

Explorando o YouTube como Fonte de Aprendizado para Desenvolvedores: Uma Análise em Larga Escala sobre Vídeos de Tecnologia

Gabriel Victor de Paula¹, João Victor Guerra¹, Luís Antônio Souza¹,
Luiz Gustavo Soares¹, Pedro Henrique Machado¹, Aline Brito^{1,2}, Laerte Xavier¹

¹ Departamento de Engenharia de Software – Instituto de Ciências Exatas e Informática – Pontifícia Universidade Católica de Minas Gerais (PUC Minas)

² Departamento de Computação – Instituto de Ciências Exatas e Biológicas
Universidade Federal de Ouro Preto (UFOP)

{gabriel.paula.1265840, joao.lima.1322940, lassousa,

luiz.soares.1310151, pedro.machado.1277720}@sga.pucminas.br

{alinebrito, laertexavier}@sga.pucminas.br

Abstract. *YouTube is a platform widely used by the software development community. It hosts videos covering a variety of scenarios, which developers use for learning and staying updated. For instance, there are videos on new techniques and trends in software development, framework usage, environment configurations, and best practices, among others. Consequently, YouTube videos have become valuable resources for supporting software development by providing easy access to up-to-date and practical knowledge. In this context, in this paper, we analyzed approximately 11K videos related to software development. Our analysis revealed several characteristics of these videos, for instance, the results suggest a significant number of videos for beginners involving Python, as well as popular videos that are short. Finally, we discuss characteristics and future research that can help in the creation of relevant programming videos.*

Resumo. *YouTube é uma plataforma amplamente utilizada pela comunidade de desenvolvimento de software. Existem vídeos abrangendo diversos cenários e linguagens de programação, que são utilizados pelos desenvolvedores para aprendizado e atualização. Por exemplo, há vídeos sobre novas técnicas e tendências em desenvolvimento de software, uso de frameworks, configurações de ambiente e melhores práticas, dentre outros. Consequentemente, os vídeos do YouTube se tornaram recursos importantes para apoiar o desenvolvimento de sistemas de software, proporcionando acesso fácil ao conhecimento atualizado e prático. Neste contexto, neste trabalho, analisam-se aproximadamente 11 mil vídeos relacionados ao desenvolvimento de software. A análise revela diversas características dos vídeos, por exemplo, os resultados sugerem um número significativo de vídeos para iniciantes envolvendo Python, bem como vídeos populares com curta duração. Por fim, discute-se características e futuras pesquisas que podem auxiliar na criação de vídeos de programação relevantes.*

1. Introdução

O YouTube tornou-se uma plataforma essencial para a disseminação do conhecimento e promoção do aprendizado em diversas áreas no mundo moderno [Carvalho et al. 2022], incluindo o desenvolvimento de *software* [MacLeod et al. 2015, Poché et al. 2017]. A plataforma hospeda milhares de vídeos, sendo adotada por grandes empresas, fóruns, e membros da comunidade de *software* [Ponzanelli et al. 2016]. Por exemplo, o canal *Google for Developers*,¹ conta com mais de dois milhões de inscritos, onde são compartilhados conteúdos sobre inteligência artificial, técnicas de programação, boas práticas, dentre outros.

Consequentemente, assim com fóruns de discussões, os vídeos do YouTube se tornaram recursos relevantes para esclarecimento de dúvidas e apoio ao desenvolvimento de *software* [MacLeod et al. 2015, Ponzanelli et al. 2016, Poché et al. 2017]. A relevância da plataforma tornou-se maior nos últimos anos, devido à demanda por materiais didáticos em modalidades de ensino assíncrona e híbrida [Castro et al. 2022, Goedert and Arndt 2020].

Especificamente, a plataforma pode ser considerada como um repositório de Objetos de Aprendizagem, visto que envolve conteúdos de mídia que podem suportar o aprendizado mediado por tecnologia [Carvalho et al. 2022]. Do ponto de vista educacional, desenvolvedores e iniciantes em programação podem acompanhar as novas técnicas e se manterem atualizados com as tendências da área de desenvolvimento de *software*. Um estudo recente reforça esta observação, mostrando que o Youtube é o segundo *website* mais popular dedicado ao desenvolvimento de *software* [Hora 2021].

Como em qualquer outra plataforma utilizada para ensino ou aprendizagem na área da computação, o contexto é diverso e amplo, pois existem várias tecnologias e linguagens de programação, bem como pessoas com níveis de conhecimento diferentes. Consequentemente, existem desafios, como por exemplo, a forma como o conteúdo é indexado e buscado, vocabulário utilizado, dentre outros [Carvalho et al. 2022]. Dessa forma, estudar as características dos vídeos sobre tecnologia no YouTube é fundamental para entender como a plataforma opera para a disseminação do conhecimento na área. Além disso, pode-se compreender melhor os elementos que tornam os vídeos mais atrativos. Entretanto, ao melhor do nosso conhecimento, análises em larga escala sobre conteúdos de tecnologia não são profundamente exploradas na literatura recente.

