

**Laboratório de Experimentação de Software**  
**Laboratório I - Principais características de sistemas populares open-source**  
**Grupo: Davi Santos; Rafael Duarte.**

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

## **1. Introdução**

Este laboratório é um estudo sobre as principais características de sistemas populares open-source. Para isso, foram coletados os dados de 1.000 repositórios com maior número de estrelas no GitHub. A partir das informações obtidas, observou-se como estes repositórios foram desenvolvidos, analisando a frequência que recebem contribuição externas e lançam releases. Outro aspecto verificado, foram as linguagens mais utilizadas nestes repositórios.

## **2. Hipóteses Informais**

Para o direcionamento do estudo foram levantadas as seguintes hipóteses:

### **RQ 01. Sistemas populares são maduros/antigos?**

Não. Podem existir tecnologias recentes que geram engajamento em um determinado repositório, portanto seria possível a existência de um sistema popular relativamente novo.

### **RQ 02. Sistemas populares recebem muita contribuição externa?**

Sim, pelo fato dos sistemas populares terem como propósito a contribuição da comunidade.

### **RQ 03. Sistemas populares lançam releases com frequência?**

Sim, pois podem ter vários contribuidores na construção do projeto, possibilitando o lançamento de várias releases.

### **RQ 04. Sistemas populares são atualizados com frequência?**

Sim, pois existem mais pessoas atuando no projeto, o que poderia ajudar na liberação de novas atualizações.

### **RQ 05. Sistemas populares são escritos nas linguagens mais populares?**

Sim, pois o que faz a linguagem ser popular está diretamente relacionada ao tanto que é usada, portanto, sistemas populares acabam sendo construídos nestas linguagens.

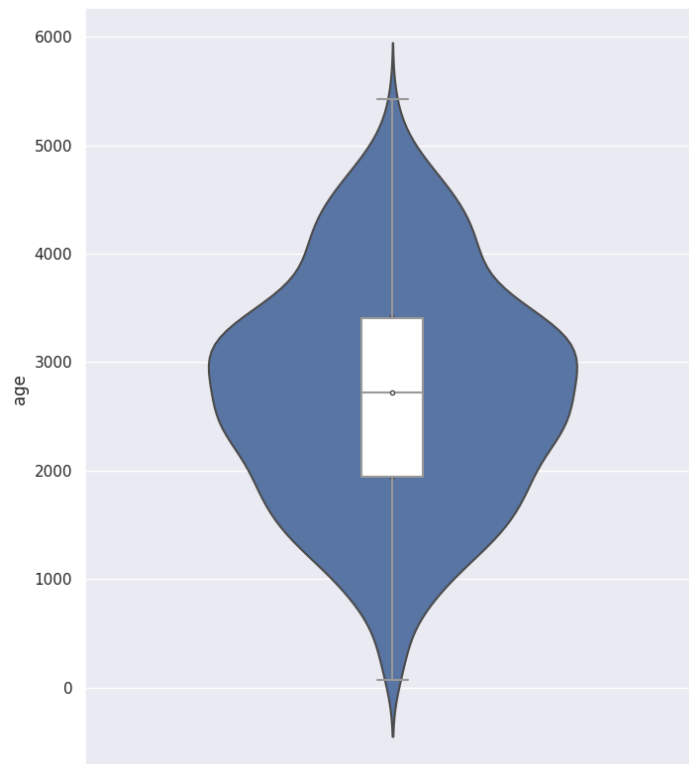
### **RQ 06. Os sistemas populares possuem um alto percentual de issues fechadas?**

Sim, pois podem ter vários contribuidores na manutenção e desenvolvimento do projeto.

## **3. Metodologia**

Para este trabalho realizou-se uma pesquisa quantitativa. Foram observados os dados dos 1000 repositórios mais populares do GitHub. As informações foram coletadas a partir da API do GitHub utilizando o GraphQL.

