

# Trabalho Interdisciplinar: Pesquisa em Engenharia de Software

## Professores:

- Felipe Augusto Lima Reis
- José Laerte Pires Xavier Junior

## Alunos:

- Guilherme Gabriel S. Pereira
- Henrique P. F. Monteiro
- Lucas Ângelo O. M. Rocha
- Victor Boaventura G. Campos
- Vinícius Marini C. e Oliveira

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



Pesquisa sobre aplicações desktop de  
código aberto em repositórios do Github

# Sumário

---

Apresentação

Glossário

1. Hipótese
2. Partes Interessadas
3. Objetivo
  - 3.1. Objetivo baseado na hipótese
  - 3.2. GQM (Goal, Question, Metric)
4. Perguntas da Pesquisa
  - 4.1. Pergunta 1
  - 4.2. Pergunta 2
  - 4.3. Pergunta 3
5. Metodologia
6. Textos relacionados
7. Proof of concept

Referência

Bibliografia



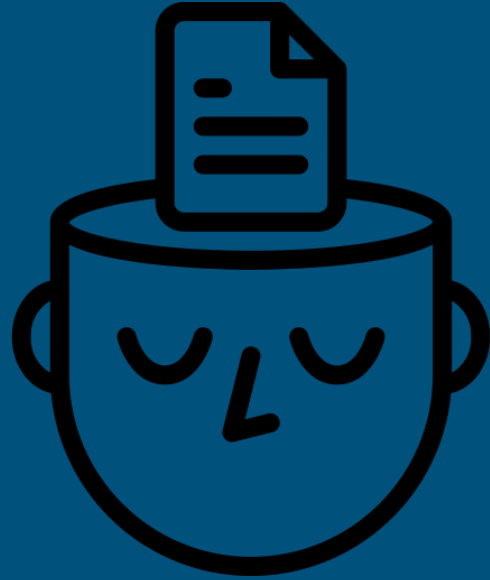
# Glossário

---

1. **Domínio:** refere-se ao propósito do software, a qual ramo ele se destina.  
Exemplo: domínio de aplicação de antivírus, editores de vídeos, etc.;
2. **Aplicações desktop:** programa que precisa estar instalado em sua máquina, independente de qual seja sua função.
3. **Repositórios desktop:** quaisquer repositórios que possuam dependências de aplicações desktop na linguagens Java, C# e JavaScript.



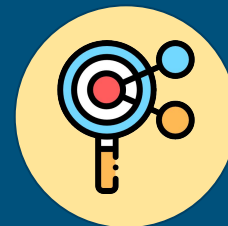
# 1 - Hipótese



# 1 - Hipótese

---

- 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, como antivírus;
- Nesse sentido, foi pensado que **ainda existem domínios populares no contexto de aplicações desktop.**



## 2 - Partes interessadas



## 2 - Partes interessadas

---

1. Desenvolvedores analisando tecnologias para projetos;
2. Analistas, clientes e gestores de novos projetos
  - a. que ainda não decidiram qual é o melhor para o seu domínio, web ou desktop.





### 3 - Objetivo



## 3.1 - Objetivo baseado na hipótese

---

- 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 descobrir domínios que ainda são populares para aplicações desktop.**



## 3.2 - 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.



## 4 - Perguntas e métricas



## 4.1 - Pergunta 1

---

**Para as aplicações desktop que ainda são mantidas, qual o domínio que elas se encontram atualmente?**

Métrica 1: Proporção de repositórios que utilizam tags ou descrições que são semelhantes versus que não utilizam tags ou descrições;

Métrica 2: Percentual da quantidade de repositórios desktop para cada domínio.



## 4.2 - Pergunta 2

---

**A quantidade de aplicações desktop vem diminuindo ao longo da última década?**

Métrica 1: Média de repositórios com dependências de aplicações desktop criados por ano para cada domínio;

Métrica 2: Proporção de repositórios com dependências de aplicações desktop versus sem dependências de aplicações desktop.



