

The Evolution of Cloud Ecosystems: How AWS, Azure, and GCP Transformed Between 2008 and 2025

Angelica dos Santos Arruda¹, Rodrigo Pereira dos Santos¹,
Pedro Nuno de Souza Moura¹, Adriana Cesário de Faria Alvim¹,
Laura de Oliveira Fernandes Moraes¹

¹Programa de Pós-Graduação em Informática (PPGI)
Universidade Federal do Estado do Rio de Janeiro (UNIRIO)
Avenida Pasteur, 458 – CEP 22.290-255 – Rio de Janeiro, RJ – Brasil

angelica.arruda@edu.unirio.br,

{rps, pedro.moura, adriana, laura}@uniriotec.br

Abstract. Research Context: Cloud computing has become foundational in the digital world, with AWS, Azure, and GCP leading dynamic ecosystems that enable rapid adoption of technologies such as infrastructure as code, MLOps pipelines, and generative AI. **Scientific and/or Practical Problem:** Frequent changes in API, services, and compliance requirements demand constant adaptation from developers. However, there is limited understanding of how communities respond to these evolving platforms over time. **Proposed Solution and/or Analysis:** We analyze cloud-related discussions on Stack Overflow from 2008 to 2025 to uncover how developer concerns evolve across adoption stages. **Related IS Theory:** This study draws on the socio-technical theory as an inspiration for analyzing digital platforms as evolving systems of interdependent actors. From this perspective, developer communities act as co-creators of platform knowledge and adaptability, shaping both technological trajectories and governance mechanisms. **Research Method:** We apply a reproducible method combining large-scale Q&A mining, topic modeling, and contextual trend analysis across three time windows. **Summary of Results:** We identify successive thematic waves in discussions, reflecting stages from IaaS (Infrastructure as a Service) and PaaS (Platform as a Service) provisioning to containers, serverless, and, more recently, data engineering and AI. These transitions mirror both platform evolution and developer needs. **Contributions and Impact to IS area:** Our findings offer theoretical and practical insights on developer engagement, documentation strategies, and platform governance. We also contribute a replicable methodology for longitudinal analysis of evolving digital ecosystems.

1. Introdução

O desenvolvimento de software moderno é impulsionado por plataformas de nuvem como Amazon Web Services (AWS), Microsoft Azure e Google Cloud Platform (GCP), que oferecem infraestrutura, ferramentas e serviços integrados. Formaram-se ao seu redor ecossistemas de software que articulam tecnologia, negócios, comunidades de desenvolvedores e órgãos normativos, moldando práticas contemporâneas de desenvolvimento e entrega de aplicações. Esses ambientes evoluem continuamente, refletindo avanços técnicos,

dinâmicas de mercado e pressões regulatórias. As arquiteturas de nuvem se consolidaram a partir do modelo de IaaS (*Infrastructure as a Service*, em português Infraestrutura como Serviço), que viabilizou o avanço de práticas como automação, *pipelines* de CI/CD (*Continuous Integration/Continuous Delivery*, em português Integração Contínua/Entrega Contínua), MLOps (*Machine Learning Operations*, em português Operações de Aprendizado de Máquina), e integração com IA (Inteligência Artificial) generativa.

Nesse contexto, as comunidades de desenvolvedores atuam como intermediárias estratégicas e instâncias de governança informal. Fóruns como Stack Overflow funcionam como repositórios vivos dessa evolução, registrando desafios e soluções ao longo das ondas de inovação. A análise dessas interações permite mapear o amadurecimento das plataformas e a consolidação de novas práticas. Neste trabalho, investigamos a evolução dos tópicos discutidos sobre AWS, Microsoft Azure e GCP no Stack Overflow entre 2008 e 2025, com o objetivo de identificar padrões temáticos, compreender a emergência e estabilização de tecnologias e explorar as dinâmicas desses ecossistemas. A pesquisa contribui para o entendimento da coevolução entre tecnologia, comunidade e governança em plataformas de nuvem, respondendo à lacuna existente sobre como comunidades de desenvolvedores reagem às transformações tecnológicas dessas plataformas, especialmente na emergência de novos paradigmas e práticas de desenvolvimento.

Este trabalho está estruturado da seguinte forma: a Seção 2 apresenta a fundamentação teórica. A Seção 3 descreve o método de pesquisa. Na Seção 4, são discutidos os resultados. Por fim, a Seção 5 traz as considerações finais.

2. Fundamentação Teórica

Nesta seção, são apresentados conceitos de computação em nuvem e ecossistemas de nuvem, além do Stack Overflow como fonte de dados e alguns trabalhos relacionados.

2.1. Computação em Nuvem

Na literatura, a computação em nuvem é definida como um “modelo para permitir acesso conveniente e sob demanda a um conjunto compartilhado de recursos computacionais configuráveis (por exemplo, redes, servidores, armazenamento, aplicações e serviços)” [Mell e Grance, 2011]. Nesse paradigma, infraestrutura, plataforma e software são oferecidos como serviços modulares, acessíveis por meio de API (*Application Programming Interface*, em português Interface de Programação de Aplicações) e painéis de gerenciamento, apoiando o desenvolvimento ágil e a escalabilidade dinâmica de aplicações [Buyya, Yeo et al. 2009]. Ao abstrair a complexidade da infraestrutura, a nuvem permite que desenvolvedores se concentrem na lógica de negócio, acelerando ciclos de entrega e incentivando a experimentação contínua. Sob essa ótica, distinguem-se dois eixos de classificação das plataformas de nuvem: modelos de serviço e modelos de implantação.