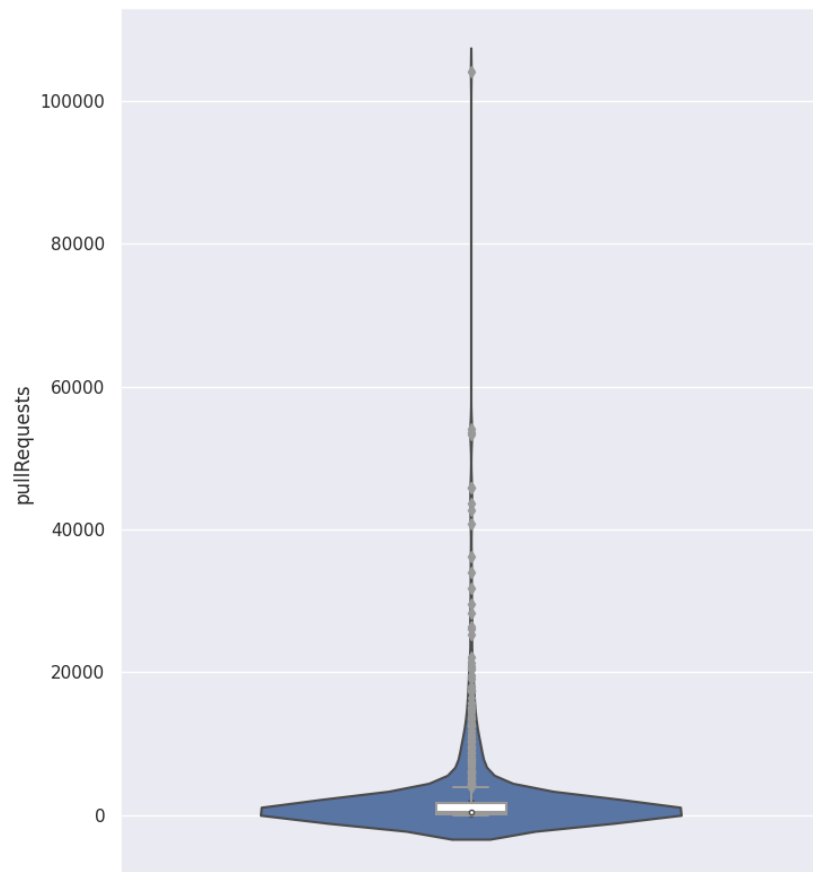
## **4. Resultados**



**Figura 1. Idade dos repositórios**

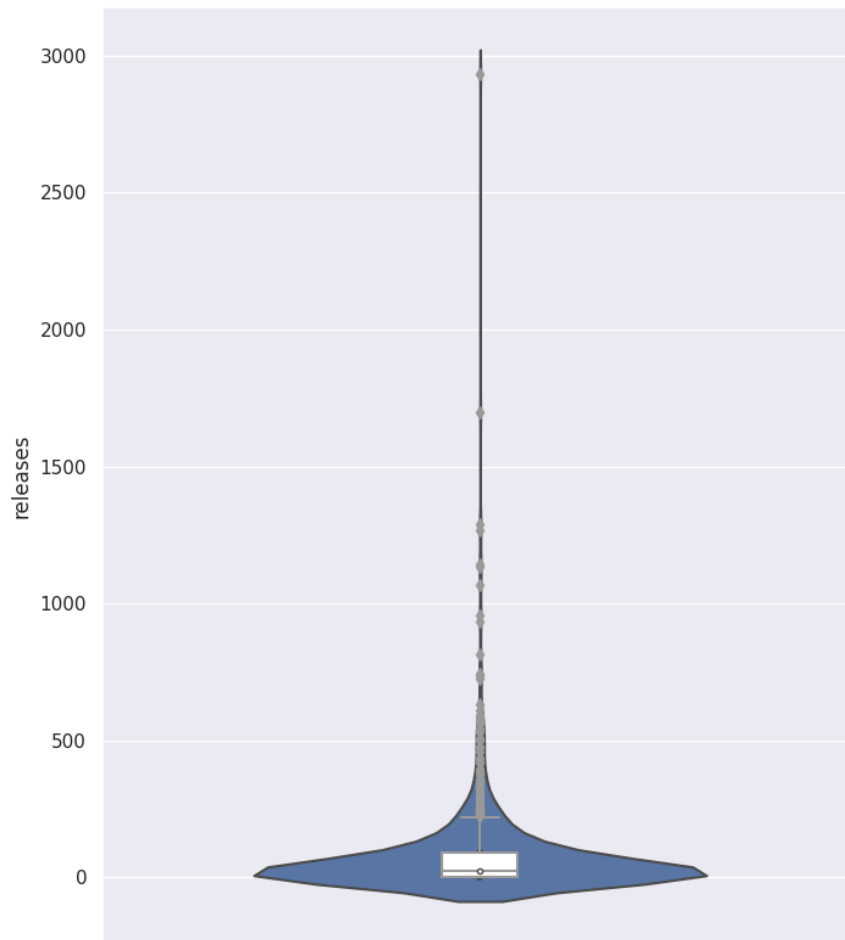
Na Figura 1, pode-se observar a idade dos repositórios em dias. A mediana é 2717 dias, o repositório mais antigo tem 5424 dias e o mais novo tem 73 dias.