## 4.3 - Pergunta 3

---

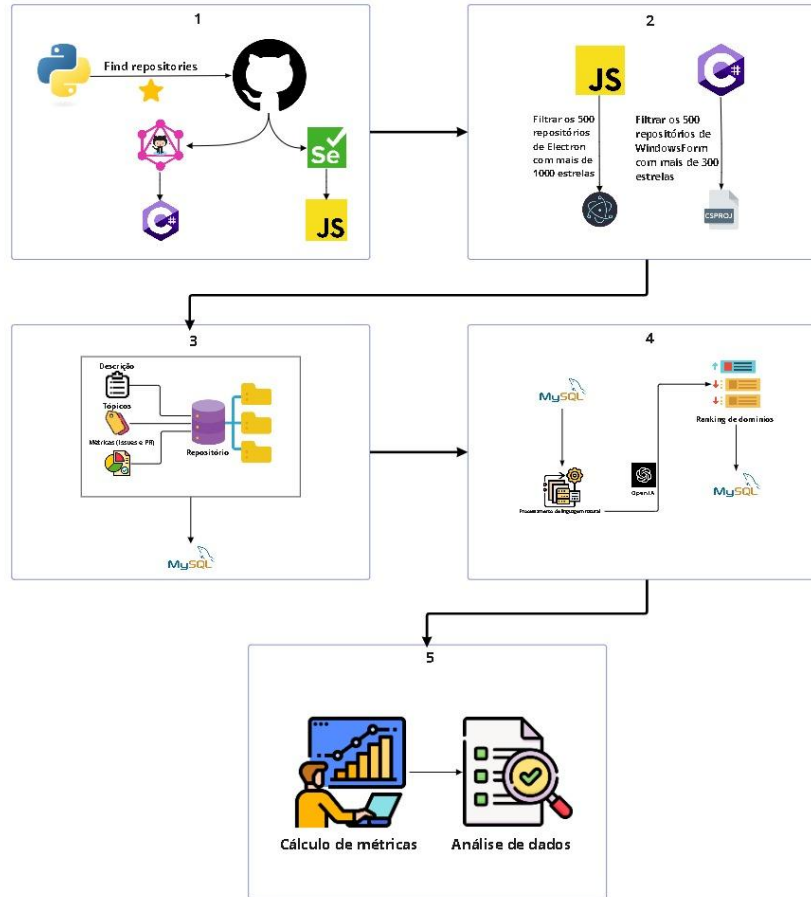
### **Aplicações desktop tem engajamento da comunidade?**

Métrica 1: Percentual de pull requests mergeados em relação aos não mergeados dos repositórios desktop por ano;

Métrica 2: Percentual de issues fechadas em relação a não fechadas em repositórios desktop por ano.