Neste trabalho, analisam-se as características de certa de 11 mil vídeos que remetem a termos da área da computação e linguagens de programação populares. Especificamente, discute-se a popularidade dos vídeos, tempo de duração de vídeos populares, e frequência de vídeos voltados para desenvolvedores iniciantes. Existe uma taxa significativa de vídeos de curta e média duração na área da tecnologia (i.e., aproximadamente 10 minutos). Entretanto, os resultados também mostram vídeos longos, que correspondem a cursos completos envolvendo o desenvolvimento de *software*. Além disso, os resultados mostram que milhares de pessoas fazem uso da plataforma para capacitação, sendo que existem também vídeos exclusivos para iniciantes em programação.

O restante deste artigo está organizado conforme descrito a seguir. A Seção 2 discute os trabalhos relacionados e fundamentação teórica. A Seção 3 detalha a metodologia,

¹<https://www.youtube.com/googledevelopers>

onde apresenta-se os procedimentos realizados para mineração e análise dos vídeos. Na Seção 4, os resultados do trabalho são apresentados, sendo, em seguida, discutidos na Seção 5. A Seção 6 inclui os riscos à validade. Por fim, a Seção 7 conclui o estudo, prospectando futuras linhas de pesquisa.

2. Trabalhos Relacionados e Fundamentação Teórica

A plataforma Youtube tem sido amplamente utilizada como fonte de conhecimento nos últimos anos no contexto de desenvolvimento de *software* [Ponzanelli et al. 2016, MacLeod et al. 2015, Ponzanelli et al. 2016, Poché et al. 2017]. Por esse motivo, a mesma tem sido alvo de pesquisas recentes visando um melhor uso dos recursos e engajamentos dos usuários. Em um estudo recente, por exemplo, os autores analisaram os comentários [Nawaz et al. 2019], com o objetivo de identificar a viabilidade de um índice de qualidade criado a partir da análise de sentimento dos respectivos vídeos. Existem pesquisas que propõe aplicações baseadas em *machine learning* para analisar a qualidade de vídeos a partir de seus comentários e curtidas, comparando-as com outras abordagens [Dabas et al. 2019]. Especificamente, esses estudos serviram como inspiração para a definição da metodologia adotada neste trabalho.

No contexto da análise de sentimentos, existem pesquisas que analisam comentários em vídeos educacionais, visando avaliar o aprendizado autodirigido [Lee et al. 2017]. Os resultados mostraram tendências positivas, mas revelam que não é possível garantir uma relação direta entre a qualidade e a popularidade dos vídeos e sua respectiva eficácia. Os resultados também motivaram o trabalho atual, assim como outros estudos em contextos similares [Sureka 2011, Agarwal and Sureka 2014]. Os mesmos evidenciam o uso YouTube como instrumento de ensino e compartilhamento de conhecimento. Entretanto, nenhum destes trabalhos concentram-se em análises em larga escala sobre linguagens de programação populares abordadas em vídeos no Youtube.

Adicionalmente, Storey et al. (2014) investigaram como as mídias sociais afetavam o trabalho e o processo de aprendizagem dos programadores. Com o surgimento de ferramentas como GitHub e Stack Overflow, a interação entre programadores em comunidades de tecnologia tornou-se cada vez mais ampla e forte. Nesse sentido, o YouTube também se destacou como um ambiente para a criação de comunidades. Sua atuação se assemelha aos antigos blogs de tecnologia, porém com o benefício do vídeo, que permite a execução simultânea da ação. Este trabalho segue a lógica da análise trazida por Storey et al. (2014), porém aplicada à realidade atual, na qual vídeos e canais do YouTube representam novos tipos de comunidade.

No âmbito educacional, a perspectiva do desenvolvedor que busca atualização através do Youtube pode se relacionar, por exemplo, com os métodos construtivistas [Becker 1992], isto é, com a maneira de construir o conhecimento. Especificamente, a plataforma possibilita um estudo autônomo e alinhado aos interesses individuais por meio das mídias sociais.

O uso do Youtube para fins educacionais pode se relacionar também com a teoria da aprendizagem social [Le and Hancer 2021, Bandura 1978], que consiste no aprendizado baseada em observação. Especificamente, os desenvolvedores podem aprender observando especialistas que disponibilizam diversos tutoriais, além de serem capazes de interagir com outros membros da comunidade de *software* nos comentários dos vídeos.

Por fim, no contexto de Objetos de Aprendizagem—“qualquer coisa que se possa referir, descrever, discutir ou experimentar” [Braga 2014]—o YouTube se destaca como uma fonte de informação para o processo de aprendizado [Carvalho et al. 2022]. Dessa forma, os vídeos na plataforma podem ser considerados Objetos, enquanto o próprio YouTube assume o papel de Classe, uma coleção específica de objetos. Já o conteúdo desses vídeos representa os Atributos, responsáveis por diferenciá-los entre si.