Em relação a hipótese inicial, os dados mostraram uma incidência maior de repositórios entre 2000 e 3000 dias. Portanto, um sistema popular precisaria de um tempo considerável para ser conhecido, contrariando o que foi levantado inicialmente.



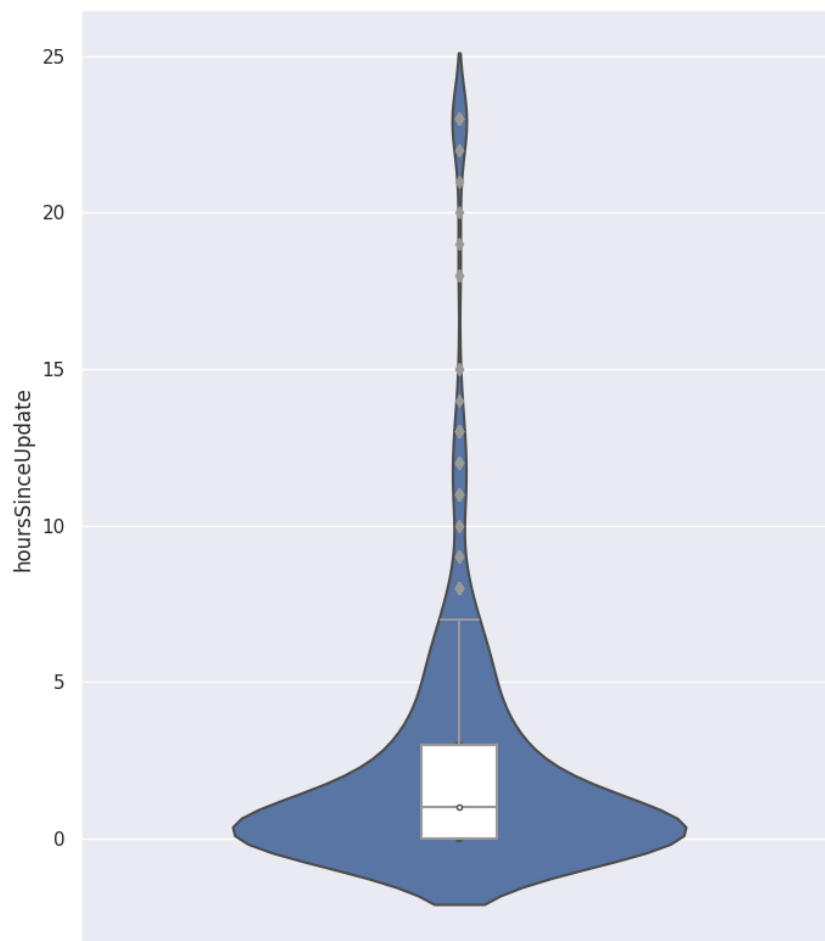
**Figura 2. Contribuições Externas - *pull requests* aceitas**

A figura 2, mostra a quantidade de contribuições externas aceitas. Estes dados são gerados através dos *pull requests* realizados nos respectivos repositórios. A mediana é de 466 *pull requests* e o repositório que possui o maior número de contribuições possui 104086. A maior parte dos repositórios se concentram em cerca de 500 *pull requests*.