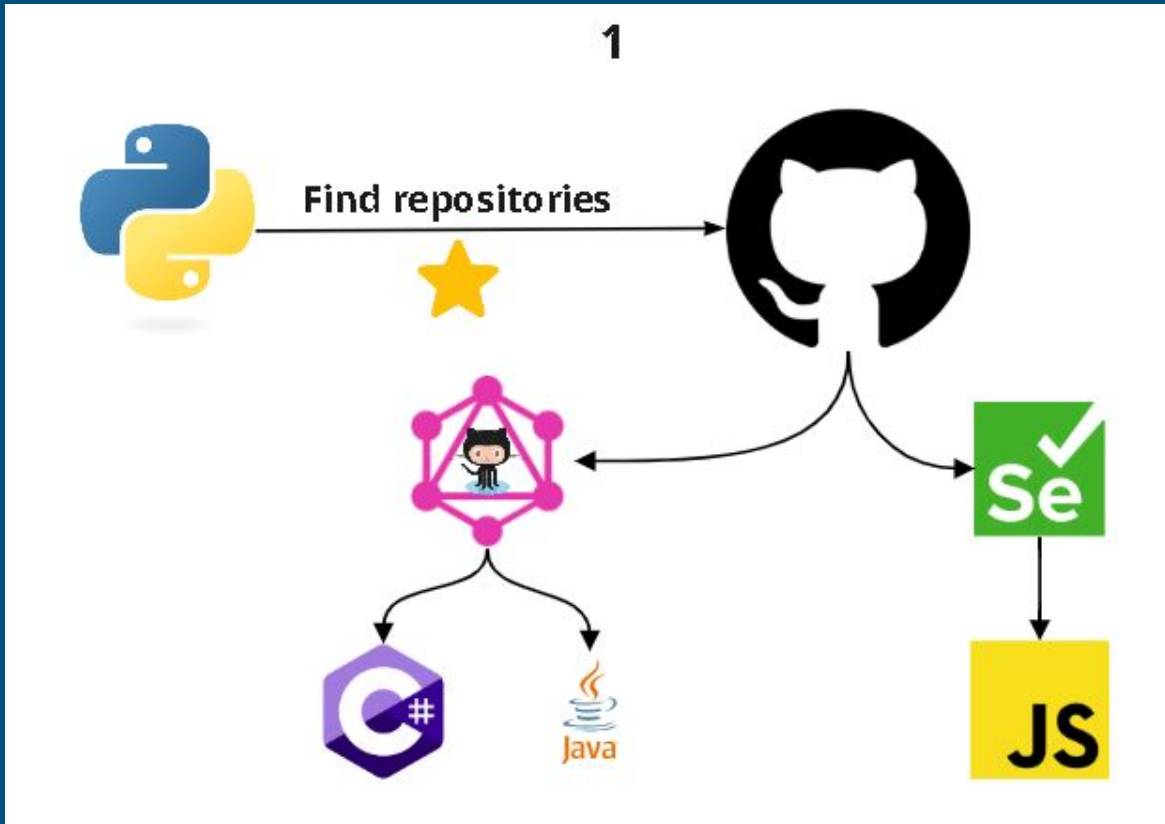
# 5 - Metodologia





## 5.1 - Busca de repositórios

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.



## 5.2 - Filtrando repositórios

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.

Dados e Métricas para: P2M1.

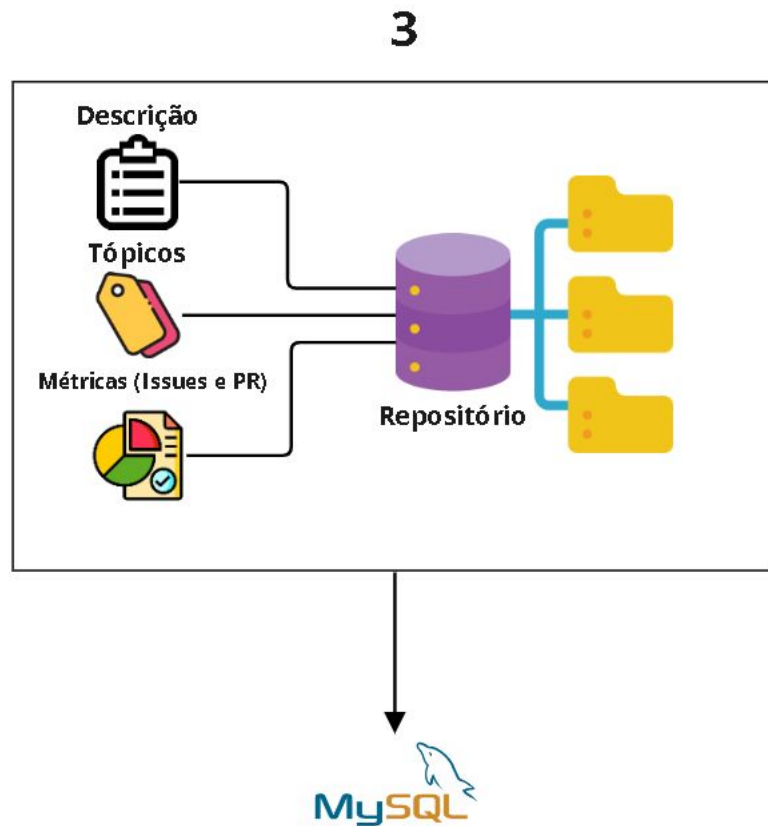
2



## 5.3 - Capturando dados

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.