3. Metodologia

A metodologia deste trabalho inclui quatro grandes etapas: (i) definição de termos para busca de vídeos, (ii) mineração de vídeos candidatos na plataforma YouTube, (iii) identificação de vídeos que mencionam linguagens de programação e (iv) análise da popularidade e características dos vídeos. Nas próximas subseções, detalha-se cada uma dessas etapas.

3.1. Mineração de Vídeos Candidatos no YouTube

A primeira etapa deste estudo consistiu na definição de termos para mineração dos vídeos. Para tanto, foi definido um conjunto de palavras-chave, que se baseiam em conceitos e práticas relacionadas à engenharia *software*, sendo frequentemente reportados na literatura recente. Especificamente, o conjunto inclui 87 termos, envolvendo cenários distintos, como palavras-chave relacionadas à cursos e boas práticas de programação (ex., “*Full Stack Development*”, “*GraphQL API Development*” e “*Coding Practices*”), desenvolvimento de aplicativos móveis (ex., “*Mobile App Development*”), e revisão de código (ex., “*Code Review Guidelines*”). Além disso, foram considerados termos gerais que remetem ao desenvolvimento de *software* (ex., “*Computer Science*” e “*Software Engineering*”). A lista de palavras-chave encontra disponível no pacote de replicação deste estudo.²

Em seguida, elaborou-se um conjunto de *scripts* para buscar vídeos considerando esses termos de busca. Para mineração em larga escala, fez-se uso da *Application Programming Interface* (API) provida pelo YouTubeData.³ Após esta etapa, os vídeos foram filtrados, visando remover conteúdos repetidos, vídeos removidos pelo dono do canal, ou vídeos sem conteúdo (isto é, zero segundos). Ao final, obteve-se-se 45.652 vídeos candidatos. Os metadados incluem diversas informações, como por exemplo, título, descrição, data da publicação, número de visualizações e métricas de engajamento dos vídeos.

3.2. Identificação das Linguagens de Programação

A terceira etapa consistiu na identificação das linguagens de programação abordadas em cada vídeo. A lista de linguagens consideradas para busca é baseada no *ranking* definido pelo GitHub [Lopes and Hora 2022].⁴ Portanto, para cada vídeo, buscou-se a ocorrência de 10 linguagens de programação mais populares nos seus campos textuais (título e descrição): JavaScript, Python, Java, TypeScript, C#, C++, PHP, Shell, C e Ruby. Esta etapa resultou em 11.850 vídeos, que são utilizados neste estudo. A Tabela 1 detalha

²github.com/alinebritto/educomp2025-replication-package-youtuber/blob/main/dataset/termos_pesquisa.csv

³developers.google.com/youtube/v3/docs?hl=pt-br

⁴octoverse.github.com/2022/top-programming-languages

o conjunto de dados. Como pode-se observar aproximadamente 90% dos vídeos compreendem conteúdos relacionados com as linguagens Java, JavaScript, TypeScript, Python, C e C++.

Ling. Programação	Vídeos	%
C e C++	3.530	29,6
Python	3.319	28,0
JavaScript e TypeScript	2161	15,9
Java	2099	17,7
PHP	476	4,0
Shell	154	1,3
Ruby	66	0,6
C#	45	0,4
Total	11.850	100

Tabela 1. Frequência das linguagens de programação por vídeo no YouTube

A Figura 1 mostra um exemplo de vídeo popular no YouTube, que é retornado ao utilizar os termos de busca definidos neste estudo. O conteúdo é disponibilizado pela FreeCodeCamp, uma plataforma web popular para aprendizagem interativa.⁵ Neste exemplo, o vídeo envolve um curso de programação em JavaScript para desenvolvedores iniciantes, contando com mais de 18 milhões de visualizações, 269 mil *likes* e mais de 8 mil comentários (Dezembro, 2024). Um material didático similar para a linguagem Python, por exemplo, possui mais de 45 milhões de visualizações e 44 mil comentários.⁶

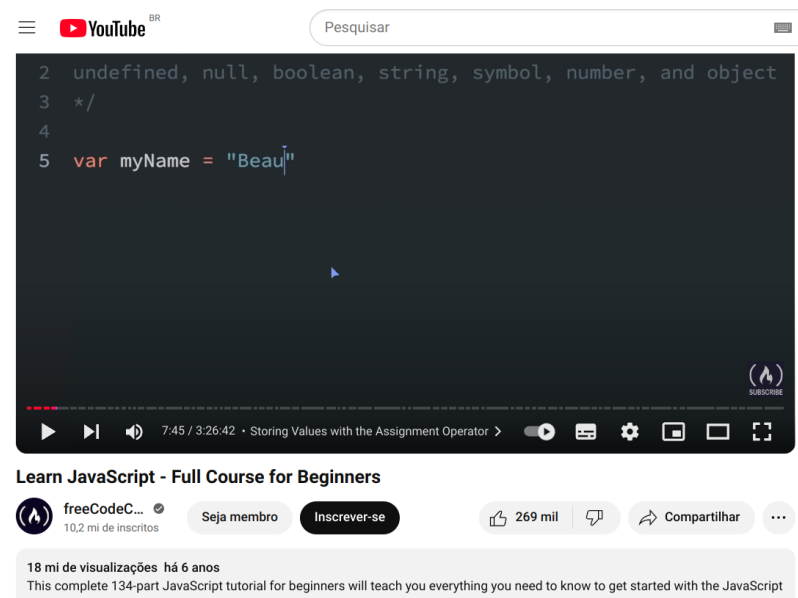


Figura 1. Exemplo de vídeo do YouTube envolvendo o ensino da linguagem de programação JavaScript (Plataforma FreeCodeCamp)