Os modelos de serviço se dividem em três categorias. **IaaS** (*Infrastructure as a Service*, em português Infraestrutura como Serviço) provê recursos básicos de computação, como máquinas virtuais, armazenamento e redes, permitindo ao usuário instalar sistemas operacionais e *middleware* [Armbrust et al. 2010]. **PaaS** (*Platform as a Service*, em português Plataforma como Serviço) oferece ambientes pré-configurados para desenvolvimento, teste e implantação de aplicações, abstraindo a infraestrutura subjacente [Q. Zhang, Cheng e Boutaba, 2010]. Por fim, **SaaS** (*Software as a Service*, em

português Software como Serviço) disponibiliza produtos de software completos via navegador ou API, sem necessidade de instalação local [Marston et al. 2011].

Quanto aos modelos de implantação, a **nuvem privada** é dedicada a uma única organização, oferecendo maior controle, porém com custos e complexidade elevados [Rimal, Choi e Lumb, 2009]. A **nuvem híbrida** combina nuvens públicas e privadas para equilibrar segurança, flexibilidade e eficiência [Buyya, Broberg e Goscinski, 2011]. Por fim, a **nuvem pública** é operada por provedores externos, baseada em recursos compartilhados e no modelo *pay-as-you-go*, oferecendo alta escalabilidade com menor granularidade de controle sobre rede e segurança [Popović e Hocenski, 2010; Q. Zhang, Cheng e Boutaba, 2010]. A compreensão desses modelos fundamenta a análise das plataformas de nuvem como sistemas sociotécnicos em evolução. AWS, Microsoft Azure e GCP detêm aproximadamente 66% do mercado global de IaaS/PaaS [Gartner, 2023], diferenciando-se pela amplitude de serviços e maturidade tecnológica.

2.2. Ecossistemas de Nuvem

Ecossistemas de software são sistemas sociotécnicos compostos por uma plataforma tecnológica comum e atores interdependentes, como empresas, desenvolvedores, usuários e integradores, que interagem de forma colaborativa e competitiva [Messerschmitt e Szyperski, 2003]. Apresentam coevolução entre participantes, dependência técnica e econômica da plataforma e mecanismos próprios de governança [Barbosa et al. 2013]. A sua saúde depende do equilíbrio entre diversidade, interoperabilidade e alinhamento estratégico [Manikas e Hansen, 2013]. Em plataformas de nuvem, o conceito envolve:

- **Componentes:** abrangem de infraestrutura (IaaS) e ambientes de desenvolvimento e execução (PaaS) a aplicações completas, API e serviços de suporte, incluindo funções *serverless*, bancos de dados gerenciados, *pipelines* de *Big Data* e *frameworks* de IA [Deng, D. Chen e Yao, 2015];
- **Atores:** incluem *keystones* (organizações centrais), integradores, desenvolvedores, consultorias e usuários finais, interagindo em fluxos de valor que combinam cooperação/competição e participação de mercado [Barbosa et al. 2013];
- **Relações:** são estruturadas por contratos técnicos (padrões de API e Acordo de Nível de Serviço) e comerciais, formando um ambiente de governança adaptável a novas demandas e regulações [Wareham, Fox e Cano Giner, 2014].

A geração de valor nesses ecossistemas envolve a articulação entre *demand side* (usuários/empresas), *supply side* (fornecedores de infraestrutura e software) e agentes de suporte, cuja coordenação expande a cadeia de valor à medida que novas ofertas e parceiros são incorporados [Manikas e Hansen, 2013]. Assim, plataformas de nuvem configuram ecossistemas em transformação contínua, caracterizados por ciclos de inovação e adaptação entre tecnologia, mercado e atores envolvidos.

2.3. Stack Overflow como Fonte de Dados

A mineração de perguntas consiste na extração de conhecimento útil a partir de repositórios de perguntas e respostas, como Stack Overflow, permitindo identificar padrões, temas recorrentes e dificuldades enfrentadas por desenvolvedores. Stack Overflow é amplamente reconhecido como uma fonte rica para análise do comportamento de desenvolvedores, reunindo milhões de interações que refletem dúvidas e práticas em constante

evolução [Barua, Thomas e Hassan, 2014]. Estudos demonstram que a mineração desses dados permite acompanhar desafios técnicos e sua evolução temporal [Bajaj, Pattabiraman e Mesbah, 2014; Fontão et al. 2018]. Pesquisas recentes, como [Zolduoarrati, Licorish e Stanger, 2024], reforçam sua relevância para investigar tendências, diversidade e lacunas de conhecimento em comunidades técnicas.

2.4. Trabalhos Relacionados

Na literatura, estudos têm abordado a evolução da computação em nuvem sob múltiplas perspectivas. O estudo [G. Zhang, Y. Chen e Li, 2020] realizou uma análise bibliométrica de 668 artigos publicados entre 2011 e 2019, evidenciando a transição do foco em questões técnicas (segurança, risco, custo e desempenho) para a integração com tecnologias emergentes, como IA, Big Data e Internet industrial. Já o estudo [Sharma e Srivastava, 2016] conduziu uma análise histórica e conceitual, discutindo a trajetória desde *mainframes* e virtualização até os modelos SaaS, PaaS e IaaS, consolidando a nuvem como revolução estrutural na TI, com forte impacto em pequenas e médias empresas.

O estudo [Gill et al. 2019] explorou como paradigmas emergentes, em especial Internet das Coisas, *Blockchain* e IA, vêm remodelando a nuvem em direção a sistemas mais inteligentes e distribuídos. Os autores propõem um modelo de “futurologia da nuvem”, que sintetiza tendências e aponta desafios em escalabilidade energética, qualidade de serviço e governança. O estudo [Alzide, 2024] analisa o papel da nuvem como base da transformação digital em setores como saúde, educação e finanças, destacando tendências atuais como *edge computing*, arquiteturas *serverless* e estratégias *multi-cloud*.

