

# Pesquisa Exploratória sobre a evolução da linguagem de programação Rust

André L. da Silva, Anna L. Gama, Edson J. Silva, Rodolfo R. Rodrigues

<sup>1</sup>Instituto de Informática e Ciências Exatas– Pontifícia Universidade de Minas Gerais (PUC MINAS)

andre.silva3357@gmail.com, annalaurard@hotmail.com , ejsengsoft@gmail.com

rodolfoforrodrigues14@gmail.com

**Abstract.** *Rust is a programming language created in 2010 by Mozilla and designed to be fast, secure and reliable. It has gained popularity in recent years due to its focus on security and performance, as well as its growing ecosystem of libraries and tools, steadily growing in the professional market. grow and stipulate whether it is a language worth studying and learning by a beginner or an experienced programmer who wants to enter the IT job market. To assess this evolution, the survey was divided into 3 research questions: The number of jobs on Rust is growing, the use of Rust is growing, and Rust has an active community. In order to answer these questions, a specific Dataset was collected for each one and, after filtering and analyzing each one, visualizations were generated that facilitated the understanding of the graphical result obtained.*

**Resumo.** *Rust é uma linguagem de programação criada em 2010 pela Mozilla e projetada para ser rápida, segura e confiável. Ela ganhou popularidade nos últimos anos devido ao seu foco em segurança e desempenho, bem como seu crescente ecossistema de bibliotecas e ferramentas, crescendo constantemente no mercado profissional. O objetivo geral deste estudo é analisar o crescimento dessa tecnologia, com o intuito de avaliar o seu crescimento e estipular se é uma linguagem que vale a pena ser estudada e aprendida por um programador iniciante ou já experiente, que deseja se introduzir no mercado de trabalho de TI. Para avaliar essa evolução, a pesquisa foi dividida em 3 questões de pesquisa: O número de vagas em Rust está crescendo , a utilização de Rust está crescendo e o Rust possui uma comunidade ativa. Com o intuito de responder a essas questões foram coletados Dataset específicos para cada uma e, após a filtragem e análise de cada um foi gerado visualizações gráficas que facilitaram a compreensão do resultado obtido.*

Palavras chave: Dataset , rust, linguagem de programação, api, conjunto de dados

## 1. Introdução

No mundo contemporâneo a tecnologia é uma área que vem constantemente crescendo e, cada vez mais, faz parte do cotidiano das pessoas, seja em ambientes de trabalho ou lazer. Segundo um estudo realizado pela C2Ti, estima-se que uma das profissões mais cotadas para o futuro seja a de desenvolvedores de software.

Paralelo a isso, um estudo de caso dirigido por Valter Moreno e Eduardo de Farias, busca analisar os desafios enfrentados por um profissional de TI. Nesse estudo, uma das

conclusões alcançadas é a de que a "tecnologia da informação é composta por diversas áreas de especialização e existem uma variedade de linguagens de programação que são empregadas no mercado". Esses fatores, acabam por dificultar o acesso de um indivíduo no mercado de TI, seja por desestímulo ou falta de conhecimento sobre qual melhor área ou linguagem de programação para começar a estudar.

Com base nisso e, sabendo que existem diversas linguagens de programação, este estudo visa aprofundar na análise de uma linguagem de programação específica, o Rust, com o intuito de determinar se é uma tecnologia que possui uma tendência de crescimento e seria indicada para um profissional que estaria buscando sua primeira linguagem de programação para estudar.

A motivação envolvida nessa escolha de tecnologia para realizar esta pesquisa, esta relacionada com a sua crescente popularidade nos últimos anos. O Rust surgiu devido ao seu foco em segurança e desempenho, além do fato de sua criação ter como foco a rapidez, segurança e confiabilidade. Tendo sido criada em 2010 pela Mozilla, o Rust é uma linguagem de código aberto, semelhante ao C++, mas que incorpora alguns conceitos de programação moderna, como programação funcional e inferência de tipos.

O objetivo geral deste estudo é o de analisar a evolução da linguagem de programação Rust, nos últimos 5 (cinco) anos. Afim de determinar se seria uma boa opção de tecnologia para um desenvolvedor iniciante ou experiente começar a aprender e utilizar.

Para atingir o objetivo geral proposto, foram definidos os seguintes objetivos específicos: i) Analisar os repositórios que são criados e mantidos no Github, que tenham como principal linguagem de programação o Rust. ii) Analisar a quantidade de vagas que são ofertadas no LinkedIn, que tenham como requisito o conhecimento da linguagem de programação o Rust. iii) Analisar a quantidade de perguntas e respostas relacionadas à linguagem de programação Rust, na plataforma Stack Overflow.

A análise do crescimento da popularidade e uso da linguagem Rust pode fornecer informações sobre as tendências e preferências do mercado profissional de programação, fazendo com que possamos entender melhor se é uma tecnologia que de fato surgiu para substituir algumas que já estão consolidadas no cenário da programação.

Esse artigo está subdividido em 5 seções: Introdução (1), Referencial Teórico (2), Metodologia (3), Resultados (4), Discussão (5) e Referências Bibliográficas (6).

## **2. Referencial Teórico**

Nesta seção serão apresentados bibliografias que fizeram parte da composição deste estudo. Estes descrevem em ordem, a escrita de artigos descritivos, introdução a linguagem Rust, Desafios da utilização desta tecnologia e

Sobre a escrita de artigos descritivos, o estudo de [Kanoksilapatham 1992] descreve a estrutura em que um artigo de pesquisa na área de engenharia deve se encontrar e que se mostra de forma eficaz. Como base do estudo foram utilizados três conjuntos de dados de introdução de artigo, em três áreas diferentes da engenharia (a de software inclusa), logo após foi feita uma análise que identifica os padrões textuais. A estrutura deste artigo, leva em consideração os resultados do estudo.

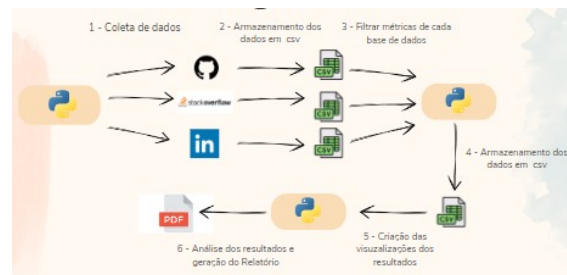
O estudo introdutório de [Dmitriy 2021] trás consigo a proposta de convencer a utilização de Rust. Como ser humano é propenso ao erro é necessário uma linguagem que traga segurança ao programar e que mostre que o desenvolvimento está correto antes de passar pelos testes. Também é mostrado outros benefícios como segurança e rapidez. Este foi um estudo que deu base no que as pessoas já conhecem sobre a linguagem e o que deveria ser feito a seguir.