⁵www.youtube.com/watch?v=PkZNo7MFNFg
⁶www.youtube.com/watch?v=rpscVS0vtbw

3.3. Categorização dos Vídeos Populares e Não Populares

Seguindo a mesma estratégia adotada por estudos anteriores para classificar bibliotecas e aplicativos populares e não populares [Xavier et al. 2017, Lima et al. 2018, Tian et al. 2015], os vídeos selecionados foram ordenados pelo número de visualizações, visto que é uma métrica para indicar a relevância e impacto na plataforma. Em seguida, os mesmos foram separados em dois grandes grupos: *populares* e *não populares*. O primeiro grupo refere-se aos vídeos com maior taxa de visualizações (*top-25%*); e o segundo grupo refere-se aos vídeos com o menor número de visualizações (*bottom-25%*). Dessa forma, comparam-se os vídeos pertencentes aos dois grupos, visando identificar características que se destacam em vídeos populares.

4. Resultados

4.1. (QP1) Qual é a popularidade dos vídeos sobre tecnologia no YouTube?

Nesta primeira questão de pesquisa, analisam-se métricas de engajamento dos 11 mil vídeos. A Figura 2 mostra a distribuição dos valores, considerando o número de visualizações, curtidas e comentários. A mediana do número de visualizações é 10.755, com valores variando em até milhões de visualizações. Em relação ao número de curtidas, os valores também são significativos. Por exemplo, existem vídeos com centenas de curtidas. O número de comentários varia entre zero e cerca de 121 mil, sendo que até 75% dos vídeos possuem até 113 comentários.

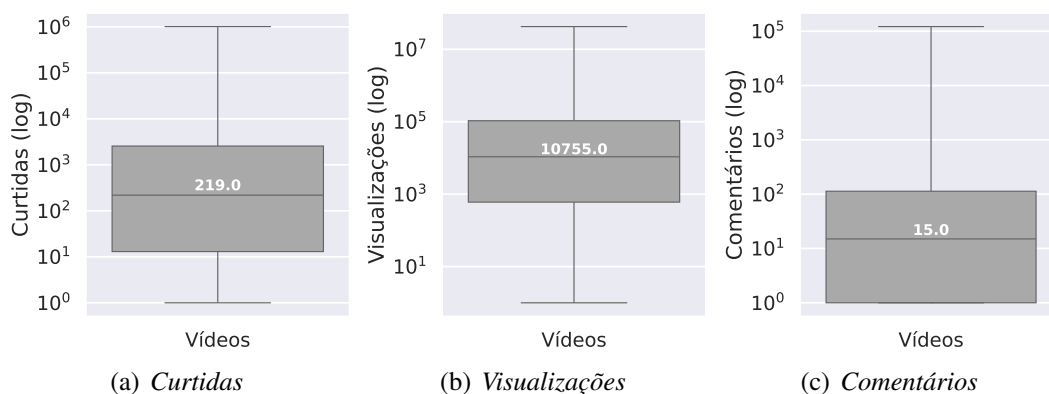


Figura 2. Distribuição de métricas de popularidade dos vídeos sobre tecnologia

O vídeo com o maior número de comentários—aproximadamente 121 mil—envolve um curso para pessoas que desejam aprender a programar utilizando a linguagem Python.⁷ Os comentários no vídeo são diversos, envolvendo, por exemplo, dúvidas, gabaritos das atividades, e avaliação do conteúdo. A Figura 3(a) mostra um exemplo, no qual um desenvolvedor sugere uma solução melhor para um problema de programação, indicando o momento do vídeo relacionado à discussão no seu comentário. Como pode-se observar, outros usuários podem responder o comentário, similar ao que ocorre em fóruns populares como Stack Overflow. Já a Figura 3(b) mostra um comentário elogiando o conteúdo do vídeo, que inclui mais de 1,4 mil curtidas de outros usuários.