**Figura 3. Total de *releases***

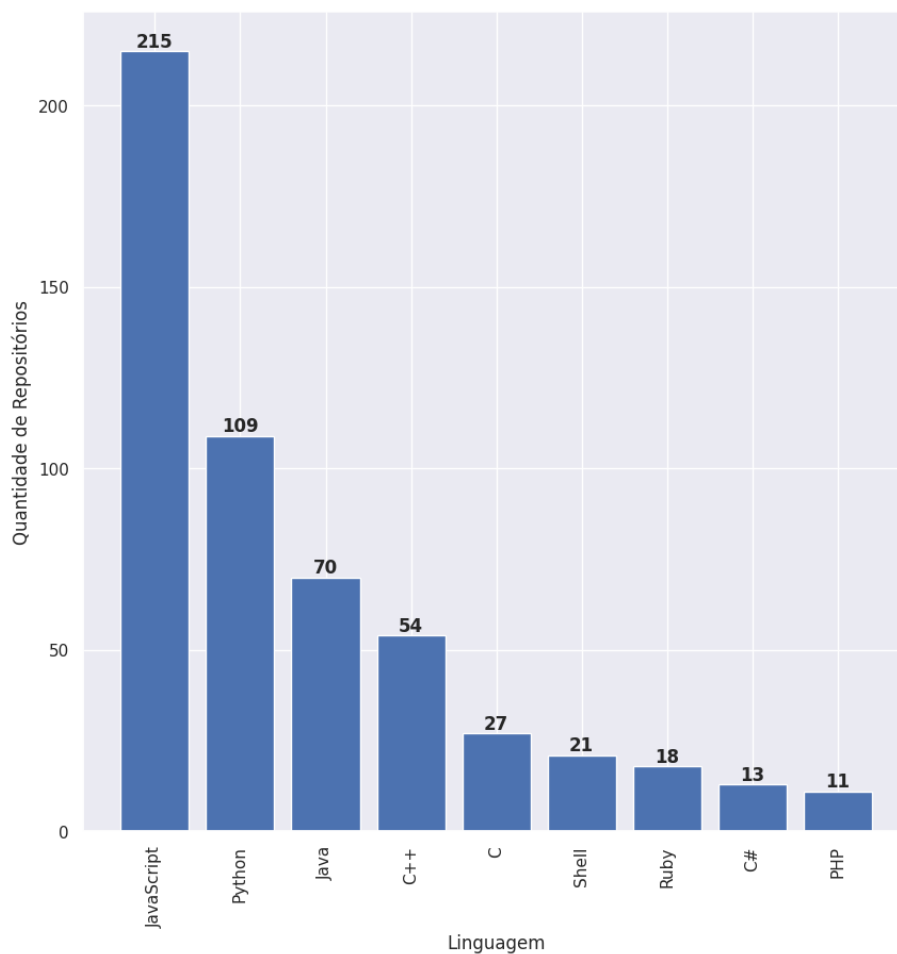
A figura 3, mostra o total de *releases*. A mediana é de 24 *releases* e o repositório que possui o maior índice tem 2934. Estes dados contrariam a hipótese inicial, pois a popularidade não está relacionada a quantidade de *releases*.



**Figura 4. Tempo em horas até a última atualização**

Na figura 4, observa-se a frequência que estes repositórios são atualizados em horas. O cálculo é realizado através da data da última atualização e a data em que se realizou a pesquisa.