Utilizando o estudo caso [Shuofei 2019], se refere a popularidade crescente e iminente de Rust e como esta vem sendo utilizada, desta forma é necessário conhecer os desafios relacionadas a linguagem. Para isso foi utilizado 100 perguntas do stackOverflow e foi feita uma pesquisa online com 101 programadores. Esse artigo foi utilizado para que este estudo conseguisse se guiar na hora de formar as perguntas utilizadas neste.

Ainda sobre os desafio da linguagem [Fulton 2021] utiliza feeedbacks da comunidade que utiliza a ferramenta para construir um estudo sobre os benefícios desta. Novamente foi apresentado o quadro de desempenho e segurança. Porém a curva de aprendizado da linguagem vira um impedimento. Este artigo foi utilizado para a formação da pergunta três, que serve de indicativo se a comunidade atua na manutenção destas dúvidas.

Por fim o estudo de [Yamamoto 2020] indica sobre a mineiração de dados no Github, que costuma ser a plataforma em que mais acontece a extração de dados. Neste estudo mostra quais são as melhores métricas de um repositório para indicar se este é adequado para a coleta de dados. Este artigo foi utilizado para guiar este estudo no momento de coleta de dados para estudo.

### 3. Metodologia



**Figure 1. Metodologia**

A figura 1 apresenta a metodologia utilizada por este estudo, que visa obter dados relevantes relacionadas a linguagem de programação Rust a partir de três diferentes âmbitos, que são: repositórios criados e mantidos, ofertas de vagas de emprego e interação da comunidade com a linguagem, por meio da análise de perguntas e respostas que surgem sobre ela.

Primeiramente, foi feita a coleta de dados das vagas empregatícias em Rust criadas nas últimas horas no LinkedIn, no decorrer dos últimos 15 (quinze) dias. Para fins de comparação, também foi coletado dados referentes a vagas de emprego relacionadas a outras linguagens de programação, nos mesmo período de tempo.

Na segunda etapa da pesquisa, foram obtidos dois conjunto de dados separados: o primeiro referente ao número total de repositórios do Github em Rust criados nos últimos

5 anos e o segundo relativo aos 1000 (mil) repositórios do Github em Rust mais populares que foram criados nos anos de 2018 ate 2022.

Posteriormente a isso,foi feita a análise da atividade da comunidade Rust no Stack-Overflow. Para tal fim, foram obtidas todas as perguntas realizadas dentre os anos de 2019 a 2023, onde foi criado um script que utilizava a API da plataforma para obter todas as informações e salva lás em um arquivo JSON. Em seguida todas as perguntas coletadas foram separadas conforme a tag que especifica qual tipo de tecnologia é referente a pergunta, sendo elas: Rust, Python, GO, javascript, Java e outras linguagens.

Tendo o conjunto de dados (Dataset) inicial definido, foram feitas análises específicas para cada Dataset obtido, com o objetivo de extrair as métricas definidas em cada questão de pesquisa. A abordagem utilizada em cada análise será explicada na próxima seção.

Por fim, com a posse dos dados já filtrados e analisados, foi utilizado o Power BI, uma ferramenta de análise e criação de gráficos, para gerar as visualizações dos resultados encontrados para cada questão de pesquisa estipulada, que serão abordadas na seção de Resultados (IV).

### **3.1. Hipóteses**

Nesta seção do artigo serão apresentadas as perguntas feitas, junto com suas métricas e as hipóteses feitas para cada. As perguntas foram feitas para guiar o rumo deste artigo, as métricas são formas encontradas para respondê-las, estas fornecem uma base para a tomada de decisões. As hipóteses são formulações temporárias usadas para responder um possível fenômeno ou problema.

#### **3.1.1. Q1 - O número de vagas em Rust está crescendo ?**

Levando em consideração o mercado de trabalho e o quão este está afunilado, uma das perguntas utilizadas leva em consideração o número de vagas em Rust. Esta é uma das principais questões que os programadores devem levar em consideração antes de decidirem qual linguagem de programação irão dedicar o seu tempo para aprender, principalmente se forem desenvolvedores iniciantes.

Foram definidas três métricas para responder essa pergunta, sendo elas: M1) Quantidade de vagas de Rust em comparação com linguagens lançadas no mesmo período, assim é apresentado se a linguagem cresce dentro do esperado ou até mais ; M2) Quantidade de vagas de Rust no mercado; e por fim, a M3) Quantidade de vagas de Rust em comparação com as linguagens mais antigas, que correlaciona Rust com as já existentes no mercado, que estão consolidadas.

Sendo assim, a hipótese inicial é de que o número de vagas em Rust está crescendo, levando em consideração a alta demanda para profissionais em projetos da área. Além do desempenho e segurança que essa linguagem proporciona em comparação com outras, como C e C++.

### **3.1.2. Q2 - A utilização de Rust está crescendo?**

Outra questão que se é levado em consideração ao definir uma linguagem nova para se aprender é se a utilização desta está em crescimento, pois é mais fácil procurar projetos de referência e até mesmo aprender coisas novas. Pensando nisso foi levantada a segunda pergunta.

Para respondê-la foi usado as seguintes métricas: M4) Número de repositórios do Github, com a linguagem Rust, que são criados por ano, levando em consideração os novos repositórios criados nos últimos cinco anos. M5) Número de contribuidores ativos em repositórios do Github, com a linguagem Rust, nesta métrica, para cada ano analisado foi pego a quantidade de contribuidores que commitaram em um determinado repositório analisado para aquele período. Por fim a M6) Número de issues em repositórios com a linguagem Rust, na qual foi feita a análise da quantidade de issues que um repositório analisado teve em determinado período de tempo avaliado.

Nesse sentido, hipótese inicial para esta pergunta é que a utilização dessa linguagem está crescendo devido à características de segurança, desempenho e eficiência. Tais coisas atraem desenvolvedores curiosos com os benefícios da utilização e dos benefícios proporcionados pela linguagem e aumentam a quantidade de projetos criados em Rust.

### **3.1.3. Q3 - O Rust possui uma comunidade ativa?**

Ao desenvolver em novas linguagens surgem algumas dúvidas e outras incertezas, para que estes problemas sejam sanados é comum recorrer a comunidade desta. Quando se tem uma comunidade ativa, podem se existir perguntas já feitas anteriormente que tirem as dúvidas de um novo desenvolvedor ou se é feita uma nova pergunta está será respondida rapidamente, não levando ao desânimo na aprendizagem. Desta forma, a última pergunta formulada leva isso em consideração.

Foram utilizadas as seguintes métricas. M7) Percentual do número total de perguntas de Rust no Stack Overflow, foi feito o armazenamento total de perguntas em Rust; M8) Frequência em que as perguntas de Rust são criadas no Stack Overflow, aqui foi coletado o z-score das perguntas criadas em Rust, em comparação às outras linguagens mais populares. Por fim a, a M9) Porcentagem no qual as perguntas de Rust são respondidas no Stack Overflow em relação a perguntas referentes a outras linguagens, que mostra o quão participava é essa comunidade.