⁷www.youtube.com/watch?v=gfDE2a7MKjA

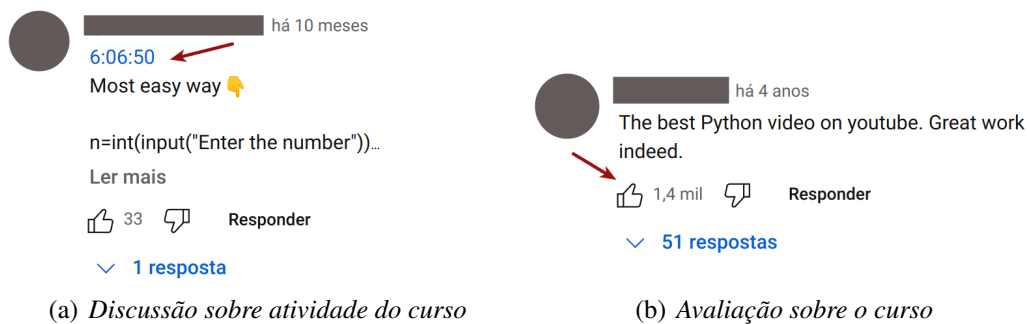


Figura 3. Exemplos de comentários em vídeos de tecnologia no Youtube

A Figura 4 mostra o número de visualizações dos vídeos por linguagem de programação. Como pode-se observar, os valores variam em torno de milhares de visualizações para a maioria das linguagens, considerando os valores medianos.

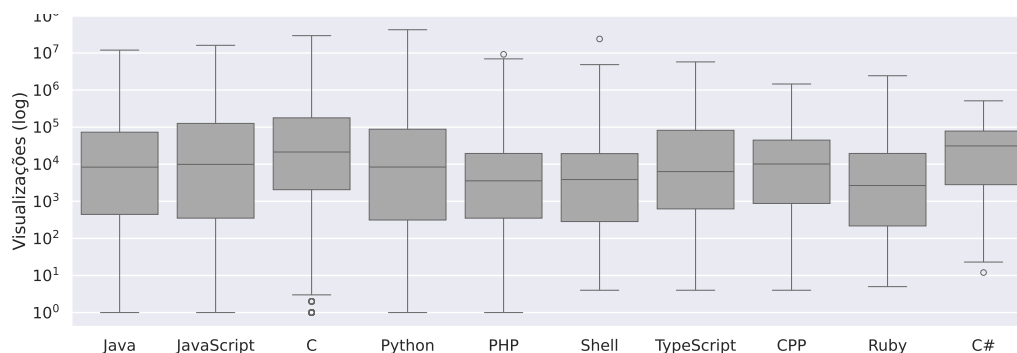


Figura 4. Distribuição do número de visualizações dos vídeos por linguagem de programação no Youtube

4.2. (QP2) Qual é o tempo de duração dos vídeos populares sobre tecnologia no YouTube?

Nesta questão de pesquisa compara-se o tempo de duração dos vídeos *populares* (*top-25%*) e *não populares* (*bottom-25%*). Observa-se que os vídeos *populares* estudados são significativamente diferentes do segundo grupo ($p\text{-value} < 0.01$ para o teste *Mann-Whitney*).

A Figura 5(a) mostra a distribuição do tempo de duração dos vídeos *populares*. Os valores variam de vídeos com alguns segundos até vídeos com horas de duração, com uma mediana de 820 segundos (isto é, aproximadamente 13 minutos). O vídeo mais longo do conjunto de dados possui cerca de 23 horas, e consiste em um curso sobre como desenvolver uma aplicação em C#, detalhando todas as etapas.⁸ Já a Figura 5(b) apresenta a distribuição do tempo para o grupo de vídeos *não populares*. Considerando os valores medianos, existe pouca variação no tempo de duração entre os grupos, isto é, em ambos, observa-se uma predominância de vídeos de curta e média duração.

⁸www.youtube.com/watch?v=wFWxdh-_k_4

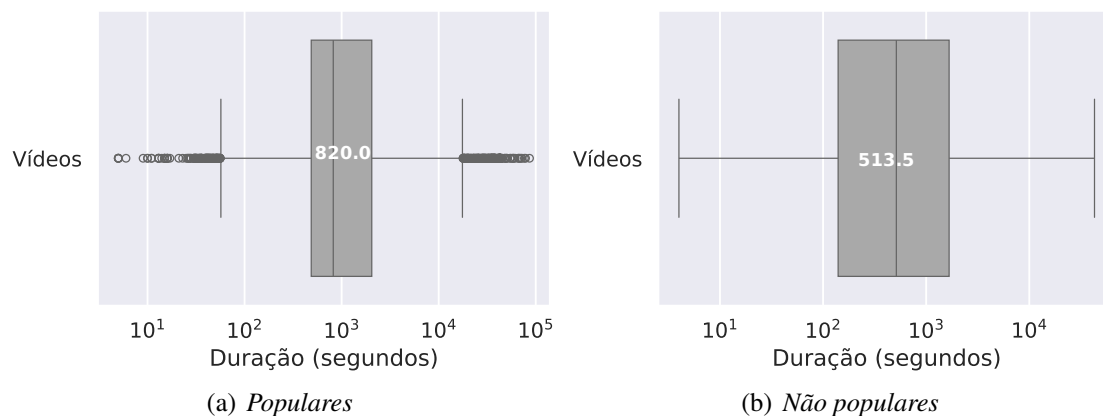


Figura 5. Distribuição da duração dos vídeos sobre tecnologia no Youtube por popularidade

4.3. (QP3) Qual é a frequência dos vídeos de tecnologia voltados para programadores iniciantes no YouTube?