Diferentemente desses trabalhos, em particular do estudo [Alzide, 2024] que adota uma abordagem mais conceitual e prospectiva sobre tendências da computação em nuvem, este trabalho investiga a evolução desses ecossistemas a partir de dados extraídos das interações reais de desenvolvedores no Stack Overflow. Ao analisar longitudinalmente como dúvidas, práticas e soluções coletivas emergem ao longo do tempo, o estudo evidencia como as comunidades técnicas refletem, e por vezes antecipam, ondas de inovação em ecossistemas de software. Essa abordagem contribui ao oferecer uma análise orientada por dados sobre a coevolução entre plataformas e comunidades de usuários, além de disponibilizar um método reprodutível aplicável a outros contextos de inovação tecnológica.

3. Método de Pesquisa

O método adotado foi inspirado em [Fontão et al. 2018], cujo estudo analisou questões técnicas para apoiar a governança de plataformas de software móveis. O estudo serviu de referência para as etapas de pré-processamento e filtragem dos dados, além da aplicação de modelagem de tópicos para identificar padrões de interesse entre desenvolvedores ao longo do tempo. Em especial, seguimos princípios de [Fontão et al. 2018], estruturando as questões de pesquisa com base na abordagem GQM (*Goal-Question-Metric*), que define metas analíticas, questões de pesquisa (QP) e métricas para coleta e análise dos dados.

3.1. Objetivo do Estudo e Questões de Pesquisa

O estudo investiga a evolução dos tópicos discutidos por desenvolvedores nos ecossistemas de nuvem, analisando como essas discussões refletem a dinâmica de desenvolvimento de software, onde práticas, ferramentas e arquiteturas se consolidam ou se transformam

ao longo do tempo. A pesquisa está estruturada no modelo GQM e concentra-se em três eixos: (i) intensidade da atividade de desenvolvedores, (ii) análise de *tags*, destacando serviços centrais e novas tecnologias e (iii) identificação de padrões emergentes via algoritmo *Latent Dirichlet Allocation* (LDA).

Objetivo 1: Analisar como a atividade dos desenvolvedores evolui em relação às discussões sobre computação em nuvem.

- **QP1:** Como evoluíram as discussões sobre computação em nuvem entre desenvolvedores na plataforma Stack Overflow ao longo do tempo?

Justificativa: A resposta para esta QP permite analisar a trajetória temporal das interações, revelando períodos de maior ou menor engajamento e possíveis correlações com a introdução de novas tecnologias ou serviços em nuvem.

Métrica: Frequência anual de perguntas relacionadas a cada ecossistema (AWS, Azure e GCP), volume de questões ao longo do tempo e variações nos períodos de crescimento ou declínio.

Objetivo 2: Analisar os tópicos emergentes e a evolução das práticas de desenvolvimento de software nos ecossistemas de nuvem.

- **QP2:** Quais tópicos técnicos emergem das perguntas de desenvolvedores relacionadas à computação em nuvem, e como esses tópicos variaram ao longo do tempo por períodos distintos?

Justificativa: A resposta para esta QP evidencia como práticas, ferramentas e arquiteturas se consolidam ou se transformam, revelando a dinâmica de desenvolvimento de software em cada ecossistema.

Métricas: Distribuição dos tópicos LDA por período, prevalência relativa dos temas e identificação de tópicos emergentes.

Objetivo 3: Identificar os principais temas técnicos, tanto centrais quanto emergentes, presentes nas discussões sobre computação em nuvem.

- **QP3.1:** Quais são as principais *tags* (temas centrais) discutidas por desenvolvedores em cada ecossistema de nuvem (AWS, Azure e GCP), e como sua popularidade variou ao longo do tempo?

Justificativa: A resposta para esta sub-QP revela as tecnologias, bibliotecas, ferramentas ou arquiteturas mais recorrentes em cada janela temporal, oferecendo uma visão da evolução temática nos ecossistemas de nuvem, assim como quais serviços e tecnologias permanecem relevantes em cada um, funcionando como indicadores de maturidade ou estabilidade técnica.

Métrica: Frequência das 10 *tags* mais populares por período e variação de popularidade ao longo do tempo.

- **QP3.2:** Quais novas *tags* surgiram em cada ecossistema ao longo dos períodos analisados, e como refletem a introdução de novos serviços, tecnologias ou paradigmas?

Justificativa: Essa sub-QP ajuda a identificar ondas de inovação e a evolução do portfólio de serviços ofertados pelas plataformas de nuvem, buscando capturar a entrada de termos que não estavam presentes na janela anterior, mas que passaram a ter frequência mínima significativa na atual.

Métrica: Frequência de novas *tags* por período e comparação com *tags* históricas.

3.2. Processamento e Análise dos Dados

Os dados utilizados neste estudo foram extraídos dos *dumps* públicos do Stack Exchange, disponibilizados sob a licença Creative Commons CC BY-SA 3.0, e acessados por meio do repositório oficial da plataforma [Stack Exchange, 2025]. Para esta pesquisa, foi utilizado o *dump* disponível em abril de 2025, correspondente a março de 2025 [Stack Exchange, 2025]. Esses arquivos representam o histórico completo da atividade da plataforma e contêm dados desde seu início até a data do *dump*.