Desta forma a hipótese inicial para essa pergunta é que a comunidade de Rust ainda é pequena em comparação às outras linguagens de programação mais consolidadas, mas está em crescimento assim como a linguagem.

## **3.2. Análise de resultados**

Os dados obtidos através da mineração do stackOveflow, Linkedin e Github passaram por um tratamento e foram armazenados em arquivos do tipo csv. Estes foram exportados para a ferramenta power BI que gerou os dashboards para o estudo e do qual serão utilizados os gráficos deste arquivo.

## 4. Resultados

Nesta seção serão mostrados os resultados obtidos após realizar todos os processos listados nas seções acima, estes que irão responder as questões feitas(Q1, Q2, Q3). Para estas foram usadas as métricas descritas(M1, M2, M3, M4, M5, M6, M7, M8, M9), que irão ser utilizadas no cálculo dos resultados.

### 4.1. O número de vagas em Rust está crescendo ?

Levando em consideração a M1 que foi utilizada para responder a primeira pergunta, foi encontrado os resultados apresentados na figura 2. foram coletados cerca de duas semana de dados de vagas no LinkedIn, dia após dia, pegando as últimas 24 horas. Destes a vagas não possuem um crescimento grande, porém tem estabilidade e não está caindo. Se for feita uma projeção de acordo com os dados obtidos, é possível que tenha um crescimento significativo no mundo a partir dos anos.

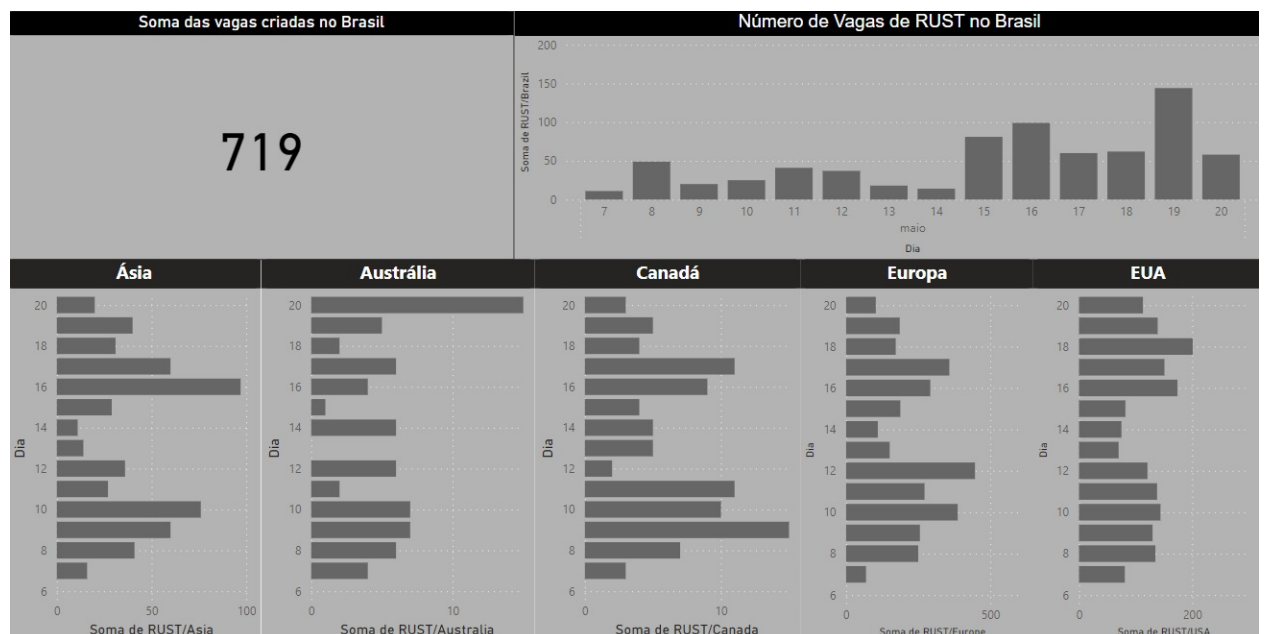


Figure 2. Vagas em Rust

Para a M2, neste mesmo período de data foram coletadas as vagas nas outras linguagens que foram criadas no mesmo período de data do Rust e feito uma mediana. Como se pode analisar na figura 3, a mediana dele é menor que a das outras linguagens, ficando na frente apenas de Dart e tende para o quartil superior. Logo, em comparação com as outras linguagens criadas na mesma época, ele fica entre uma das últimas linguagens com vagas criadas, o que é um cenário desafiador.

Por fim com a M3, foram coletadas as vagas em Rust em comparação com as cinco linguagens mais populares. Ao analisar a figura 4, se pode ver que Rust tem uma mediana muito inferior com a destas linguagens mais consolidadas no mercado, possuindo um nível ínfimo de vagas criadas.

### 4.2. A utilização de Rust está crescendo?

Levando em consideração a M4 que foi utilizada para responder esta pergunta, os resultados encontrados, como se pode analisar na figura 5 é que a utilização de RUST está em