Para identificar vídeos com conteúdos iniciantes, buscou-se por um conjunto de palavras-chaves no título e descrição. Especificamente, considerou-se os termos *start*, *begin*, *entry*, e *learn*, bem como suas variações verbais.

A Tabela 2 mostra uma visão geral dos resultados por linguagem de programação, desconsiderando vídeos que não possuem as palavras-chave. Em 2.116 vídeos, existem os termos que remetem à conteúdos para iniciantes. Como pode-se observar, Python é a linguagem com a maior frequência de conteúdos para iniciantes (1.048 ocorrências). Existem também taxas significativas de vídeos para iniciantes envolvendo as linguagens C, C++, Java e JavaScript.

Ling. Programação	Vídeos	%
Python	1.048	49,5
C e C++	414	19,6
Java	382	18,1
JavaScript e TypeScript	176	8,3
PHP	62	2,9
Shell	20	0,9
Ruby	11	0,5
C#	3	0,1
Total	2.116	100

Tabela 2. Frequência de vídeos sobre tecnologia para iniciantes no Youtube

A Figura 6 mostra um exemplo, no qual os autores do conteúdo mencionaram explicitamente no título e descrição que o vídeo é voltado para iniciantes em programação com JavaScript.⁹ O vídeo possui mais de 43 mil curtidas e 1,2 milhão de visualizações.

⁹www.youtube.com/watch?v=lI1ae4REbFM

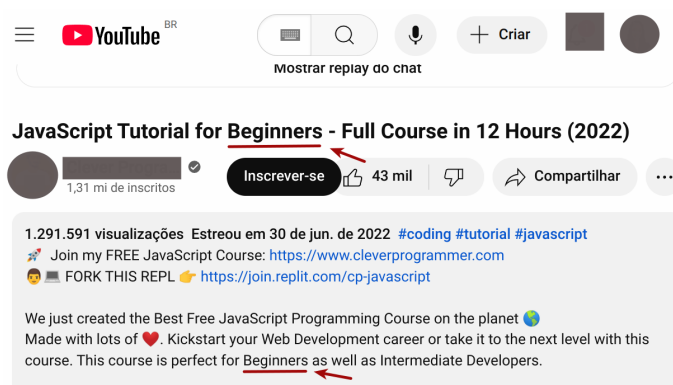


Figura 6. Exemplo de vídeo sobre tecnologia voltado para iniciantes no Youtube

5. Discussão

Os resultados sugerem que *educadores e desenvolvedores de conteúdo* devem produzir vídeos de curta duração, visando um maior engajamento. É importante considerar também tendências de mercado e características das linguagens de programação utilizadas. Adicionalmente, o YouTube é popular no contexto educacional, podendo se beneficiar de novas funcionalidades voltadas para educadores e estudantes de tecnologia, como integração com ambientes de programação e suporte a sintaxes para escrita de texto, similares a plataformas como o GitHub.

Dentre os vídeos analisados, a linguagem de programação Python se destaca em relação aos conteúdos para iniciantes (1.048 ocorrências, 49,5%). Os resultados sugerem que linguagens interpretadas são populares para elaboração de conteúdos voltados para capacitação e conhecimentos gerais, porém, com distinção em relação ao nível de conhecimento do público-alvo. Além disso, os resultados complementam a perspectiva de *rankings* populares, como o TIOBE,¹⁰ GitHub,¹¹ e StackOverflow,¹² bem como a adoção dessas linguagens em outros nichos [Meyerovich and Rabkin 2013].

Em relação às características dos vídeos, pode-se observar a predominância de vídeos de curta e média duração envolvendo linguagens de programação. Por exemplo, a mediana dos vídeos *populares* é de aproximadamente 13 minutos, enquanto a mediana de duração dos vídeos *não populares* é cerca de 9 minutos. Os resultados podem estar relacionados com critérios da plataforma Youtube para monetização. Por exemplo, vídeos com duração inferior à 10 minutos podem ter apenas um anúncio publicitário. Adicionalmente, observou-se ocorrências de vídeos com maior duração abordando tutoriais ou introduzindo novos conceitos de forma aprofundada, bem como, vídeos curtos mostrando instruções rápidas, dicas ou resolução de problemas recorrentes e específicos.

Dessa forma, vislumbram-se futuras linhas de pesquisa envolvendo a inspeção manual dos vídeos. Especificamente, deseja-se identificar quais são as linguagens de programação mais adotadas por nicho de mercado, bem como aprofundar os estudos sobre as linguagens mais adequadas por nível de conhecimento. Além disso, outras características podem ser identificadas, visando a criação de diretrizes para aumentar o engaja-

¹⁰<https://www.tiobe.com/tiobe-index>

¹¹<https://octoverse.github.com/2022/top-programming-languages>

¹²<https://survey.stackoverflow.co/2023/>

mento em vídeos relacionados ao ensino de programação (ex.: uso de *tags*).