Para identificar e extrair apenas registros relevantes de computação em nuvem, foi desenvolvido um processo automatizado de leitura, filtragem e exportação em blocos. Este estudo se concentra nas plataformas AWS, Azure e GCP devido à sua dominância no mercado global de computação em nuvem e à disponibilidade de volume longitudinal de dados suficiente para sustentar análises comparativas robustas por meio de modelagem de tópicos. A filtragem foi realizada a partir de um conjunto estruturado de termos associados aos provedores analisados, incluindo nomes das plataformas, serviços centrais e variações terminológicas comumente utilizadas pela comunidade de desenvolvedores. Foram selecionadas apenas perguntas cujo título, corpo do texto ou *tags* contivessem ao menos um desses termos. A lista completa dos termos utilizados, bem como os *scripts* de filtragem está disponível em: <https://doi.org/10.5281/zenodo.17296393>.

A modelagem de tópicos foi realizada com o algoritmo LDA, proposto em [Blei, Ng e Jordan, 2003], utilizado para identificar padrões textuais em grandes conjuntos de textos. O modelo foi aplicado sobre perguntas e respostas do Stack Overflow relacionadas a AWS, Azure e GCP, seguindo práticas semelhantes às de estudos anteriores que analisaram a interação entre desenvolvedores nessas comunidades [Wang, Lo e Jiang, 2013].

O pré-processamento incluiu limpeza de HTML, URL e pontuações, além de lematização com o modelo `en_core_web_sm` do spaCy. Foram removidas *stop words* padrão e termos genéricos (`aws`, `azure`, `gcp` e `cloud`) para garantir que os tópicos refletissem questões técnicas, e não apenas nomes de provedores. A vetorização foi feita com `CountVectorizer` (`max_df = 0.75`, `min_df = 5`). O modelo LDA foi configurado com 4 tópicos por período, `max_iter = 20` e `learning_method = 'batch'`, seguindo recomendações da literatura [Treude e Wagner, 2019].

A qualidade dos tópicos foi avaliada por meio da coerência semântica `C_v`, calculada com o `CoherenceModel` da biblioteca Gensim, métrica amplamente utilizada por favorecer maior interpretabilidade [Muthusami et al. 2024]. Adicionalmente, os tópicos foram analisados qualitativamente a partir da interpretação contextual das palavras predominantes e sua associação com práticas tecnológicas discutidas pelos desenvolvedores ao longo dos períodos analisados. Os parâmetros do modelo foram definidos com base na literatura e priorizando a comparabilidade temporal entre plataformas e períodos, reconhecendo-se que análises de sensibilidade constituem uma possibilidade para trabalhos futuros. Eventuais termos residuais após o pré-processamento foram tratados considerando o contexto semântico dos tópicos, minimizando o impacto de artefatos lexicais.

4. Resultados e Discussão

Esta seção apresenta as respostas para as QP bem como a discussão dos resultados.

4.1. QP1: Como evoluíram as discussões sobre computação em nuvem entre desenvolvedores na plataforma Stack Overflow ao longo do tempo?

A Figura 1 apresenta a evolução anual no número absoluto de novas perguntas associadas às plataformas AWS, Azure e GCP no Stack Overflow, no período de 2008 a 2024 (completo). De maneira geral, observa-se um padrão de crescimento no número de perguntas, sustentado até 2019, seguido por uma queda a partir de 2020, comum a todas as plataformas analisadas. A plataforma AWS, lançada em 2006 com produtos como Amazon S3 (*Simple Storage Service*) e EC2 (*Elastic Compute Cloud*) e pioneira no modelo de infraestrutura como serviço, apresentou um crescimento sutil no número de perguntas nos primeiros anos da série, refletindo uma adoção ainda gradual da computação em nuvem. A partir de 2013, a curva se torna mais acentuada, marcando o início de uma fase de expansão significativa das *clouds*. Entre 2015 e 2020, observa-se um crescimento ainda mais expressivo, culminando em 2020 com mais de 35 mil novas perguntas registradas, o maior volume anual da série histórica. A partir de 2021, nota-se uma redução progressiva.

No caso da Azure, cuja versão oficial foi lançada em 2010 após um período de testes iniciado em 2008, sob o nome Windows Azure, identifica-se um padrão de crescimento semelhante ao da AWS. Após um início discreto, o número de perguntas passou a crescer de forma mais consistente a partir de 2012, ano em que a plataforma foi rebatizada como Microsoft Azure, refletindo sua ampliação para além do sistema operacional Windows. Azure, representada pela linha laranja, parte de uma participação próxima a 30% em 2008 e cresce gradualmente até 2012. A partir de aproximadamente 2014, observa-se que sua participação passa a se manter em patamar bastante próximo ao da AWS.

GCP, que contava com serviços de nuvem isolados nos primeiros anos e passou a operar como uma plataforma estruturada a partir de 2011, apresenta um crescimento nas discussões mais gradual e tardio. Até aproximadamente 2013, seu volume de participação ainda é residual, mas a partir de 2014, período em que GCP passou a incorporar tecnologias populares como o recém-adquirido Firebase, o lançamento do Kubernetes e a consolidação do BigQuery, observa-se um aumento significativo nas interações relacionadas à plataforma. O crescimento se mantém constante até atingir o pico em 2020, com aproximadamente 25 mil novas perguntas. Como as demais, GCP entra em trajetória de queda a partir de 2021, mantendo um volume absoluto inferior ao das concorrentes.

Embora esse declínio a partir de 2020 possa ser parcialmente atribuído ao amadurecimento dos ecossistemas, há indícios de que também reflete uma mudança estrutural nos canais preferidos pelos desenvolvedores. O estudo [Burtch, Lee e Z. Chen, 2024] mostra que a IA generativa têm sido progressivamente adotada como ambientes principais para a busca de conhecimento técnico, oferecendo respostas mais ágeis e contextualizadas que as encontradas no Stack Overflow. Essa migração, somada à melhoria da documentação oficial e ao uso crescente de ferramentas baseadas em IA generativa, como assistentes de código e modelos de linguagem natural, contribui para a retração do volume de interações públicas em fóruns tradicionais de perguntas e respostas.