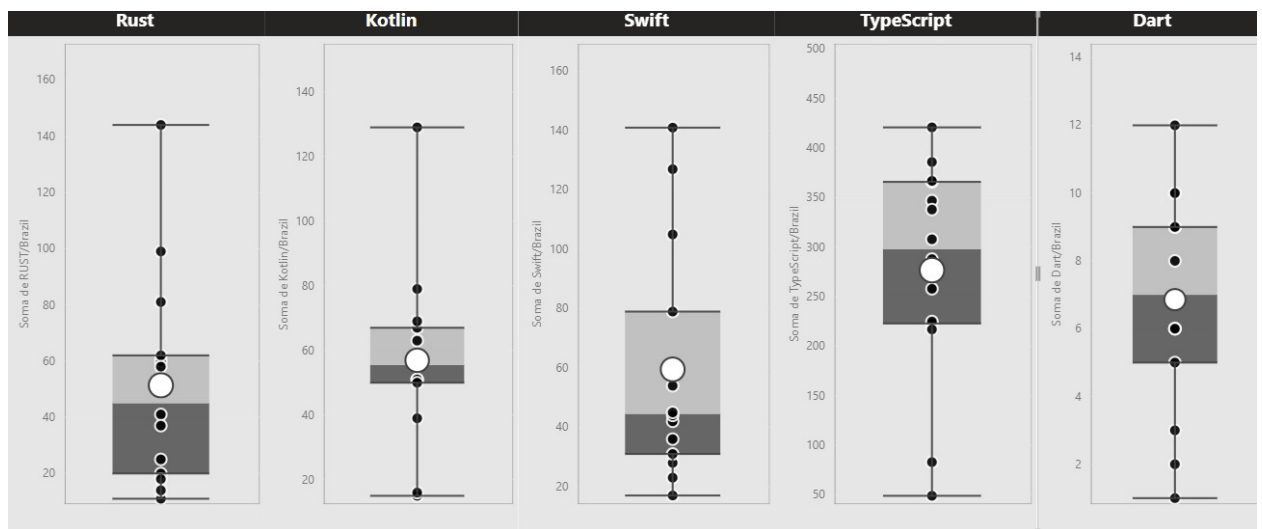


Figure 3. Rust X Linguagens mais recentes

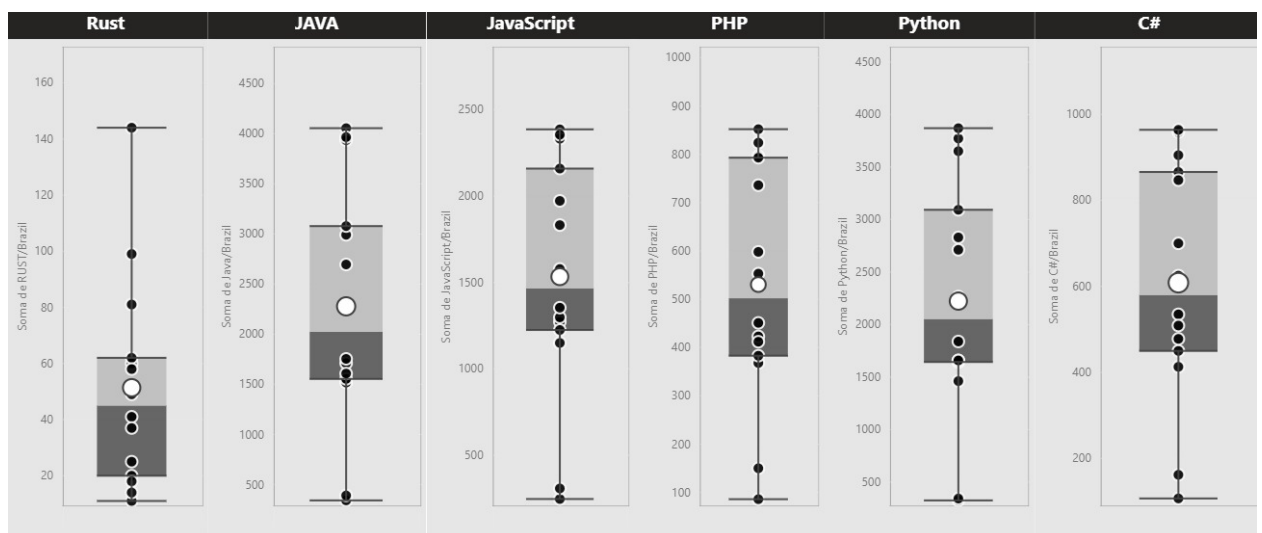
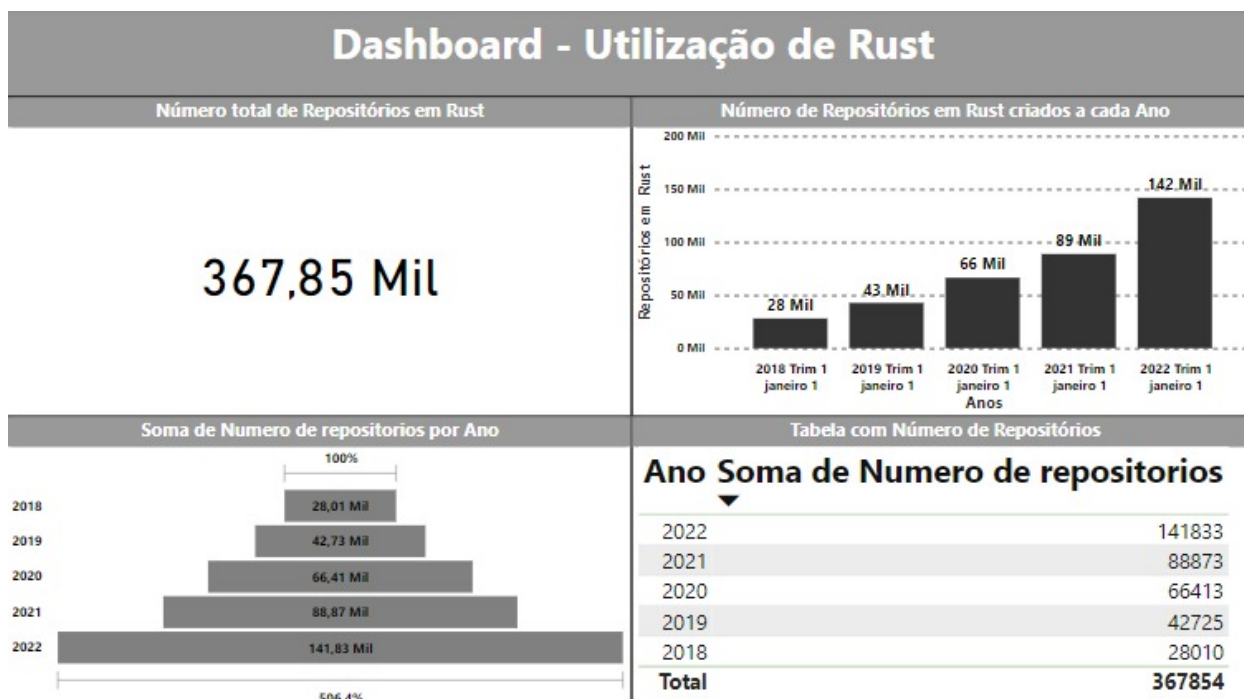


Figure 4. Rust x Linguagens mais antigas

alta. O número de repositórios vai de 29 mil em 2018, para 142 mil em 2021, que é um crescimento excelente para cinco anos de análise da linguagem.

Em relação a M5 pode se analisar que o número de contribuidores em relação ao anos de análise teve aumento, mas um aumento pequeno em relação ao últimos anos e a mediana é continua, ou seja, como dito antes o crescimento de contribuidores de fato não é significativo.

Por fim, na métrica M6 é possível examinar a quantidade de issues abertas por ano e se pode perceber que esse valor vem na verdade diminuindo ao passar dos anos. Este fator pode ter relação tanto ao desinteresse de continuar novos projetos na linguagem ou a baixa manutenção destes. De fato é uma queda preocupante e que pode desanimar novos desenvolvedores.



**Figure 5. Utilização de Rust nos últimos anos**

### 4.3. O Rust possui uma comunidade ativa?

Para a M7 foram coletados os dados de todas as perguntas feitas no stackOverflow nos últimos quatro anos e separados de acordo com o tópico da pergunta. Como visualizado nas figuras 8, 9, 10 e 11, as perguntas feitas em Rust são minoria, porém ao comparar somente com as que são direcionadas a uma linguagem, é perceptível certo crescimento e que ela não está tão longe das outras.