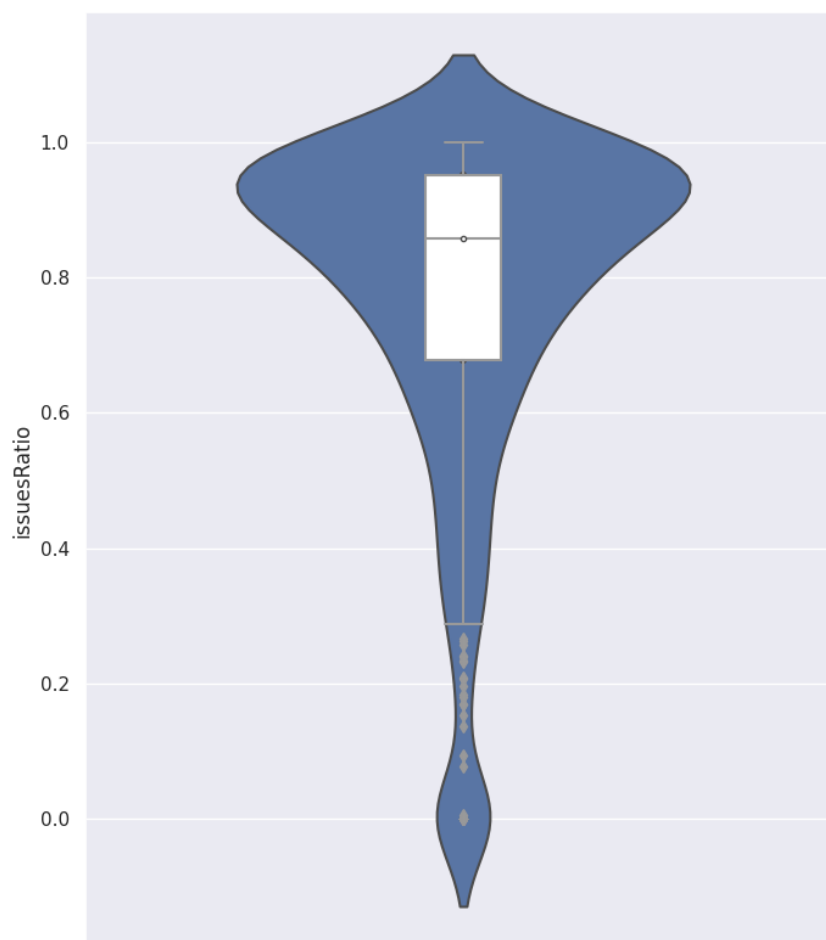
Os dados obtidos coincidiram com a hipótese inicial, pois os repositórios populares são atualizados em sua grande maioria com frequência inferior a 5 horas.



**Figura 5. Linguagem primária dos repositórios**

Na figura 5, observa-se a quantidade de repositórios em relação as 10 linguagens mais populares do GitHub.

A soma dos repositórios que utilizam linguagens populares equivale a 53,8% dos dados pesquisados, portanto a hipótese inicial corresponde com as informações extraída do GitHub.



**Figura 6. Razão entre número de *issues* fechadas pelo total de *issues***

A figura 6, mostra a razão entre o número de *issues* fechadas em relação ao total de *issues*. A mediana da razão de *issues* fechadas é 85,76%, a maior razão é 100% e a menor é 0%.

A maior parte dos repositórios possuem entre 80% e 90% de *issues* fechadas, portanto, a hipótese inicial corresponde com os dados obtidos.

## 5. Discussão

### 1. Sistemas populares são maduros/antigos?

A pergunta 1 busca compreender se sistemas populares são maduros/antigos. A hipótese levantada foi de que não necessariamente um sistema popular precisa ser antigo, visto que podem existir tecnologias recentes que geram engajamento em um determinado repositório. Os resultados obtidos na análise dos dados corroboram com essa hipótese, uma vez que a mediana de idade dos sistemas foi de 2717.5 dias, ou seja, pouco mais de 7 anos. No entanto, foi possível observar que o sistema mais antigo analisado tinha 5424 dias, o que equivale a cerca de 14 anos. Por outro lado, o sistema mais novo tinha apenas 73 dias, o que reforça a ideia de que sistemas recentes também podem ser populares.

Portanto, a popularidade de um sistema não depende exclusivamente de sua idade, mas sim de outros fatores, como sua funcionalidade, facilidade de uso e interesse da comunidade em seu desenvolvimento e manutenção. Sistemas mais antigos podem ter mais tempo de

desenvolvimento e aprimoramento, o que pode resultar em um produto mais maduro e completo, mas isso não é uma regra. Um sistema recente pode ter sido desenvolvido por uma equipe altamente qualificada e com grande foco na experiência do usuário, por exemplo, o que pode gerar um engajamento considerável por parte da comunidade. Dessa forma, é possível concluir que a idade de um sistema não é um fator determinante para a sua popularidade, mas sim sua qualidade e relevância para a comunidade de usuários e desenvolvedores.