A Figura 2 apresenta a proporção percentual anual de perguntas por plataforma em relação ao total entre 2008 e 2024. Isso permite observar a representatividade de cada plataforma, independentemente do volume total de perguntas em cada ano. AWS, pela linha azul, destaca seu pioneirismo, com predominância absoluta nos anos iniciais: em 2008, a plataforma concentrava mais de 60% das perguntas. Contudo, esse valor cai

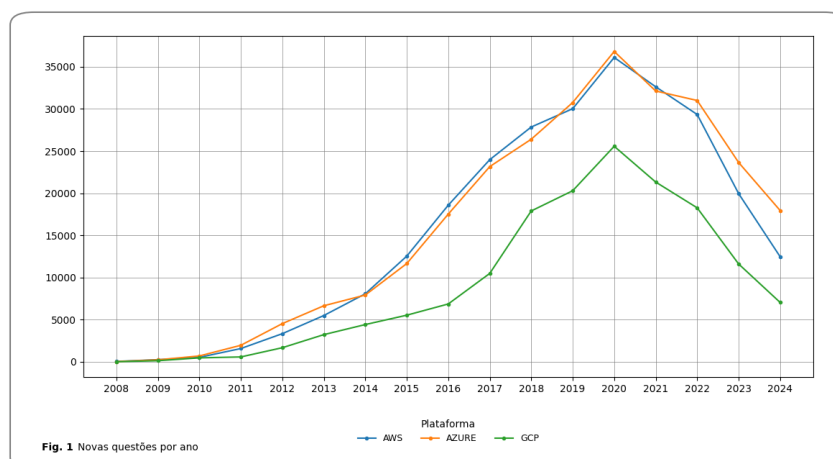


Figura 1. Evolução anual no número absoluto de perguntas por plataforma (2008–2024).

abruptamente até 2010, estabilizando-se em torno de 35% a 40% nas décadas seguintes. A partir de 2012, AWS apresenta leve oscilação, mas sua participação relativa permanece consideravelmente estável até 2022, com um leve declínio nos anos recentes.

Azure, pela linha laranja, parte de uma participação próxima a 30% em 2008 e cresceu gradualmente até 2012. A partir de então, mantém-se em patamar muito próximo à AWS, oscilando entre 35% e 45% por vários anos. Em 2024, atinge o valor mais alto da série, ultrapassando as demais plataformas e chegando a cerca de 47% da participação anual. GCP, pela linha verde, inicia o período com uma presença quase residual (cerca de 8% em 2008), crescendo rapidamente até 2010. Nos anos seguintes, sua participação oscila entre 20% e 30%, mantendo-se em um patamar modesto, porém estável. GCP atinge seus melhores índices relativos entre 2018 e 2020, com posterior declínio até 2024.

Esses resultados sugerem diferenças estruturais entre os ecossistemas analisados. Enquanto AWS apresenta trajetória marcada pelo pioneirismo e estabilidade relativa ao longo do tempo, Azure demonstra crescimento associado à forte integração com ambientes corporativos. Por sua vez, GCP evidencia expansão mais tardia, porém fortemente associada à adoção de tecnologias voltadas à engenharia de dados e aplicações modernas. Esses padrões indicam que a evolução da atividade dos desenvolvedores reflete não apenas a maturidade tecnológica das plataformas, mas também suas estratégias de posicionamento no ecossistema de nuvem.

4.2. QP2: Quais tópicos técnicos emergem das perguntas de desenvolvedores relacionadas à computação em nuvem, e como esses tópicos variaram ao longo do tempo por períodos distintos?

LDA representa uma técnica de modelagem de tópicos para texto. Neste trabalho, LDA permite agrupar palavras frequentes em tópicos que refletem os principais assuntos das perguntas. Isso é útil para entender a evolução das discussões técnicas, mostrando quais tópicos se tornam mais ou menos relevantes ao longo do tempo. Por isso, aplicamos LDA em diferentes períodos de tempo para cada plataforma, e, a partir destes tópicos, buscamos classificá-los em temas comuns.

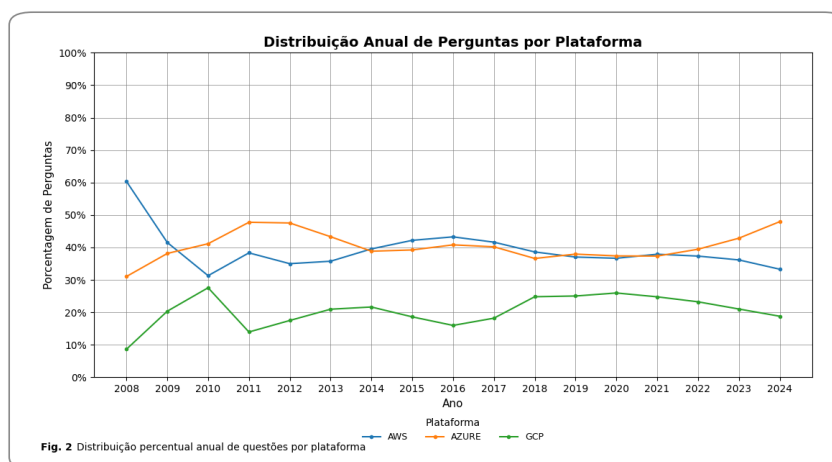


Figura 2. Participação percentual anual das plataformas de nuvem nas perguntas do Stack Overflow (2008–2024).