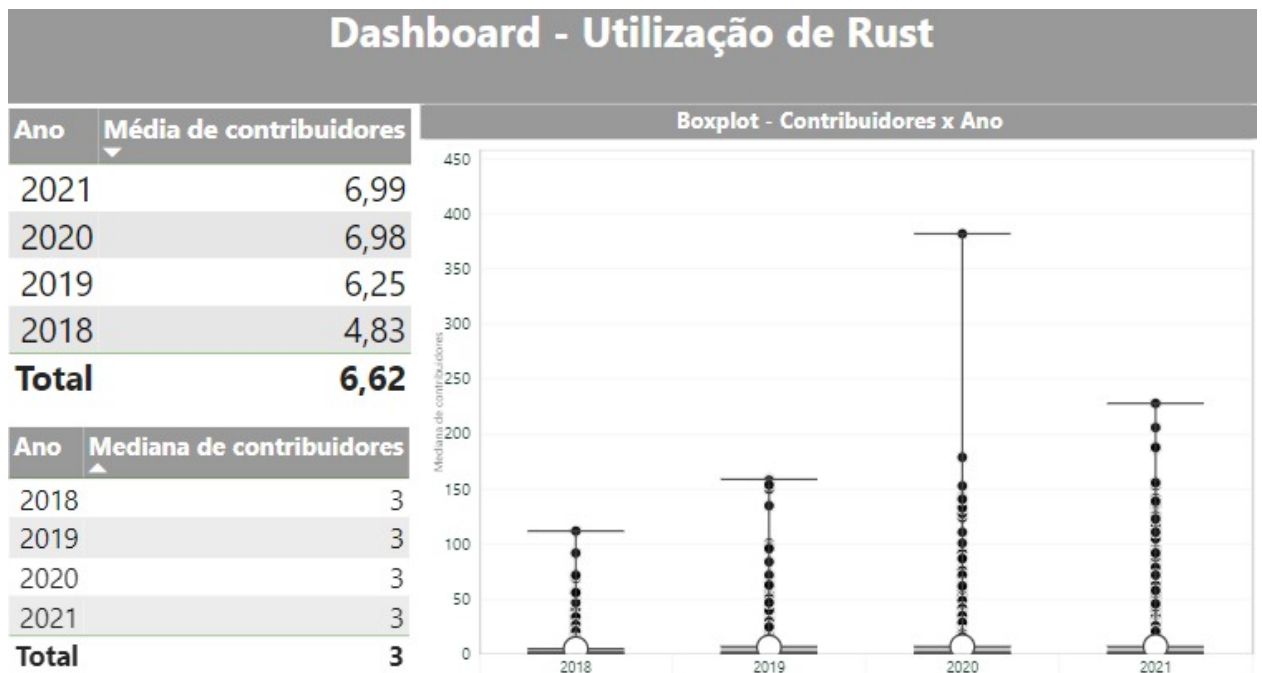
Para a M8 foi coletada a frequência em que essas perguntas foram criadas nos últimos quatro anos em comparação com outras quatro linguagens que possuem mais fama. Como se pode analisar nos gráficos das imagens 12, 13, 14 e 15 a frequência de criação das perguntas é grande e está crescendo, tendo um crescimento alto em 2020 e em 2022, até mesmo ultrapassou as linguagens mais famosas.

Por fim, para a M9 foi coletado o percentual de resposta das perguntas feitas para cada linguagem, utilizando as mesmas quatro linguagens da M8 e os últimos 4 anos. Como analisado nos gráficos das imagens 16, 17, 18 e 19, as respostas as perguntas feitas para a linguagem Rust é inclusive maior que as outras linguagens mais populares, porém tem uma tendência de queda assim como as outras.

## 5. Ameaças a validade

Ao decorrer da coleta de dados e ao escolher as ferramentas que seriam utilizadas, foi encontrada duas ameaças à validade. A primeira diz razão a quantidade de dados coletados em relação às vagas. A API do LinkedIn, que não fosse de forma paga não trazia dados relevantes e não era possível de se fazer muitas requisições. Para contornar isso foi coletado dia a dia as novas vagas criadas e para que se obtesse uma base maior de dados, foram coletados dados de outros países para análise.





**Figure 6. Contribuidores por ano em repositórios RUST**

Outra ameaça encontrada foi em relação aos contribuidores em cada repositório, para visualizar o crescimento não se poderia contabilizar os mesmos contribuidores para os mesmos repositórios em anos diferentes. Para contornar esta ameaça foi feito no script de coleta um limite em que não se contabilizasse contribuidores duplicados.

## 6. Discussão

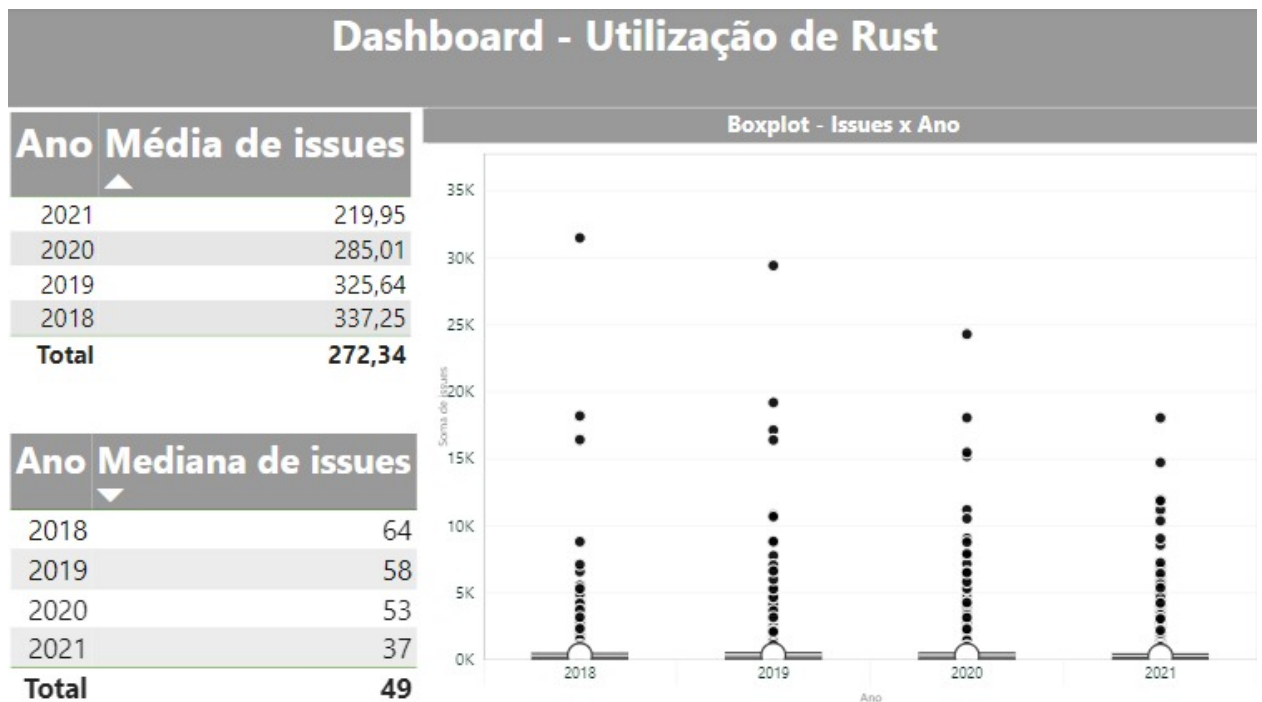
Como mostrado nos resultados obtidos nas seções anteriores, o Rust tem um índice de crescimento estável e uma comunidade ativa. Por outro lado a quantidade de vagas não mostra um crescimento e levando em consideração as linguagens já consolidadas e até mesmo as criadas na mesma época que Rust os números de vagas da tecnologia em estudo são muito baixas.