## **2. Sistemas populares recebem muita contribuição externa?**

A pergunta 2 busca compreender se sistemas populares recebem muita contribuição externa. A hipótese levantada foi de que sim, visto que sistemas populares têm como propósito a contribuição da comunidade. Os resultados obtidos na análise dos dados apontam para essa direção, uma vez que os sistemas populares analisados apresentaram uma quantidade significativa de pull requests, que são solicitações de alterações no código fonte feitas por desenvolvedores externos ao projeto. A mediana de pull requests foi de 466, o que sugere que os sistemas populares tendem a receber uma quantidade considerável de contribuições da comunidade.

Isso indica que a popularidade de um sistema pode estar diretamente relacionada com a quantidade e qualidade das contribuições externas que ele recebe. Quanto mais atrativo o sistema for para a comunidade, maior será o número de desenvolvedores interessados em contribuir com seu desenvolvimento e manutenção. Sistemas populares tendem a ter uma comunidade ativa e engajada, o que pode resultar em uma grande quantidade de pull requests e, consequentemente, em um produto mais completo e atualizado. Por outro lado, sistemas menos populares podem ter uma base de usuários e desenvolvedores menor, o que pode impactar negativamente na quantidade e qualidade das contribuições externas recebidas.

Em suma, pode-se afirmar que a contribuição da comunidade é um fator importante para a popularidade e sucesso de um sistema. Sistemas populares tendem a atrair uma comunidade ativa e engajada, o que pode resultar em um maior número de contribuições externas e, consequentemente, em um produto mais completo e atualizado. A qualidade das contribuições externas também é um fator determinante, uma vez que a contribuição de desenvolvedores experientes e qualificados pode resultar em melhorias significativas no sistema.

## **3. Sistemas populares lançam releases com frequência?**

A pergunta 3 busca investigar se sistemas populares lançam releases com frequência. A hipótese levantada foi de que sim, visto que sistemas populares podem ter vários contribuidores na construção do projeto, possibilitando o lançamento de várias releases. A análise dos dados obtidos reforça essa hipótese, já que a mediana de releases para os sistemas populares analisados foi de 24, com o maior número de releases alcançando o valor de 2934.

Esses resultados sugerem que sistemas populares têm uma grande quantidade de lançamentos, o que pode ser atribuído, em parte, à grande quantidade de contribuidores envolvidos no desenvolvimento do projeto. Uma grande equipe de desenvolvedores pode permitir um ciclo de desenvolvimento acelerado, com maior frequência de atualizações e correções de bugs, o que pode manter o interesse da comunidade no sistema. Além disso, o lançamento frequente de novas releases pode ser uma forma efetiva de manter o sistema atualizado e funcional.



No entanto, é importante ressaltar que a frequência de lançamentos não é necessariamente um indicador de qualidade. Lançamentos frequentes podem até indicar uma alta atividade no desenvolvimento do sistema, mas sem um processo adequado de testes e validação, as novas releases podem apresentar problemas e defeitos. É importante ter um equilíbrio entre a frequência de lançamentos e a qualidade do produto final, para que o sistema seja confiável e eficiente. Em conclusão, pode-se afirmar que a frequência de lançamentos é um fator importante na popularidade de um sistema, mas deve ser acompanhada por uma atenção cuidadosa à qualidade do produto final.

#### **4. Sistemas populares são atualizados com frequência?**

A pergunta 4 investiga se sistemas populares são atualizados com frequência. A hipótese levantada foi de que sim, já que sistemas populares possuem mais pessoas atuando no projeto, o que poderia ajudar na liberação de novas atualizações. A análise dos dados obtidos aponta para essa hipótese, uma vez que a mediana de horas desde a última atualização para os sistemas populares analisados foi de apenas 1 hora, com o maior valor alcançando 23 horas e o menor, 0.

Esses resultados sugerem que sistemas populares são atualizados frequentemente, o que pode ser atribuído, em parte, ao grande número de contribuidores envolvidos no desenvolvimento do projeto. Uma equipe maior pode permitir uma maior rapidez na identificação e correção de problemas, bem como no lançamento de novas funcionalidades. Além disso, a atualização frequente pode ser uma forma efetiva de manter o sistema seguro e confiável.

