

Estudo de caracterização de aplicações desktop de repositórios do Github por meio de Inteligência Artificial

Guilherme Gabriel Silva Pereira¹, Henrique Penna Forte Monteiro², Lucas Ângelo Oliveira Martins Rocha³, Victor Boaventura Góes Campos⁴, Vinicius Marini Costa e Oliveira⁵

**¹Instituto de Informática e Ciências Exatas– Pontifícia Universidade de Minas Gerais (PUC MINAS)
Belo Horizonte – MG – Brasil**

¹ggspereira@sga.pucminas.br, ²henrique.forte@sga.pucminas.br,
³laomrocha@sga.pucminas.br, ⁴vbgcampos@sga.pucminas.br,
⁵vinicius.marini@sga.pucminas.br

1. Introdução

Ao longo dos anos, os navegadores evoluíram e passaram a ter mais funcionalidades e suportar aplicações mais elaboradas. Nesse contexto, a evolução da internet e a praticidade de acessar os sistemas via web diminuiu a necessidade de se ter uma aplicação desktop. Contudo, alguns domínios de aplicações necessitam de ser desktop.

Nesse sentido, foi pensado que ainda existem domínios populares no contexto de aplicações desktop. Esta pesquisa é importante porque, como já mencionado, ajuda a identificar os domínios que ainda são populares para aplicações desktop. Isso é importante porque, muitas vezes, os desenvolvedores de software, gestores de novos projetos e analistas seguem uma tendência e acabam criando aplicações para um domínio que talvez já esteja em declínio em desktop.

Como resultado, o projeto pode ajudar os desenvolvedores a criarem aplicações para os domínios que ainda são populares, por meio da caracterização destes principais domínios de aplicações desktop na atualidade.

A pesquisa também é importante para outros atores do mercado de TI, como fornecedores de soluções e produtos. Isto porque, ao identificar os principais domínios de aplicações desktop, eles podem direcionar suas ofertas para atender a essa demanda.

Além disso, a pesquisa pode contribuir para o aprimoramento das ferramentas de análise de mercado, uma vez que elas poderão ser ajustadas para melhor atender às demandas do mercado de aplicações desktop.

A pesquisa foi realizada com base em um conjunto de dados coletados a partir de dados de repositórios do Github de aplicações desktop. Os dados foram analisados e as principais tendências foram identificadas.

1.1 Objetivo principal

O objetivo principal do projeto é testar a hipótese inicial e registrar o resultado. Dessa forma, por meio de pesquisa e análise de dados, busca-se caracterizar domínios que ainda são populares para aplicações desktop.

1.2 GQM (Goal, Question, Metric)

Análise de repositórios de aplicações desktop

com o objetivo de caracterizar aplicações desktop e seus domínios

com relação a quantidade de aplicações e suas respectivas métricas

do ponto de vista de analistas, gerentes e clientes de novos projetos

no contexto dos repositórios do Github que possuem dependências de aplicações desktop das linguagens Java, C# e JavaScript.

2. Referencial Teórico

A seguir serão descritos o referencial teórico do trabalho, foram buscados artigos que se relacionam com o tema da pesquisa. Os temas selecionados foram Aplicações Desktop, Busca de dados, Análise e grupamento de palavras e Métricas.

A seguir, os trabalhos relacionados:

1. *Transforming JavaScript-Based Web Application to Cross-Platform Desktop with Electron:*