Em outras palavras, Rust está em crescimento e tem um potencial evolutivo muito grande, porém o mercado de trabalho ainda está relativamente fechado para a linguagem. Este último ponto pode desanimar no aprendizado da tecnologia, porém com o crescimento da comunidade e projetos criados, no fim é um bom investimento a se fazer visando alguns anos.

## References

Kaito Kaide, Haruaki Tamada, "Argo: ferramenta de busca e visualização de dados de séries temporais para GitHub", 2022 23<sup>a</sup> ACIS International Summer Virtual Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD-Summer) , pp .141-147, 2022.

K. Yamamoto, M. Kondo, K. Nishiura e O. Mizuno, "Quais métricas devem os pesquisadores usar para coletar repositórios: um estudo empírico," 2020 IEEE 20<sup>a</sup> Conferência Internacional sobre Qualidade, Confiabilidade e Segurança de Software (QRS) , Macau, China , 2020, pp. 458-466, doi: 10.1109/QRS51102.2020.00065.



**Figure 7. Issues por ano em RUST**

S. Zhu, Z. Zhang, B. Qin, A. Xiong and L. Song, "Learning and Programming Challenges of Rust: A Mixed-Methods Study," 2022 IEEE/ACM 44th International Conference on Software Engineering (ICSE), Pittsburgh, PA, USA, 2022, pp. 1269-1281, doi: 10.1145/3510003.3510164.

M. Ling, Y. Yu, H. Wu, Y. Wang, J. R. Cordy and A. E. Hassan, "In Rust We Trust – A Transpiler from Unsafe C to Safer Rust," 2022 IEEE/ACM 44th International Conference on Software Engineering: Companion Proceedings (ICSE-Companion), Pittsburgh, PA, USA, 2022, pp. 354-355, doi: 10.1145/3510454.3528640.

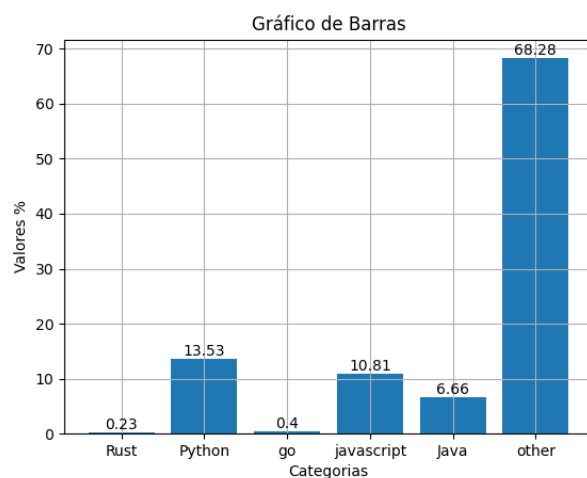
Dmitriy Kashitsyn, "Why Rust?," 2018/ Published in Parity Technologies

A. Tomb, S. Pernsteiner and M. Dodds, "Symbolic Testing for C and Rust," 2020 IEEE Secure Development (SecDev), Atlanta, GA, USA, 2020, pp. 33-33, doi: 10.1109/SecDev45635.2020.00021.

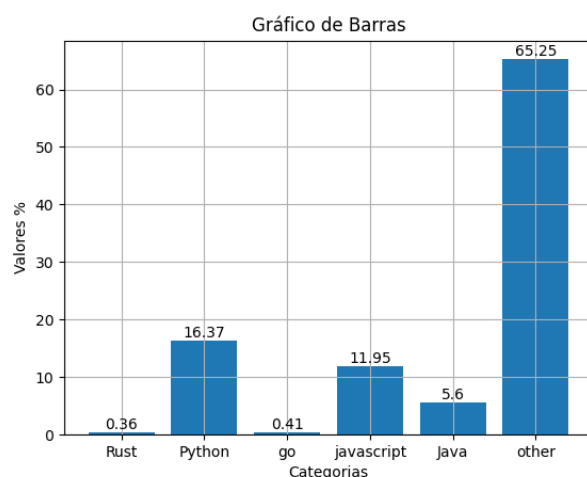
Kelsey R. Fulton and Anna Chan, University of Maryland; Daniel Votipka, Tufts University; Michael Hicks and Michelle L. Mazurek, University of Maryland, "Benefits and Drawbacks of Adopting a Secure Programming Language: Rust as a Case Study", 2021 August

B. Kanoksilapatham, "Research Article Structure of Research Article Introductions in Three Engineering Subdisciplines," in IEEE Transactions on Professional Communication, vol. 55, no. 4, pp. 294-309, Dec. 2012, doi: 10.1109/TPC.2012.2223252.

Gerald Rau, "Rediscovery of Developmental Research Articles in Electrical Engineering and Description of Their Macrostructure", IEEE Transactions on Professional Communication, vol.64, no.4, pp.427-443, 2021.



**Figure 8. Perguntas Totais em 2019**



**Figure 9. Perguntas Totais em 2020**

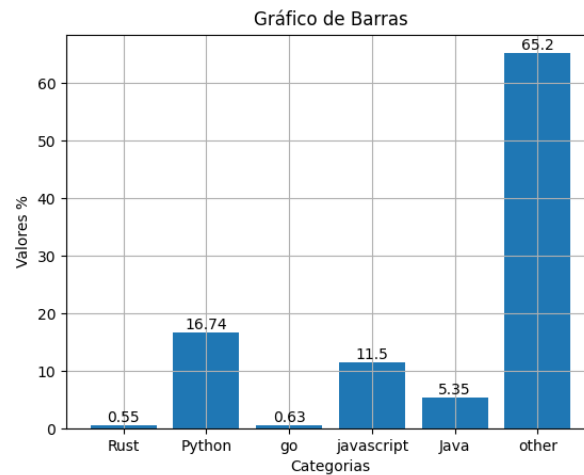
Guangsa Jin, Chenle Li, Ya Sun, "Exploring the Macrostructure of Research Articles in Economics", IEEE Transactions on Professional Communication, vol.63, no.3, pp.227-243, 2020.