No entanto, é importante lembrar que a frequência de atualizações não é o único indicador de qualidade. Atualizações frequentes podem indicar uma equipe ativa e comprometida com o desenvolvimento do sistema, mas também podem apresentar riscos, especialmente se as atualizações não forem adequadamente testadas e validadas. É crucial manter um equilíbrio entre a frequência de atualizações e a qualidade do produto final para garantir a segurança e a funcionalidade do sistema. Em conclusão, pode-se afirmar que a frequência de atualizações é um fator importante na popularidade de um sistema, mas deve ser acompanhada por uma atenção cuidadosa à qualidade do produto final.

#### **5. Sistemas populares são escritos nas linguagens mais populares?**

Ao analisar os dados obtidos sobre a linguagem de programação predominante nos repositórios populares, é possível observar que a linguagem mais utilizada é o JavaScript, com um total de 215 ocorrências, seguido pelo Python, com 109 ocorrências. Java e C++ também aparecem com frequência, com 70 e 54 ocorrências, respectivamente. Em contrapartida, as linguagens C#, PHP e Ruby apresentam uma menor quantidade de ocorrências nos repositórios populares.

Com base nesses resultados, podemos inferir que a popularidade de uma linguagem de programação pode estar diretamente relacionada com sua utilização em projetos populares de código aberto. A popularidade dessas linguagens pode ser explicada por sua facilidade de aprendizado e uso, bem como sua eficiência e capacidade de lidar com problemas complexos. Além disso, a popularidade de uma linguagem pode influenciar a sua adoção por uma comunidade de desenvolvedores, tornando-a ainda mais popular em um ciclo virtuoso. No entanto, é importante ressaltar que a escolha da linguagem também pode ser influenciada por

fatores como a finalidade do projeto, a disponibilidade de recursos e a preferência pessoal dos desenvolvedores.

Outro fator que pode influenciar a escolha da linguagem é a disponibilidade de ferramentas e recursos para a comunidade de desenvolvedores. Por exemplo, uma linguagem pode ser popular devido à disponibilidade de bibliotecas e frameworks que facilitam o desenvolvimento de aplicativos e sistemas. Além disso, a popularidade de uma linguagem também pode ser influenciada pela sua capacidade de integração com outras tecnologias populares. Por exemplo, a linguagem JavaScript é frequentemente utilizada em conjunto com frameworks como React e Node.js, que são amplamente utilizados no desenvolvimento web. Em resumo, a escolha da linguagem de programação pode ser influenciada por uma série de fatores, e a popularidade da linguagem pode ser um reflexo da sua adequação para atender às necessidades da comunidade de desenvolvedores.

## **6. Sistemas populares possuem um alto percentual de issues fechadas?**

A pergunta "Os sistemas populares possuem um alto percentual de issues fechadas?" busca analisar a relação entre a popularidade de um projeto e sua capacidade de gerenciar e resolver problemas levantados pelos usuários. A hipótese formulada sugere que sistemas populares possuem um alto percentual de issues fechadas, uma vez que a maior quantidade de contribuidores pode agilizar o processo de solução de problemas.

Ao analisar os dados obtidos, é possível verificar que a mediana do percentual de issues fechadas é de 0.857, o que representa um valor relativamente alto. Além disso, o percentual máximo é de 1.0, o que indica que alguns projetos populares conseguem solucionar todas as issues levantadas. Esses resultados corroboram a hipótese formulada, indicando que a popularidade de um projeto pode ser um fator importante para garantir que os problemas reportados pelos usuários sejam solucionados com eficiência.

No entanto, é importante ressaltar que a análise do percentual de issues fechadas por si só não é suficiente para avaliar a qualidade de um projeto. Outros fatores como a velocidade de resolução dos problemas, a capacidade de lidar com problemas complexos e a satisfação dos usuários também são importantes e devem ser considerados. Além disso, é preciso levar em conta que projetos menos populares podem ter uma taxa de resolução de issues igualmente alta, mesmo que com uma quantidade menor de contribuidores.