- a. A obra "*Transforming JavaScript-Based Web Application to Cross-Platform Desktop with Electron*" mostra como transformar uma aplicação web baseada em JavaScript em um aplicativo de desktop multiplataforma usando o Electron. O livro começa com uma introdução ao Electron e às suas vantagens em relação às aplicações web tradicionais. Em seguida, o leitor é guiado através do processo de criação de um aplicativo de desktop básico usando o Electron. O livro também aborda tópicos avançados, como a criação de interfaces gráficas ricas e acessíveis e a integração de bibliotecas externas. Com este livro, os leitores aprenderão como o Electron pode simplificar o desenvolvimento de aplicativos desktop e torná-los acessíveis a um público mais amplo.
2. *An In-depth Comparison Of Software Frameworks For Developing Desktop Applications Using Web Technologies*
 - a. A obra "*AN IN-DEPTH COMPARISON OF SOFTWARE FRAMEWORKS FOR DEVELOPING DESKTOP APPLICATIONS USING WEB TECHNOLOGIES*" é um estudo aprofundado sobre os diferentes frameworks de software disponíveis para o desenvolvimento de aplicações desktop usando tecnologias web. O estudo analisa as vantagens e desvantagens de cada um dos frameworks e fornece uma recomendação sobre qual deles é o melhor para cada tipo de projeto.
3. Usando desktop na educação
 - a. O livro "Usando o desktop na educação" é um guia prático para ajudar os educadores a usar o desktop para melhorar a educação. Ele fornece instruções passo-a-passo sobre como configurar um desktop, personalizar as configurações, instalar software educacional e criar conteúdo educacional. Também aborda como os educadores podem usar o desktop para se comunicar com outros educadores e alunos.
4. *Cross-Platform Desktop Applications: Using Node, Electron, and NW.js*
 - a. "*Cross-Platform Desktop Applications: Using Node, Electron, and NW.js*" é um livro que ensina como desenvolver aplicativos desktop usando Node, Electron e NW.js. Ele fornece uma introdução ao desenvolvimento de aplicativos desktop e mostra como criar aplicativos

usando essas três tecnologias. Também aborda tópicos avançados, como o gerenciamento de janelas, a criação de interfaces gráficas de usuário e o acesso ao sistema operacional.

5. *Challenges in Developing Desktop Web Apps: a Study of Stack Overflow and GitHub*

- a. Desenvolver aplicativos da Web para desktop pode ser um desafio, especialmente quando se trata de garantir que eles sejam compatíveis com todos os principais navegadores. Neste estudo, analisa-se o Stack Overflow e o GitHub para identificar os principais problemas enfrentados pelos desenvolvedores ao criar aplicativos da Web para desktop. Os resultados mostraram que os principais problemas enfrentados são a compatibilidade do navegador, a falta de suporte para determinadas tecnologias e a dificuldade de implementar recursos avançados. Além disso, descobriram que os desenvolvedores estão cada vez mais optando por frameworks de terceiros para simplificar o processo de desenvolvimento.

6. *Web Frameworks for Desktop Apps: an Exploratory Study*

- a. Neste estudo, exploramos os principais frameworks de aplicativos da Web para desktop. Nosso objetivo era avaliar as vantagens e desvantagens de cada um dos frameworks em relação à produtividade, flexibilidade e facilidade de uso. Para isso, analisou-se a documentação e os exemplos disponíveis para cada um dos frameworks. Além disso, entrevistamos desenvolvedores experientes para obter suas opiniões sobre os frameworks. Os resultados mostraram que os frameworks mais populares são o Electron, o NW.js e o Qt for WebAssembly. Esses frameworks oferecem uma boa produtividade, flexibilidade e facilidade de uso. Também apresentam algumas desvantagens, como a falta de suporte para algumas plataformas e a necessidade de um conhecimento mais avançado de JavaScript.

7. *What's in a GitHub Star? Understanding Repository Starring Practices in a Social Coding Platform*