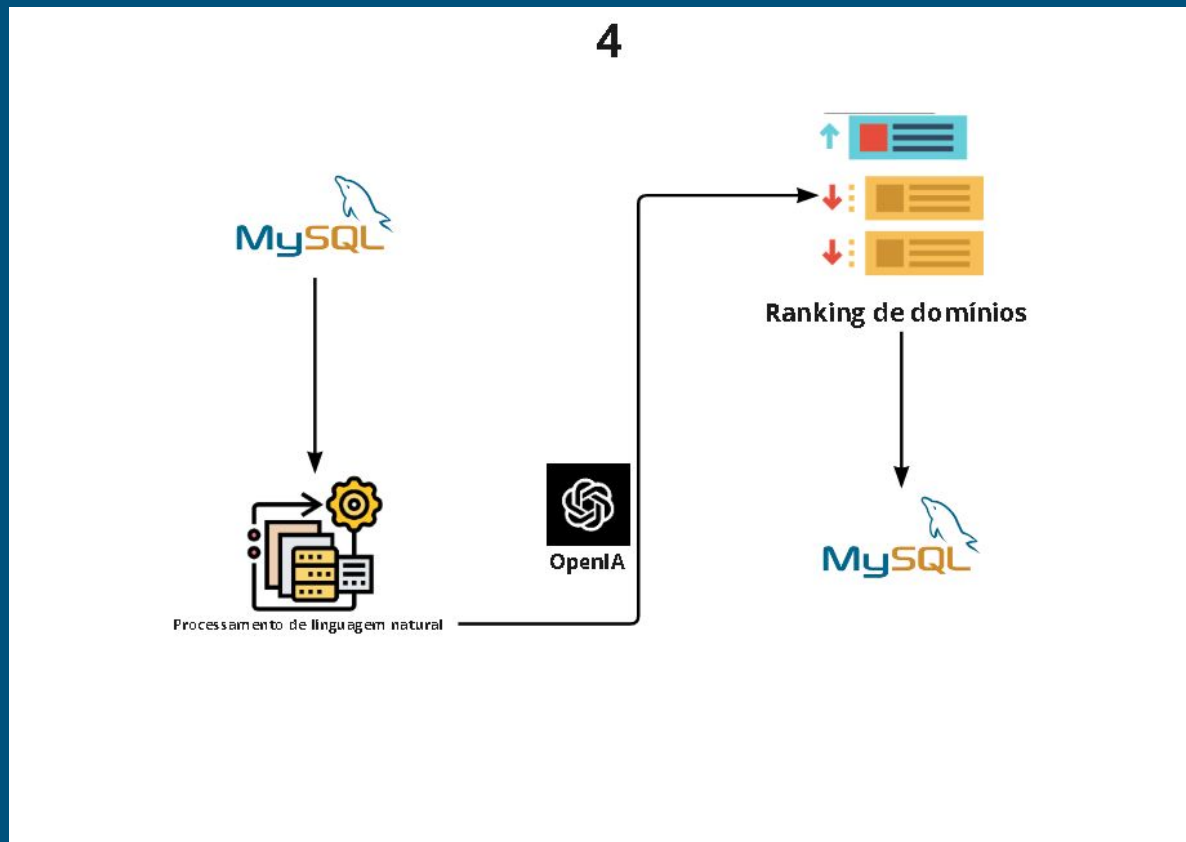
Dados e Métricas para: P1M1, P2M2, P3M1 e P3M2.



## 5.4 - Processando domínios

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.

Dados e Métricas para: P1M2.



## 5.5 - Análise de resultados

Analisa-se os resultados e calculadas as métricas para auxiliar a responder as perguntas levantadas e alcançar o objetivo definido nas fases iniciais do trabalho