Adicionalmente, existe uma vasta literatura que baseia-se em linguagens de programação populares, como, estudos empíricos [Brito et al. 2023, Bibiano et al. 2019, Lopes and Hora 2022, Vándér et al. 2024] e diversas ferramentas para melhorar a produtividade dos desenvolvedores, tais como soluções para revisão de código [Brito and Valente 2021, Hayashi et al. 2013, Ge et al. 2014, Ge et al. 2017], bem como refatoração e manutenção [Tsantalis et al. 2020, Silva and Valente 2017, Brito et al. 2018, Spadini et al. 2018]. Dessa forma, este estudo complementa trabalhos anteriores, destacando a importância de considerar também redes sociais e plataformas digitais para compreender melhor as demandas do público, e desenvolver ferramentas alinhadas às tendências de mercado de engenharia de *software*.

6. Ameaças à Validade

Nesta seção, apresenta-se as ameaças à validade deste estudo e as estratégias utilizadas para mitigá-las [Wohlin et al. 2012]. Primeiro, como comum em estudos empíricos, os resultados não podem ser generalizados para outros cenários. A popularidade e disponibilidade dos vídeos também pode ser influenciada por fatores externos, como o tempo desde a publicação e as estratégias de *marketing*. Entretanto, o estudo compreende uma análise em larga escala com mais de 11 mil vídeos, com diferentes características.

Adicionalmente, a busca foi baseada em um conjunto de 87 palavras-chave, que podem não capturar todo o conteúdo sobre tecnologia. O mesmo se aplica para a identificação das linguagens de programação, que é passível de erros, como usual em técnicas de processamento de texto em linguagem natural. Para mitigar essa ameaça, selecionou-se um conjunto diversificado de palavras-chave, bem como linguagens de programação listadas em *rankings* reconhecidos pela comunidade de *software*. Os termos utilizados, bem como os vídeos analisados, encontram-se disponíveis no pacote de replicação deste estudo.

Por fim, a identificação de vídeos para iniciantes também foi baseada em palavras-chave presentes no título e descrição. No entanto, a estratégia reflete práticas comuns no YouTube para atrair o público-alvo desejado.

7. Conclusão

Neste estudo, foram analisados cerca de 11 mil vídeos relacionados a termos que remetem a tecnologia e desenvolvimento de *software* na plataforma YouTube. O conjunto de dados inclui vídeos que cobrem cenários distintos, como inteligência artificial e estruturas de dados. Analisou-se característica como a duração dos vídeos, o número de visualizações, curtidas, bem como conteúdos voltados para iniciantes. Os resultados sugerem uma taxa significativa de conteúdos de curta e média duração. Além disso, a linguagem Python destaca-se entre os vídeos direcionados para programadores iniciantes.

Trabalhos futuros podem incluir diretrizes para a criação de vídeos mais eficazes direcionados para desenvolvimento de *software*. A análise temática dos comentários dos vídeos também pode ser realizada, visando identificar os principais desafios enfrentados pelos desenvolvedores.

O pacote de replicação deste estudo encontra-se disponível em: github.com/alinebrito/educomp2025-replication-package-youtuber

Referências

- Agarwal, S. and Sureka, A. (2014). A focused crawler for mining hate and extremism promoting videos on youtube. page 294–296. Association for Computing Machinery.
- Bandura, A. (1978). Self-efficacy: Toward a unifying theory of behavioral change. *Advances in Behaviour Research and Therapy*, 1(4):139–161.
- Becker, F. (1992). O que é construtivismo. *Revista de educação AEC, Brasília*, 21(83):7–15.
- Bibiano, A. C., Garcia, E. F. D. O. A., Kalinowski, M., Fonseca, B., Oliveira, R., Oliveira, A., and Cedrim, D. (2019). A quantitative study on characteristics and effect of batch refactoring on code smells. In *13th International Symposium on Empirical Software Engineering and Measurement (ESEM)*, pages 1–11.
- Braga, J. C. (2014). *Objetos de Aprendizagem: Introdução e Fundamentos*. UFABC.
- Brito, A., Hora, A., and Valente, M. T. (2023). Towards a catalog of composite refactorings. *Journal of Software: Evolution and Process*, e2530.
- Brito, A., Xavier, L., Hora, A., and Valente, M. T. (2018). APIDiff: Detecting API breaking changes. In *25th International Conference on Software Analysis, Evolution and Reengineering (SANER), Tool Track*, pages 507–511.
- Brito, R. and Valente, M. T. (2021). RAID - Refactoring aware and intelligent diffs. In *29th International Conference on Program Comprehension (ICPC)*, pages 265–275.
- Carvalho, H. C. F. B., Dorça, F. A., Pitangui, C. G., Assis, L. P. d., Andrade, A. V., and Trindade, E. A. C. (2022). Classificação automática de vídeos educacionais por meio de comentários apoiada por técnicas de aprendizado de máquina: uma análise experimental utilizando o youtube. *Revista Brasileira de Informática na Educação*, 30:419–448.
- Castro, R., Classe, T., and Siqueira, S. (2022). Técnicas e tecnologias diversas no ensino remoto emergencial de engenharia de software. In *II Simpósio Brasileiro de Educação em Computação (Educomp)*, pages 163–170.
- Dabas, C., Kaur, P., Gulati, N., and Tilak, M. (2019). Analysis of comments on youtube videos using hadoop. In *5th International Conference on Image Information Processing (ICIIP)*, pages 353–358.
- Ge, X., Sarkar, S., and Murphy-Hill, E. (2014). Towards refactoring-aware code review. In *7th International Workshop on Cooperative and Human Aspects of Software Engineering (CHASE)*, pages 99–102. ACM.
- Ge, X., Sarkar, S., Witschey, J., and Murphy-Hill, E. (2017). Refactoring-aware code review. In *Symposium on Visual Languages and Human-Centric Computing (VL/HCC)*, pages 71–79.
- Goedert, L. and Arndt, K. B. F. (2020). Mediação pedagógica e educação mediada por tecnologias digitais em tempos de pandemia. *Criar Educação*, 9(2):104–121.
- Hayashi, S., Thangthumachit, S., and Saeki, M. (2013). Rediffs: Refactoring-aware difference viewer for Java. In *20th Working Conference on Reverse Engineering (WCRE)*, pages 487–488.