- a. Este estudo analisou o uso de estrelas em GitHub e tentou compreender as práticas de utilização de estrelas. Os autores analisaram um conjunto de dados de 2,8 milhões de estrelas de usuários em projetos de software e descobriram que as estrelas são usadas de diversas maneiras. Alguns usuários estrelam muitos projetos, outros poucos, e alguns usuários adicionam estrelas somente a projetos que eles mesmos criaram. Os autores também descobriram que as estrelas são usadas para indicar diversos tipos de interesse, desde o interesse em usar o código do projeto para o seu próprio uso, até o interesse em acompanhar o projeto para ver as atualizações. As estrelas também podem ser usadas para indicar apoio a um projeto ou para mostrar gratidão aos seus criadores. Este estudo fornece uma compreensão valiosa sobre como e por que os usuários estrelam os projetos de outros usuários na plataforma GitHub.
8. *Cross-Platform Development Frameworks: Overview of contemporary technologies and methods for cross-platform application development*
 - a. A obra *Cross-Platform Development Frameworks: Overview of contemporary technologies and methods for cross-platform application development* apresenta uma visão geral das tecnologias e métodos contemporâneos para o desenvolvimento de aplicativos multiplataforma. O livro começa com uma discussão sobre o que é necessário para criar um aplicativo multiplataforma e quais são as vantagens e desvantagens dessa abordagem. Em seguida, o livro apresenta uma visão geral das principais tecnologias e frameworks de desenvolvimento multiplataforma disponíveis atualmente. Por fim, o livro fornece uma discussão sobre os métodos e técnicas de desenvolvimento de aplicativos multiplataforma.
9. *A Systematic Mapping Study of Software Development With GitHub*
 - a. *A Systematic Mapping Study of Software Development With GitHub* é um estudo sistemático sobre o desenvolvimento de software com o GitHub. O objetivo deste estudo foi identificar, classificar e analisar as principais tendências e práticas de desenvolvimento de software com o GitHub. Foram considerados trabalhos publicados entre 2011 e 2016 em diversas

bases de dados. Após a análise dos dados, foi possível identificar quatro principais tendências no desenvolvimento de software com GitHub: aumento do uso de GitHub para fins de desenvolvimento de software; crescente integração de GitHub com outras ferramentas e plataformas; crescente adoção de metodologias ágeis de desenvolvimento de software; e crescente utilização de GitHub para fins de colaboração e compartilhamento de código.

10. *Bringing Desktop Applications to the Web*

- a. A obra "*Bringing Desktop Applications to the Web*" trata de como aplicativos de desktop podem ser levados para a web. O livro discute as vantagens e desvantagens de fazer isso, bem como os desafios técnicos envolvidos. Ele também apresenta alguns exemplos de aplicativos de desktop que foram bem-sucedidos na web.

11. *Got issues? Who cares about it? A large scale investigation of issue trackers from GitHub*

- a. A obra "*Got issues? Who cares about it? A large scale investigation of issue trackers from GitHub*" é uma investigação em larga escala dos rastreadores de problemas do GitHub. O objetivo desta investigação foi analisar como os usuários do GitHub interagem com os rastreadores de problemas e quais são as principais dificuldades que eles enfrentam ao usá-los. Os autores analisaram um conjunto de dados de 6,7 milhões de problemas abertos no GitHub e descobriram que apenas um terço deles recebe algum tipo de interação. Além disso, eles também identificaram vários problemas com os rastreadores de problemas do GitHub, incluindo a falta de um mecanismo de notificação eficaz, a dificuldade de navegação e a falta de um sistema de classificação eficaz.

12. *A Cross-Repository Model for Predicting Popularity in GitHub*

- a. A *Cross-Repository Model for Predicting Popularity in GitHub* é um modelo de previsão de popularidade para o GitHub. Ele foi criado para ajudar os usuários do GitHub a escolherem os projetos mais populares para contribuir. O modelo é baseado em três fatores: o número de estrelas, o número de *forks* e o número de *watchers*.

13. *Detecting similar sentences using word embedding*
 - a. A obra *Detecting similar sentences using word embedding* é um estudo sobre modelos de *embedding* de palavras e sua aplicação na detecção de sentenças semelhantes. Os autores analisaram diferentes modelos de *embedding* e concluíram que o modelo de *embedding* de palavras é um método eficaz para a detecção de sentenças semelhantes.
14. *Word Graph-Based Multi-sentence Compression: Re-ranking Candidates Using Frequent Words*
 - a. A obra "*Word Graph-Based Multi-sentence Compression: Re-ranking Candidates Using Frequent Words*" é um estudo sobre a compressão de múltiplas sentenças usando gráficos de palavras. O estudo propõe um novo método para re-classificar candidatos usando palavras frequentes, o que pode melhorar a precisão da compressão.
15. *Analysis of The Characteristics of Similar Words Computed by Word Embeddings*
 - a. A obra "*Analysis of The Characteristics of Similar Words Computed by Word Embeddings*" analisa as características de palavras similares computadas por meio de *embeddings* de palavras.