5



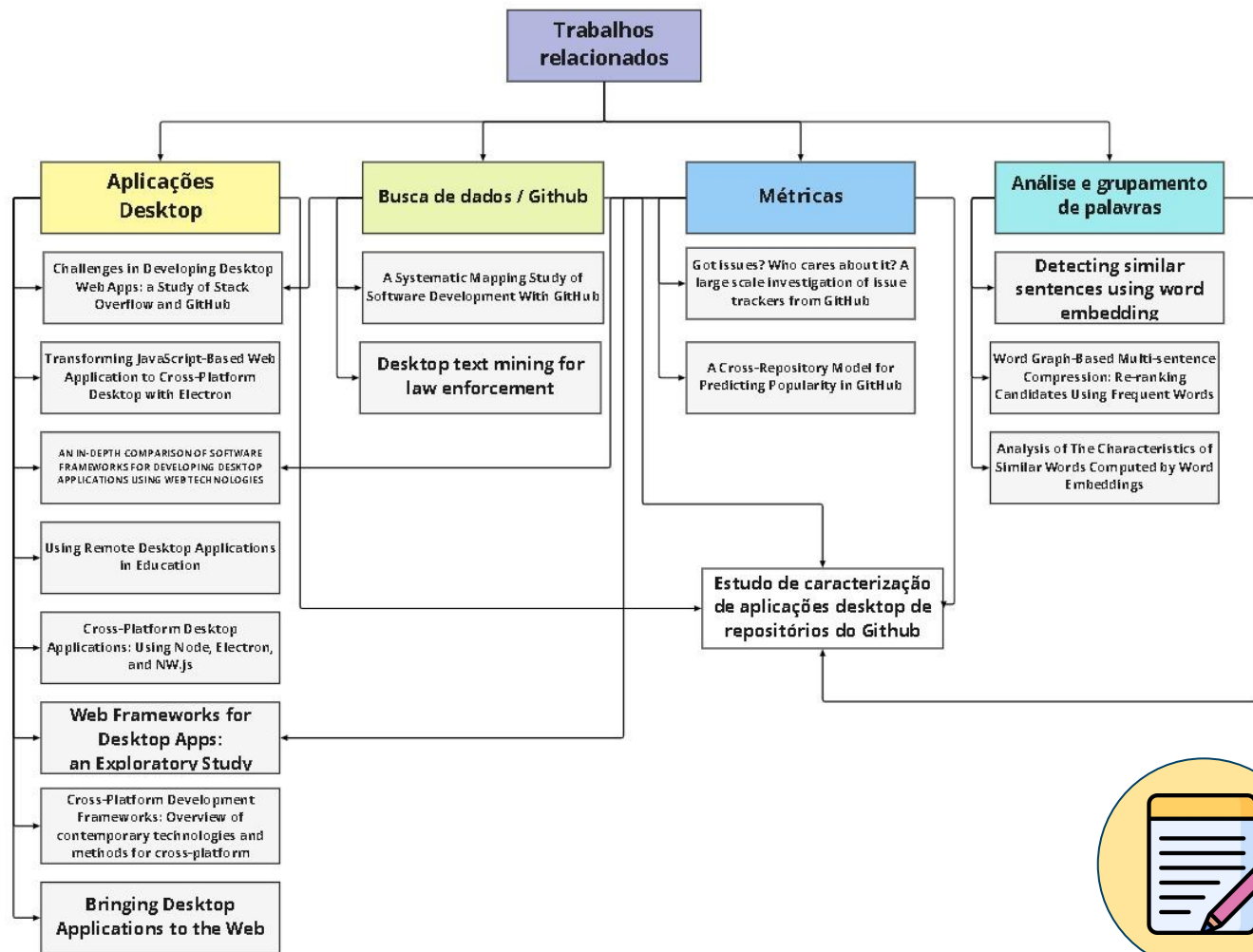
**Cálculo de métricas**



**Análise de dados**



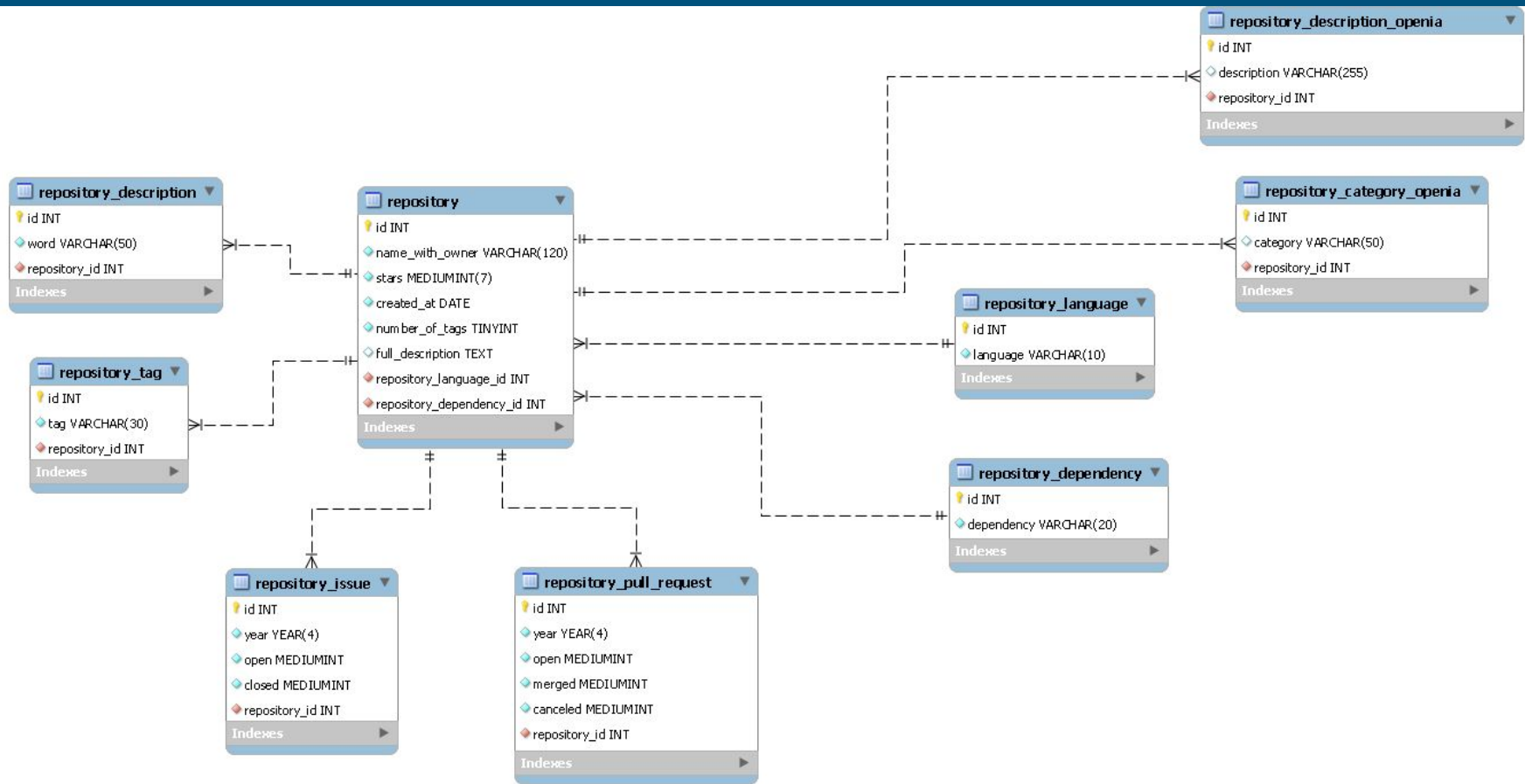
## 6 - Textos relacionados



## 5 - Proof of concept



# 5 - Proof of concept (Modelo do banco de dados)





# 5 - Proof of concept (Planilha da POC)

---

	A	B	C
1	LOWER(category)	contagem	válido
2	analysis	3	1
3	animation	3	1
4	artificial intelligence	2	1
5	authenticator	2	1
6	automation	6	1
7	backup	4	1
8	browser	14	1
9	browser extension	3	1
10	browsers	4	1
11	converter	2	1
12	cryptocurrency	13	1
13	cryptography	4	1
14	data analysis	4	1
15	data visualization	8	1
16	database	6	1
17	databases	7	1
18	deep learning	5	1
19	desktop automation	2	1
20	downloader	4	1
21	downloading	2	1
22	editor	20	1
23	encryption	7	1
24	extension	10	1
25	extensions	3	1
26	file conversion	2	1
27	file management	4	1
28	file sharing	3	1
29	firewall	3	1
30	framework	20	1
31	image processing	3	1
32	image viewer	2	1
33	installer	3	1
34	launcher	5	1
35	libraries	3	1

# 7 - Proof of concept (Domínios em planilha)

Domínios de aplicações descobertas pela OpenAI (utilizando GPT-3), de aplicações Electron e WindowsForms:



[https://docs.google.com/spreadsheets/d/18AUkPrT\\_lcXHZNaq\\_RolO4Y97ceDX6BHAixA5vG210s](https://docs.google.com/spreadsheets/d/18AUkPrT_lcXHZNaq_RolO4Y97ceDX6BHAixA5vG210s)



# Referência

---

- <https://xgen.com.br/blog/aplicacao-web-ou-desktop-qual-a-melhor-solucao>
- <https://wktechnology.com.br/aplicacoes-desktop-o-que-sao-e-como-funcionam/>
- <https://www.quora.com/Is-it-still-worth-to-build-desktop-applications>
- <https://www.devmedia.com.br/forum/aplicacoes-desktop/599445>
- <https://skymail.com.br/2022/07/05/e-possivel-manter-aplicacoes-desktop-em-nuvem/>
- <https://beta.openai.com/>



# Bibliografia

---

- PRESSMAN, Roger S.; MAXIM, Bruce R.. Engenharia de software: uma abordagem profissional. 8. ed. Porto Alegre: AMGH, 2016. ISBN 978-8580555349. E-book (Livro Eletrônico).
- SOMMERVILLE, Ian. Engenharia de software. 10. ed. São Paulo: Pearson, 2019. ISBN 978-8543024974. E-book (Livro Eletrônico).
- Basili, Victor; Gianluigi Caldiera; H. Dieter Rombach (1994). The Goal Question Metric Approach.



FIM

