15 15:34:11 2023
```

```
End time: Wed Nov 15 16:40:00 2023
Elapsed time: 65:49 minutes
================================================================================
Instâncias = 1 - PARALLEL - TRIMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Wed Nov 15 18:42:05 2023
End time: Wed Nov 15 21:27:09 2023
Elapsed time: 165:03 minutes
================================================================================

================================================================================
Instâncias = 9 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Thu Nov 16 10:47:33 2023
End time: Thu Nov 16 11:28:59 2023
Elapsed time: 41:25 minutes
================================================================================
Instâncias = 6 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Thu Nov 16 11:39:58 2023
End time: Thu Nov 16 12:23:16 2023
Elapsed time: 43:18 minutes
================================================================================
Instâncias = 3 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Thu Nov 16 12:23:38 2023
End time: Thu Nov 16 13:19:14 2023
Elapsed time: 55:36 minutes
================================================================================
Instâncias = 1 - PARALLEL - QUARTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Thu Nov 16 14:04:29 2023
End time: Thu Nov 16 16:25:12 2023
Elapsed time: 140:43 minutes
================================================================================
```

```
================================================================================
Instâncias = 6 - PARALLEL - SEMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Thu Nov 16 18:33:41 2023
End time: Thu Nov 16 19:15:54 2023
Elapsed time: 42:12 minutes
================================================================================
Instâncias = 3 - PARALLEL - SEMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Thu Nov 16 20:44:59 2023
End time: Thu Nov 16 21:45:30 2023
Elapsed time: 60:31 minutes
================================================================================
Instâncias = 1 - PARALLEL - SEMESTER
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Thu Nov 16 21:54:59 2023
End time: Wed Nov 17 00:09:12 2023
Elapsed time: 134:13 minutes
================================================================================

================================================================================
Instâncias = 3 - PARALLEL - YEAR
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sat Nov 18 17:52:42 2023
End time: Sat Nov 18 19:00:12 2023
Elapsed time: 67:30 minutes
================================================================================
Instâncias = 1 - PARALLEL - YEAR
2020/2022 - (1/12)
Quantidade de ações = 4809
Start time: Sat Nov 18 19:00:25 2023
End time: Sat Nov 18 21:11:52 2023
Elapsed time: 131:27 minutes
================================================================================
```

```
================================================================================
Instâncias = 12 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sat Nov 18 21:12:33 2023
End time: Sat Nov 18 21:33:20 2023
Elapsed time: 20:47 minutes
================================================================================
Instâncias = 9 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sat Nov 18 21:47:02 2023
End time: Sat Nov 18 22:08:58 2023
Elapsed time: 21:56 minutes
================================================================================
Instâncias = 6 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sat Nov 18 22:09:17 2023
End time: Sat Nov 18 22:44:08 2023
Elapsed time: 34:50 minutes
================================================================================
Instâncias = 3 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sat Nov 18 22:47:01 2023
End time: Sat Nov 18 23:32:13 2023
Elapsed time: 45:12 minutes
================================================================================
Instâncias = 1 - CONCURRENT
2020/2022 - (1/12)
Quantidade de ações = 4824
Start time: Sat Nov 18 23:32:41 2023
End time: Sun Nov 19 01:36:21 2023
Elapsed time: 123:40 minutes
================================================================================
```

Anexo IV

Resultados completos do benchmark WebXPRT 4

Tabela IV.1: Resultado do benchmark - Máquina 1.

Item de Teste	Valor	
Teste	WebXPRT 4 (v3.73)	
ID do Teste	228487	
Data	2023-11-19 15:59:52	
Navegador	Chrome 119.0.0.0	
Item de Teste	Valor	Variação
Pontuação Geral	285	7
Aprimoramento de Foto	265	5.85%
Organizar Álbum usando IA	1769	0.85%
Precificação de Opções de Ações	75	11.4%
Notas Criptografadas e Digitalização OCR	726	0.18%
Gráficos de Vendas	252	4.4%
Tarefa de Casa Online	1127	2.08%

Tabela IV.2: Resultado do benchmark - Máquina 2.

Item de Teste	Valor	
Teste	WebXPRT 4 (v3.73)	
ID do Teste	230664	
Data	2023-11-19 20:42:38	
Navegador	Chrome 119.0.0.0	
Item de Teste	Valor	Variação
Pontuação Geral	237	5
Aprimoramento de Foto	317	4.18%
Organizar Álbum usando IA	1804	0.91%
Precificação de Opções de Ações	95	4.62%
Notas Criptografadas e Digitalização OCR	1014	0.72%
Gráficos de Vendas	241	3.57%
Tarefa de Casa Online	1653	1.83%

Tabela IV.3: Resultado do benchmark - Máquina 3.

Item de Teste	Valor	
Teste	WebXPRT 4 (v3.73)	
ID do Teste	230593	
Data	2023-11-19 15:59:52	
Navegador	Chrome 119.0.0.0	
Item de Teste	Valor	Variação
Pontuação Geral	222	6
Aprimoramento de Foto	497	0.3%
Organizar Álbum usando IA	1615	0.48%
Precificação de Opções de Ações	115	11.83%
Notas Criptografadas e Digitalização OCR	978	1.63%
Gráficos de Vendas	223	4.02%
Tarefa de Casa Online	1610	1.59%