2.1. Trabalhos relacionados

Nesta seção serão apresentados por meio da Figura 1, os trabalhos relacionados ao tema da pesquisa. Esses foram buscados em repositórios de artigos e textos acadêmicos (*Google Scholar / IEEE*). Após análise de artigos, foram selecionados os 15 artigos a seguir:

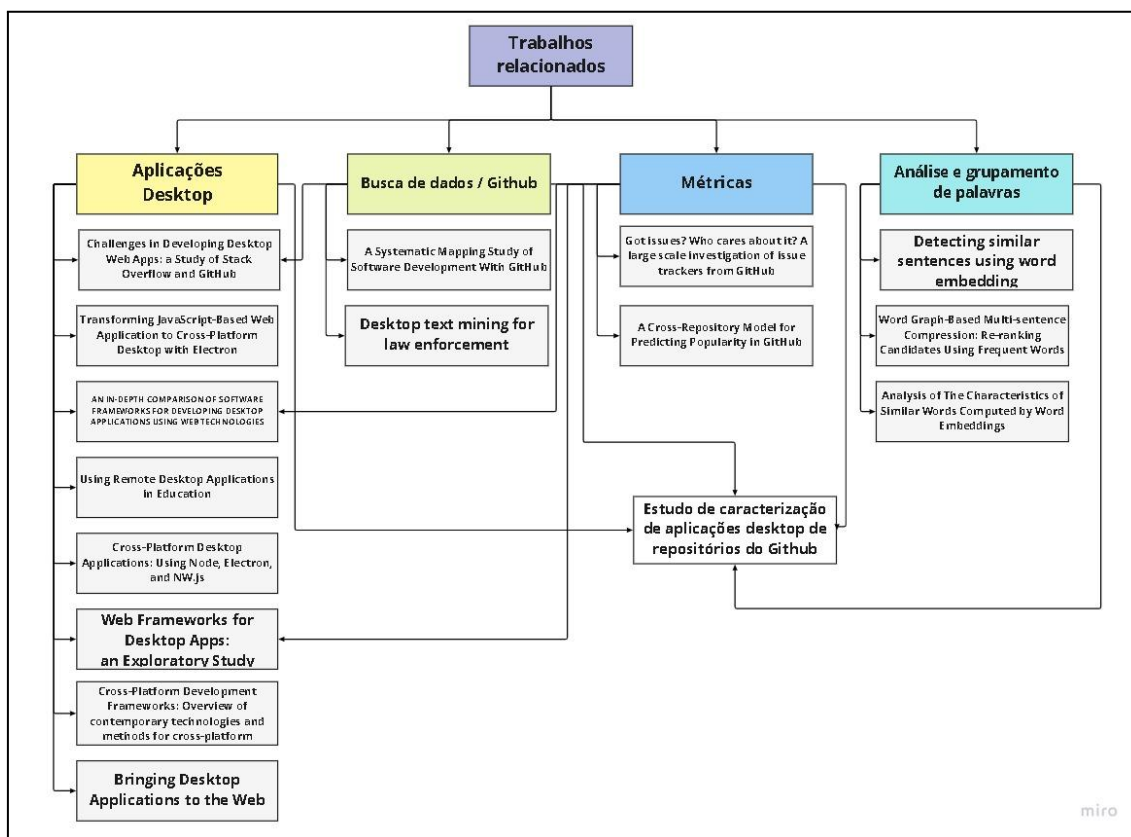


Figura 1 - Trabalhos relacionados

A seguir são detalhadas as áreas dos trabalhos relacionados. A primeira, "Aplicações Desktop" envolve trabalhos que buscam estudar algum aspecto relacionado com aplicações que foram desenvolvidas para serem instaladas no sistema operacional.

Pelo fato do estudo estar focado em aplicações que estão no Github, foi estudando por meio de alguns artigos como encontrar informações relevantes por meio de API e busca massiva de dados, representado pela área "Busca de dados / Github".