Li Lian Khaw, Wei Wu Tan, "Creating Contexts in Engineering Research Writing Using a Problem-Solution-Based Writing Model: Experience of Ph.D. Students", IEEE Transactions on Professional Communication, vol.63, no.2, pp.155-171, 2020.

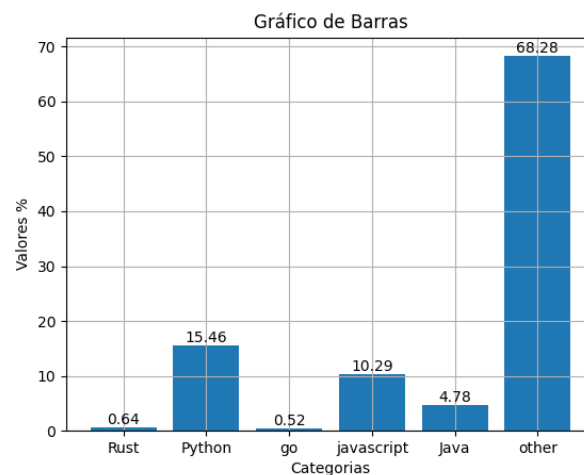
Danilo Y. Fukuda, "Uma avaliação comparativa das linguagens Rust e C para o ensino de programação", Brasília 2021.

Xiuwen Lu, Zhicong Zhang, Hui Xu, "Safe Rust Code Recommendation Based on Siamese Graph Neural Network", 2022 IEEE 27th Pacific Rim International Symposium on Dependable Computing (PRDC), pp.1-11, 2022.

Catalin Cimpanu. "Microsoft: 70 percent of all security bugs are memory safety issues." <https://www.zdnet.com/article/microsoft-70-percent-of-all-security-bugs->



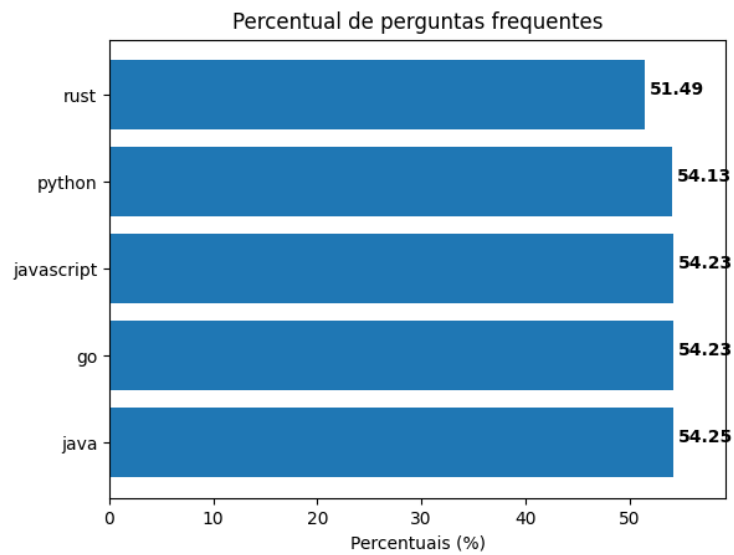
**Figure 10. Perguntas Totais em 2021**



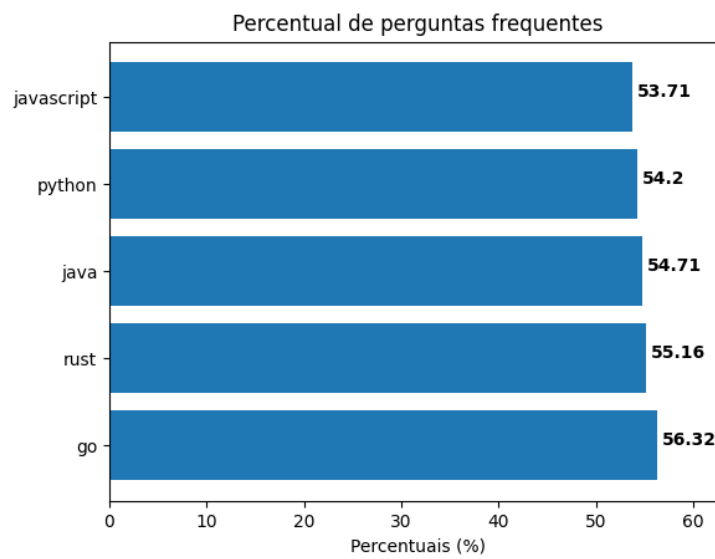
**Figure 11. Perguntas Totais em 2022**

are-memorysafety-issues/, 2019.

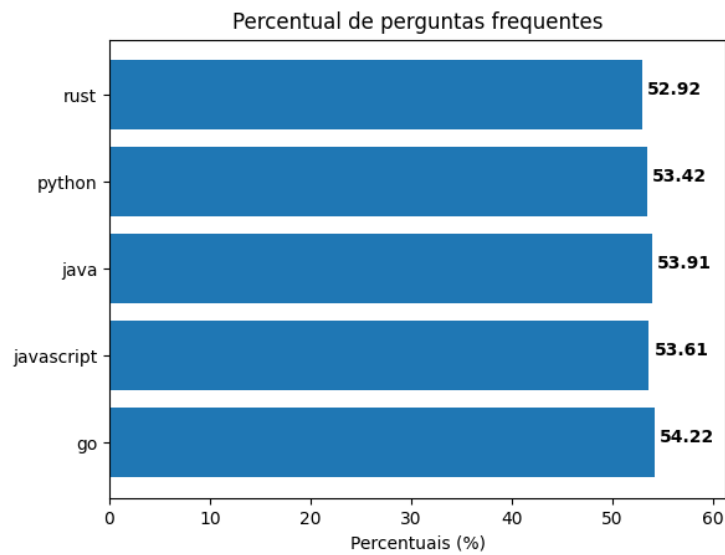
Yung-Yu Chang, Pavol Zavorsky, Ron Ruhl, and Dale Lindskog. Trend analysis of the cve for software vulnerability management. In Proceedings of the 2011 IEEE Third International Conference on Privacy, Security, Risk and Trust and 2011 IEEE Third International Conference on Social Computing, pages 1290– 1293. IEEE, 2011.