AWS. Com base nos tópicos apresentados na Tabela 1 para a plataforma AWS, no período inicial (2008–2010), é possível observar termos como *store*, *access*, *database*, *host*, *service* e *instance*, que indicam um foco predominante em infraestrutura e hospedagem. Esses tópicos sugerem uma fase de transição, em que as práticas ainda estavam fortemente baseadas no gerenciamento tradicional de servidores e bancos de dados, exigindo que os desenvolvedores aprendessem a lidar com o modelo IaaS, abstraindo a infraestrutura física por meio de serviços como EC2 e S3. As discussões dessa fase refletem um esforço inicial de compreensão e estabilização dos recursos básicos de nuvem.

No período de 2017 a 2019, os tópicos apontam para uma transformação significativa, com destaque para *docker*, *lambda*, *dependency* e *springframework*. Esses termos estão fortemente relacionados a tecnologias como containerização, funções serverless e infraestrutura como código temas centrais em arquiteturas modernas baseadas em microsserviços e *pipelines* de entrega contínua. Os tópicos indicam que os desenvolvedores começaram a lidar com desafios relacionados à automação de ambientes, integração de serviços e escalabilidade dinâmica.

Entre 2023 e 2025, observa-se um novo amadurecimento do ecossistema, em que o *pipeline* de *build*/empacotamento convive com processamento distribuído (*spark*) e ainda com funções serverless (*lambda*). A presença desses termos sugere que a comunidade da AWS está lidando com arquiteturas mais sofisticadas do que no período inicial, com preocupações relacionadas à interoperabilidade, segurança e governança. Esses tópicos descrevem uma evolução das dúvidas sobre “como subir e operar” para “como automatizar, decompor e orquestrar” aplicações e dados em escala.

Microsoft Azure. Conforme a Tabela 2, no período inicial (2008–2010), os tópicos *storage*, *blob*, *database*, *wcf* e *role* indicam que as discussões giravam em torno da configuração e do consumo de serviços de armazenamento, bancos de dados e componentes de aplicação em um ambiente fortemente vinculado ao ecossistema .NET. As questões levantadas refletem o esforço de migrar e adaptar aplicações corporativas tradicionais para o modelo PaaS, explorando *web services* e integrações básicas.

Tabela 1. Principais tópicos extraídos por LDA para AWS.

Área	2008–2010	2017–2019	2023–2025
Infraestrutura Hospedagem	store, access, database, host, application, look, datum, service, web, instance	host, local, docker, application, gem, command, lib, create, error, instance	host, log, connection, group, port, connect, error, create, service, instance
Armazenamento Buckets	instance, test, end, product, parclip, video, create, photo, upload, bucket	table, resource, log, bucket, error, api, create, datum, function, lambda	bucket, role, build, line, package, var, lambda, arn, resource, error
Desenvolvimento Frameworks	bundle, require, config, rake, bundler, local, rail, ruby, lib, gem	dependency, spring, springframework, info, version, spark, jar, apache, org, java	import, response, lambda, request, datum, function, string, return, const, error
Integração e API	follow, org, end, amazonaw, class, signature, java, request, string, error	function, log, err, upload, string, import, var, return, request, error	artifactid, groupid, dependency, info, jar, spark, version, apache, org, java

Tabela 2. Principais tópicos extraídos por LDA para Azure.

Área	2008–2010	2017–2019	2023–2025
Infraestrutura Hospedagem	dll, project, exe, role, net, exception, message, error, microsoft, web	bot, project, error, api, task, version, application, net, web, microsoft	true, deploy, test, project, package, task, version, error, pipeline, build
Armazenamento Bancos de Dados	grid, object, var, property, table, query, storage, blob, public, string	resource, test, microsoft, database, sql, datum, build, error, service, create	query, database, service, microsoft, sql, table, resource, error, datum, create
Desenvolvimento Frameworks	entity, create, store, option, model, connection, table, datum, database, sql	package, info, node, version, lib, apache, service, error, org, java	authentication, login, client, error, application, microsoft, api, org, token, java
Integração e API	asp, client, wcf, host, role, window, net, web, application, service	client, null, error, request, return, message, function, public, var, string	request, public, log, return, var, error, message, microsoft, string, function

Entre 2017 e 2019, tópicos como *bot*, *api* e *node* mostram que as dúvidas passaram a abordar integração contínua, automação de processos e *frameworks* multiplataforma, além de arquiteturas orientadas a eventos. Nesse período, as discussões técnicas evidenciam maior ênfase na orquestração de serviços, criação de *pipelines* de entrega contínua e práticas DevOps. De 2023 a 2025, tópicos como *authentication*, *login*, *token*, *pipeline* e *build* revelam que as questões estão concentradas em segurança, identidade e automação de fluxos complexos, destacando a relevância de governança de acesso e controle de API.

GCP. Na Tabela 3, no período inicial (2008–2010), termos como *python*, *gae*, *datas-tore* e *entity* mostram que as discussões estavam associadas à construção de aplicações web a partir do Google App Engine, explorando modelos de persistência simplificados e abstraindo a infraestrutura física. Entre 2017 e 2019, termos como *beam*, *firestore*, *firebase* e *gcloud* indicam que as questões técnicas passaram a envolver *pipelines* de dados (*Beam/Dataflow*), automação via linha de comando (*gcloud*) e *backends serverless* (Firebase e Firestore), com foco em produtividade e escalabilidade. Já no período de 2023 a 2025, permanecem em destaque *beam*, *firebase*, *firestore*, *deploy* e *artifactid*, sugerindo discussões centradas em empacotamento automatizado (como com Maven), orquestração

Tabela 3. Principais tópicos extraídos por LDA para GCP.