Na área de "Métricas", foram buscados artigos que auxiliam a extrair métricas dos repositórios do Github que estão sendo estudados. Os artigos foram lidos e, a partir deles, foi possível identificar as métricas que foram extraídas dos repositórios. São elas: número de *issues* por ano, número de *pull requests* por ano, número de *releases* por ano e informações de dependências no código. Além dessas métricas, também serão extraídas as seguintes informações: nome do repositório, linguagem utilizada, data de criação do repositório e data da última atualização.

Para completar, é necessário definir a qual domínio os repositórios das aplicações desktop se destinam. Com isso, foram estudados alguns artigos na área “Análise e agrupamento de palavras”, que trazem algumas formas de conseguir detectar as palavras chaves por meio de processamento de linguagem natural. A análise das palavras chave é importante para compreender o domínio dos repositórios e, assim, poder agrupá-los de forma adequada.

3. Metodologia

Para buscar os repositórios populares, foi utilizado o site do Github. O site do Github é uma plataforma de desenvolvimento colaborativo que oferece aos seus usuários um controle de versão distribuído do código fonte, além de um sistema de gerenciamento de tarefas e de um fórum para discussão. A API do Github permite que os usuários acessem os dados do site de forma programática.

A análise de artigos, destacou que a dependência do Electron do JavaScript está em alta na quantidade de aplicações desenvolvidas atualmente devido a popularidade da própria linguagem. Outro fator importante, é que foi possível pesquisar pelo próprio GitHub quais são as aplicações em Electron de forma nativa da plataforma. Além disso, para o C#, é possível identificar que a biblioteca WindowsForms é uma das mais populares para aplicações desktop em C# de acordo com os estudos, isto pelo fato de ser uma das bibliotecas para aplicações desktop mais tradicionais da linguagem C# e ser relativamente prática de se desenvolver utilizando o Visual Studio..

A partir destes dados, é possível concluir que o Electron é uma das principais ferramentas para o desenvolvimento de aplicações desktop. Além disso, a biblioteca WindowsForms é uma das mais populares para aplicações desktop em C#.

Para detectar os repositórios de JavaScript, foi utilizado o Selenium em um script Python. O Selenium é uma ferramenta de automação de testes para aplicações web. Com o Selenium, foi possível detectar os repositórios que dependem da dependência de aplicação desktop Electron, por meio dos repositórios dependentes do repositório do próprio Electron no Github. Já, para capturar as aplicações da C#, foi utilizando a API de GraphQL do próprio Github, em ordem decrescente dos repositórios

com mais estrelas, além disso também foi utilizado em Selenium para filtrar quais repositórios possuem o WindowsForms e capturar os dados restantes de métricas.

A coleta de dados é realizada através de um *script* que estuda repositórios das linguagens JavaScript que possuam como dependência desktop o Electron e repositórios C# que utilizem a biblioteca do WindowsForms.

O *script* obtém dados dos repositórios do Github que são analisadas pela inteligência artificial GTP-3 da OpenIA, no qual delimita os principais domínios da aplicação de cada repositório de acordo com a descrição na página inicial dele no GitHub.

Os dados registrados são analisados e, a partir daí, é possível tirar conclusões sobre os domínios mais populares para aplicações desktop. Além disso, também é possível identificar quais são os domínios que não estão em declínio.

É importante ressaltar que, para garantir que os domínios não estão em declínio de popularidade, o estudo foi feito utilizando métricas de popularidade do Github, como quantidade de estrelas, atividades de issues, pull requests e releases ativas e com datas de contribuições.

Para representar a metodologia da pesquisa, foi confeccionada a Figura 2, que representa todas as etapas do processo de busca e categorização de repositórios e será descrita em sequência.