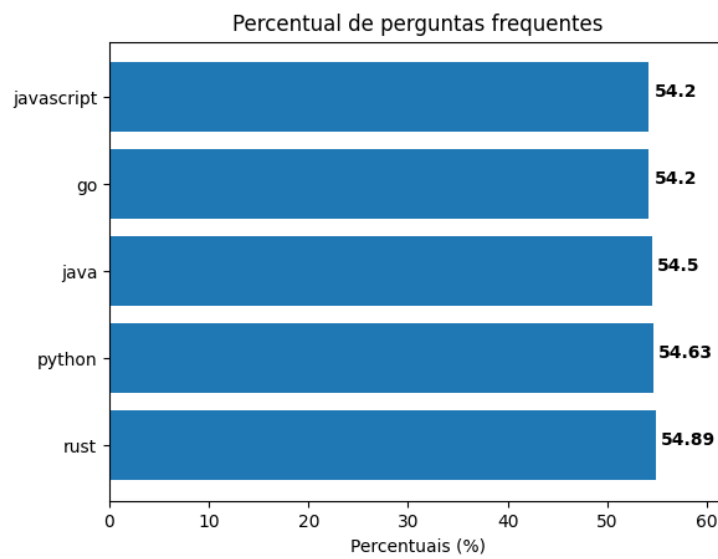
**Figure 12. Frequência de criação de perguntas 2019**



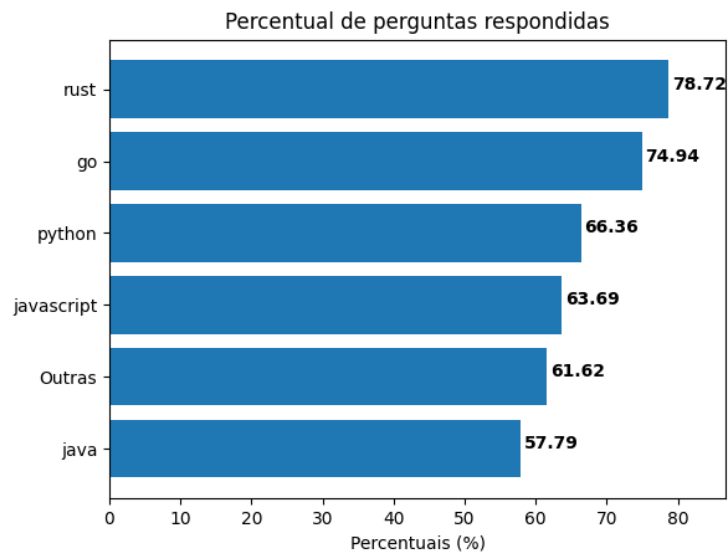
**Figure 13. Frequência de criação de perguntas 2020**



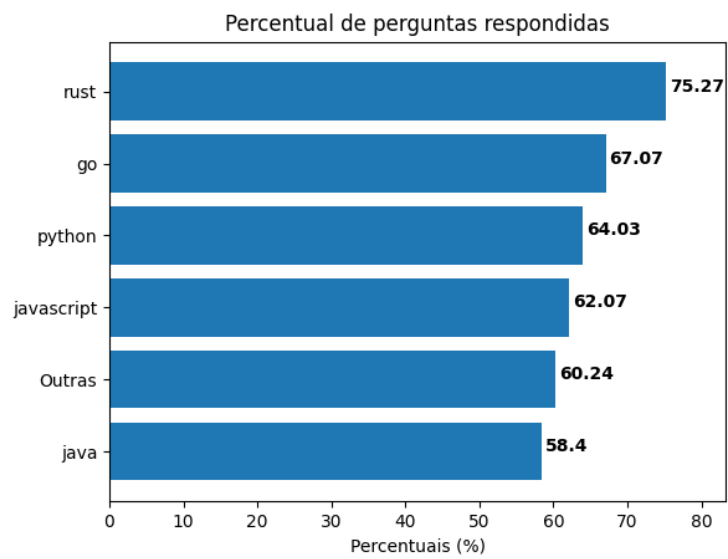
**Figure 14. Frequência de criação de perguntas 2021**



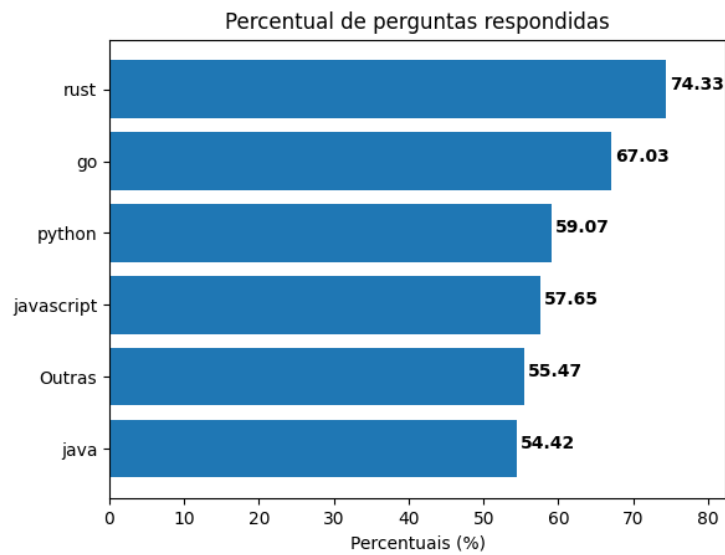
**Figure 15. Frequência de criação de perguntas 2022**



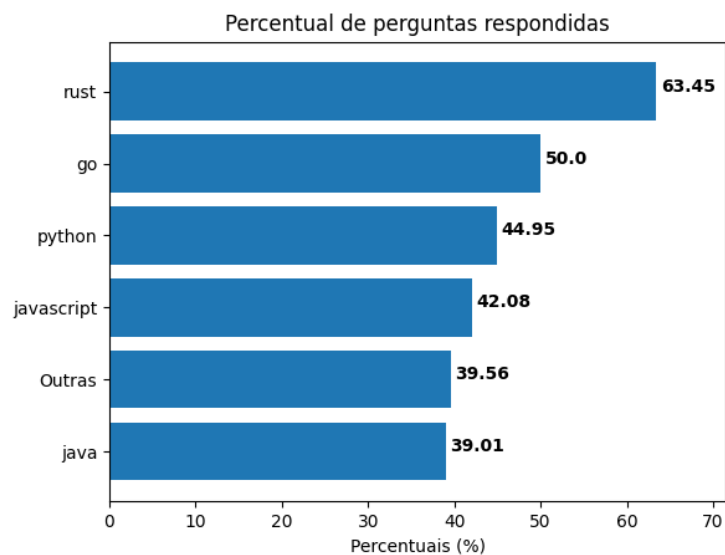
**Figure 16. Percentual de perguntas respondidas 2019**



**Figure 17. Percentual de perguntas respondidas 2020**



**Figure 18. Percentual de perguntas respondidas 2021**



**Figure 19. Percentual de perguntas respondidas 2022**