Área	2008–2010	2017–2019	2023–2025
Infraestrutura Hospedagem	line, persistent, private, resource, org, string, content, public, appengine, java	return, void, import, class, line, package, public, lib, android, string	self, deploy, service, python, project, line, package, lib, build, error
Armazenamento Bancos de Dados	datastore, store, entity, list, engine, gae, application, python, property, datum	query, job, table, error, create, import, beam, bucket, storage, datum	message, create, datum, project, const, log, firebase, request, error, function
Desenvolvimento Frameworks	string, list, class, filter, child, parent, object, entity, query, datastore	gcloud, info, apache, version, api, service, error, project, org, java	spring, beam, artifactid, dependency, groupid, apache, version, android, org, java
Integração e API	result, true, response, write, query, request, stringproperty, class, model, self	document, console, log, return, firestore, datum, const, error, firebase, function	firebase, error, doc, string, return, collection, datum, document, firestore, const

de *pipelines* de dados e implementação contínua, confirmando um ecossistema orientado à produtividade e automação.

Síntese Comparativa De forma geral, os tópicos extraídos via LDA revelam trajetórias distintas entre os ecossistemas. Na AWS, observa-se uma evolução das discussões sobre infraestrutura básica para temas de automação, *serverless* e *Big Data*, refletindo a adoção de *pipelines* e *toolchains* baseados em Java. Na Azure, os tópicos indicam uma transição do *stack* corporativo .NET para práticas DevOps, com ênfase recente em segurança, autenticação e identidade. Por fim, na GCP, há uma continuidade no foco em engenharia de dados e *backends serverless*, com destaque para *Beam*, *Firestore* e *Firebase*.

QP3.1: Quais são as principais tags (temas centrais) discutidas por desenvolvedores em cada ecossistema de nuvem (AWS, Azure e GCP), e como sua popularidade variou ao longo do tempo?

As Figuras 3, 4 e 5 apresentam a evolução temporal das dez tags mais frequentes em postagens relacionadas a AWS, Azure e GCP, em cinco períodos distintos (2008–2011, 2012–2015, 2016–2019, 2020–2023 e 2024–2025). Nos anos iniciais (2008–2011), na AWS, observa-se a dominância de serviços fundamentais da nuvem, como *amazon-ec2* e *amazon-s3*. EC2 oferece servidores virtuais escaláveis sob demanda, enquanto S3 fornece armazenamento de objetos em larga escala, sendo ambos pilares da computação em nuvem desde o início da AWS. Esses serviços aparecem ao lado de linguagens populares à época, como *php*, *java* e *ruby-on-rails*, evidenciando o uso inicial da plataforma principalmente como infraestrutura para aplicações web. A partir de 2012, nota-se o crescimento de serviços gerenciados, como o *elastic-beanstalk*, que facilita a implantação automática de aplicações, além do surgimento de linguagens mais recentes como *node.js*, sinalizando a diversificação do ecossistema e a adoção de arquiteturas mais modernas.

Entre 2016 e 2019, destaca-se a ascensão da tag *aws-lambda*, que se torna uma das mais mencionadas. Esse serviço representa o modelo *serverless*, no qual os desenvolvedores executam trechos de código sem precisar gerenciar servidores diretamente, uma mudança significativa no paradigma de desenvolvimento em nuvem. Durante esse período, também ganham relevância ferramentas voltadas à automação e gerenciamento

de infraestrutura, como *aws-sdk* (kit de desenvolvimento de software oficial da AWS), *cloudformation* (serviço de infraestrutura como código para provisionamento automatizado) e *amazon-dynamodb* (banco de dados NoSQL gerenciado). As linguagens *python* e *node.js* continuam em crescimento, refletindo sua ampla adoção.

O período mais recente (2020–2023) confirma a consolidação do *aws-lambda* como principal tecnologia mencionada, ultrapassando os volumes nos anos anteriores. Há aumento expressivo no uso de *terraform*, ferramenta amplamente utilizada para definição de infraestrutura como código, e *docker*, tecnologia de containerização que permite empacotar aplicações com todas as suas dependências. No intervalo de 2024–2025, com dados parciais, *aws-lambda* se mantém na liderança, seguido de *amazon-s3*, *python* e *amazon-ec2*. Observa-se ainda o surgimento de novas *tags*, como *aws-amplify*, plataforma que simplifica o desenvolvimento de aplicações web e móveis, e *amazon-cognito*, serviço de autenticação e gerenciamento de usuários. A continuidade de ferramentas como *terraform* e *docker* reforça a maturidade de um ecossistema voltado à automação e ao desenvolvimento *fullstack*. Os dados, portanto, apontam para uma transição progressiva da AWS de provedora de infraestrutura básica para uma plataforma abrangente, que integra computação, automação, dados, autenticação e desenvolvimento moderno.