- Hora, A. (2021). Googling for software development: What developers search for and what they find. In *18th International Conference on Mining Software Repositories (MSR)*, pages 317–328.
- Le, L. H. and Hancer, M. (2021). Using social learning theory in examining youtube viewers’ desire to imitate travel vloggers. *Journal of Hospitality and Tourism Technology*, 12(3):512–532.
- Lee, C. S., Osop, H., Goh, D. H.-L., and Kelni, G. (2017). Making sense of comments on youtube educational videos: A self-directed learning perspective. *Online information review*, 41(5):611–625.
- Lima, C., de Moraes, P. H., and Hora, A. (2018). Um estudo em larga-escala sobre características de apis populares. 6th Brazilian Workshop on Software Visualization, Evolution and Maintenance (VEM), pages 1–8.
- Lopes, M. and Hora, A. (2022). How and why we end up with complex methods: A multi-language study. *Empirical Software Engineering*, 27:1–42.
- MacLeod, L., Storey, M.-A., and Bergen, A. (2015). Code, camera, action: How software developers document and share program knowledge using youtube. In *23rd International Conference on Program Comprehension*, pages 104–114.
- Meyerovich, L. A. and Rabkin, A. S. (2013). Empirical analysis of programming language adoption. In *International Conference on Object Oriented Programming Systems Languages & Applications (OOPSLA)*, page 1–18.
- Nawaz, S., Rizwan, M., and Rafiq, M. (2019). Recommendation of effectiveness of youtube video contents by qualitative sentiment analysis of its comments and replies. *Pakistan Journal of Science*, 71(4):91.
- Poché, E., Jha, N., Williams, G., Staten, J., Vesper, M., and Mahmoud, A. (2017). Analyzing user comments on youtube coding tutorial videos. In *25th International Conference on Program Comprehension (ICPC)*, pages 196–206.
- Ponzanelli, L., Bavota, G., Mocci, A., Di Penta, M., Oliveto, R., Hasan, M., Russo, B., Haiduc, S., and Lanza, M. (2016). Too long; didn’t watch! extracting relevant fragments from software development video tutorials. In *38th International Conference on Software Engineering (ICSE)*, page 261–272.
- Silva, D. and Valente, M. T. (2017). RefDiff: Detecting refactorings in version histories. In *14th International Conference on Mining Software Repositories (MSR)*, pages 269–279.
- Spadini, D., Aniche, M., and Bacchelli, A. (2018). PyDriller: Python framework for mining software repositories. In *26th Software Engineering Conference and Symposium on the Foundations of Software Engineering (FSE)*, pages 908–911.
- Storey, M.-A., Singer, L., Cleary, B., Figueira Filho, F., and Zagalsky, A. (2014). The (r) evolution of social media in software engineering. In *Future of software engineering*, pages 100–116.
- Sureka, A. (2011). Mining user comment activity for detecting forum spammers in youtube. *arXiv preprint arXiv:1103.5044*.

- Tian, Y., Nagappan, M., Lo, D., and Hassan, A. E. (2015). What are the characteristics of high-rated apps? a case study on free android applications. In *International Conference on Software Maintenance and Evolution (ICSME)*, pages 301–310.
- Tsantalis, N., Ketkar, A., and Dig, D. (2020). RefactoringMiner 2.0. *IEEE Transactions on Software Engineering (TSE)*, pages 1–22.
- Vánder, N., Antal, G., Hegedüs, P., and Ferenc, R. (2024). On the usefulness of python structural pattern matching: An empirical study. In *International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 501–511.
- Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. C., Regnell, B., and Wesslén, A. (2012). *Experimentation in software engineering*. Springer Science & Business Media.
- Xavier, L., Brito, A., Hora, A., and Valente, M. T. (2017). Historical and impact analysis of api breaking changes: A large-scale study. In *24th International Conference on Software Analysis, Evolution and Reengineering (SANER)*, pages 138–147.