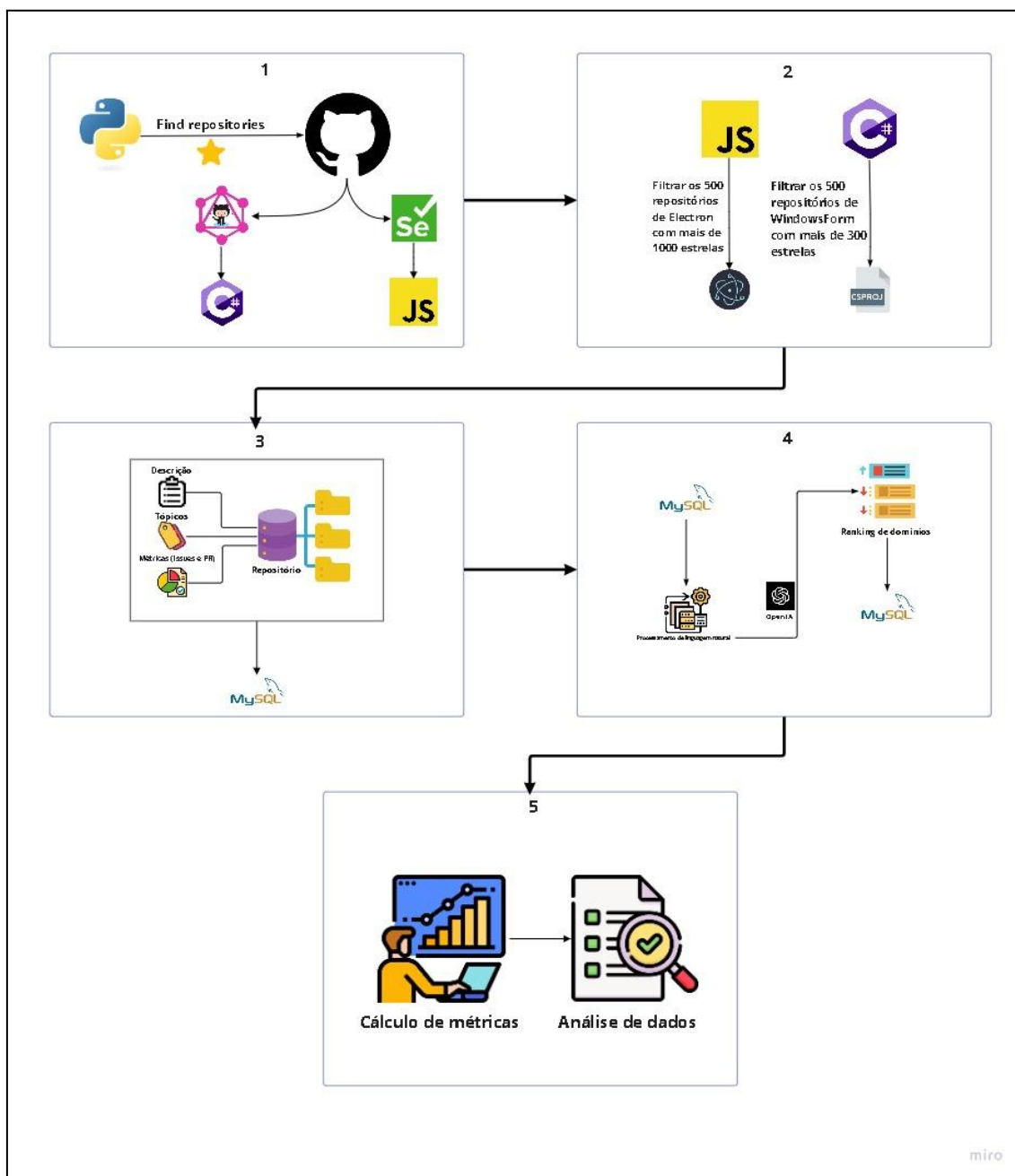


Figura 2 - Metodologia de pesquisa

O passo 1 desta metodologia, foi buscar repositórios populares por meio de scripts desenvolvidos em Python, que acessam o site do Github por meio da API (Interface de Programação de Aplicação, do inglês *Application Programming Interface*) de GraphQL e o Selenium. Com a API, foi buscado os repositórios das linguagens C# que possuem WindowsForms, ou seja, uma aplicação desktop. Por fim, para detectar os

repositórios de JavaScript, foi utilizado o Selenium que detecta os repositórios dependentes da dependência de aplicação desktop Electron.

Após isso, começa o passo 2, que se baseia em processar todos os repositórios detectados, capturando as informações das dependências, textos e as métricas do repositório, que inclui dados de quantitativos como quantidade de estrelas, *issues* e *pull requests* em diferentes estados, isto para cada repositório.

Já o passo 3, salva as informações de métricas, descrição e tópicos (tags do Github) dos repositórios no MySQL. Tais informações serão essenciais para o objetivo do projeto nos próximos passos.

No passo 4, inicia-se um estágio muito complexo, no qual se baseia em utilizar processamento de linguagem natural da OpenIA, criado a partir da GPT-3, para detectar as palavras chaves com os melhores níveis de precisão. Sendo possível assim, determinar automaticamente qual é o mais provável domínio de cada repositório a partir dos dados salvos no MySQL. Ao finalizar, os prováveis domínios e seus percentuais são salvos no MySQL.

Ao fim, no passo 5, é analisado os resultados e calculadas as métricas para auxiliar a responder as perguntas levantadas e alcançar o objetivo definido nas fases iniciais do trabalho.

4. Resultados (De acordo apenas, com a POC inicial)

Os resultados da pesquisa mostraram que ainda existem alguns domínios populares para aplicações desktop, como jogos, gravadores de tela, visualização de dados, *frameworks* e bibliotecas, editores de texto, ferramentas para carteiras digitais, criptomoedas, *backup*, extensões de navegador, inteligência artificial como *machine learning* e detecção de objetos, até mesmo software voltados para funções bem específicas como modificadores de jogos da Steam e também para baixar imagens e vídeos do Youtube.

5. Conclusões

Referências

16. https://link.springer.com/chapter/10.1007/978-981-13-1056-0_56
(Transforming JavaScript-Based Web Application to Cross-Platform Desktop with Electron)
17. <http://www.jsju.org/index.php/journal/article/view/314>(AN IN-DEPTH COMPARISON OF SOFTWARE FRAMEWORKS FOR DEVELOPING DESKTOP APPLICATIONS USING WEB TECHNOLOGIES)
18. <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.465.1768&rep=rep1&type=pdf> (Usando desktop na educação)
19. <https://books.google.com.br/books?hl=pt-BR&lr=&id=CzkzEAAQBAJ&oi=fnd&pg=PT17&dq=where+use+desktop+app&ots=0gxLFD4KWB&sig=fhaCrOxeq2cvLd4bUE3JAx0vqBM#v=onepage&q=where%20use%20desktop%20app&f=false> (book do electru)
20. <https://ieeexplore.ieee.org/document/9463138> (Challenges in Developing Desktop Web Apps: a Study of Stack Overflow and GitHub)
21. Web Frameworks for Desktop Apps: an Exploratory Study (<https://gianlucascoccia.github.io/assets/pdf/ESEM2020.pdf>)
22. (<https://homepages.dcc.ufmg.br/~mtov/pub/2018-jss-github-stars.pdf>) (*What's in a GitHub Star? Understanding Repository Starring Practices in a Social Coding Platform*)
23. (https://www.researchgate.net/publication/314680454_Cross-Platform_Development_Frameworks_Overview_of_contemporary_technologies_and_methods_for_cross-platform_application_development) Cross-Platform Development Frameworks: Overview of contemporary technologies and methods for cross-platform application development.
24. <https://ieeexplore.ieee.org/document/7887704> (A Systematic Mapping Study of Software Development With GitHub)
25. <https://ieeexplore.ieee.org/document/7389272> (Bringing Desktop Applications to the Web)

26. <https://ieeexplore.ieee.org/document/6698918> (Got issues? Who cares about it? A large scale investigation of issue trackers from GitHub)
27. <https://ieeexplore.ieee.org/document/8947641> (A Cross-Repository Model for Predicting Popularity in GitHub)
28. <https://ieeexplore.ieee.org/document/8404391> (Detecting similar sentences using word embedding)
29. <https://ieeexplore.ieee.org/document/7371758> (Word Graph-Based Multi-sentence Compression: Re-ranking Candidates Using Frequent Words)
30. <https://ieeexplore.ieee.org/document/9152307> (Analysis of The Characteristics of Similar Words Computed by Word Embeddings)