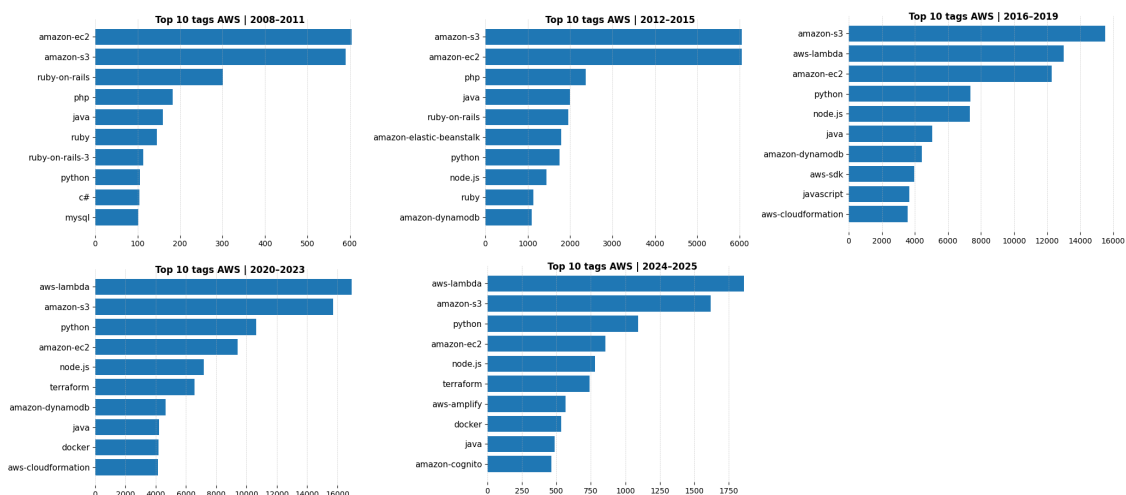


Figura 3. Top 10 tags por período para perguntas relacionadas a AWS.

A Figura 4 apresenta a evolução das dez tags mais mencionadas em postagens relacionadas à Azure. Observa-se o predomínio da linguagem *c#*, evidenciando a forte ligação inicial entre a Azure e o ecossistema *.NET*. Tags como *azure-sql-database*, *asp.net* e *azure-storage*, serviço de armazenamento de dados do Azure, indicam que os primeiros usos da plataforma estiveram centrados em soluções tradicionais de aplicações web hospedadas na nuvem, com foco em bancos de dados relacionais e armazenamento de arquivos. Tecnologias como *wcf* (Windows Communication Foundation) e *asp.net-mvc* reforçam esse padrão, ao passo que *cloud* e *windows* aparecem como conceitos genéricos associados ao modelo emergente de computação em nuvem da época.

No período de 2012 a 2015, *c#* permanece como linguagem central, mas nota-se o surgimento de novas tags representativas da ampliação da plataforma. A tag *azure-devops* aparece pela primeira vez, refletindo a introdução do antigo Visual Studio Online (posteriormente renomeado), que trouxe recursos integrados de versionamento, testes auto-

matizados e *pipelines* de integração contínua (CI/CD). Também ganham destaque *azure-web-app-service* e *azure-mobile-services*, plataformas que oferecem suporte gerenciado para a publicação de API, aplicações web e soluções móveis. A presença de *sql-server* e *azure-sql-database* reforça a continuidade do foco em persistência de dados relacional, enquanto *azure-storage* demonstra o crescimento do uso da Azure como *backend* para aplicações distribuídas e conectadas.

A partir de 2016, observa-se uma mudança na forma como a plataforma é utilizada. A *tag azure-devops* salta para quase a liderança, refletindo o avanço das práticas DevOps no ecossistema Microsoft. Neste mesmo período, destacam-se *azure-functions*, serviço *serverless* da Azure que permite a execução de código sob demanda sem necessidade de provisionamento de servidores, e *azure-active-directory*, solução para controle de identidade, autenticação federada e login único (*Single Sign-On*). O gráfico mostra ainda a entrada de *azure-cosmosdb*, banco de dados NoSQL multimodelo altamente escalável, e de *powershell*, linguagem de automação administrativa amplamente usada para orquestração de serviços em nuvem.

Essa tendência se intensifica entre 2020 e 2023, quando *azure-devops* atinge seu pico de popularidade, seguido por *azure-functions*, *azure-pipelines*, *azure-data-factory* (serviço de integração e transformação de dados) e *python*, que se consolida como linguagem versátil para automação e engenharia de dados. No intervalo mais recente (2024–2025), observa-se a continuidade da liderança de *azure-devops* e a permanência de *c#* e *azure-functions* entre as primeiras posições. A presença conjunta de *azure-pipelines*, *azure-data-factory*, *python* e *powershell* reforça a centralidade de fluxos automatizados de entrega de software, *pipelines* de dados e infraestrutura como código nas práticas atuais de desenvolvimento em nuvem. Já a permanência de *azure-web-app-service* e *.NET* no grupo das dez *tags* indica que, embora o ecossistema tenha se diversificado, a base legada do desenvolvimento Microsoft continua ativa.

Esses resultados indicam um amadurecimento progressivo da Azure. De uma plataforma voltada principalmente ao *hosting* de aplicações baseadas em *.NET*, transforma-se em um ambiente robusto de automação, integração contínua, processamento de dados e identidade gerenciada, adaptando-se às demandas contemporâneas de escalabilidade e modularidade no desenvolvimento em nuvem.

A Figura 5 apresenta a evolução das dez *tags* mais mencionadas em postagens relacionadas à plataforma GCP. Nos anos iniciais (2008–2011), destacam-se as *tags google-cloud-datastore*, banco de dados NoSQL gerenciado, e *google-app-engine*, serviço de PaaS para hospedagem de aplicações. Tais serviços aparecem associados a linguagens como *python* e *java*, indicando o uso inicial da GCP como uma plataforma voltada ao desenvolvimento de aplicações web escaláveis. A *tag jdo* (Java Data Objects) era uma API de persistência utilizada inicialmente no Google App Engine para aplicações em Java, permitindo o mapeamento de objetos para o *google-cloud-datastore*. Já *django*, um *framework* web escrito em Python, estava entre os poucos oficialmente suportados pela plataforma para aplicações hospedadas no App Engine, o que explica sua presença recorrente nas discussões técnicas. A *tag transactions*, por sua vez, se refere às operações atômicas suportadas pelo Datastore, que permitiam manter a consistência em gravações múltiplas de dados, mesmo em um banco NoSQL. Juntas, essas tecnologias indicam que, no período inicial da GCP, o foco estava na construção de aplicações web tradicionais

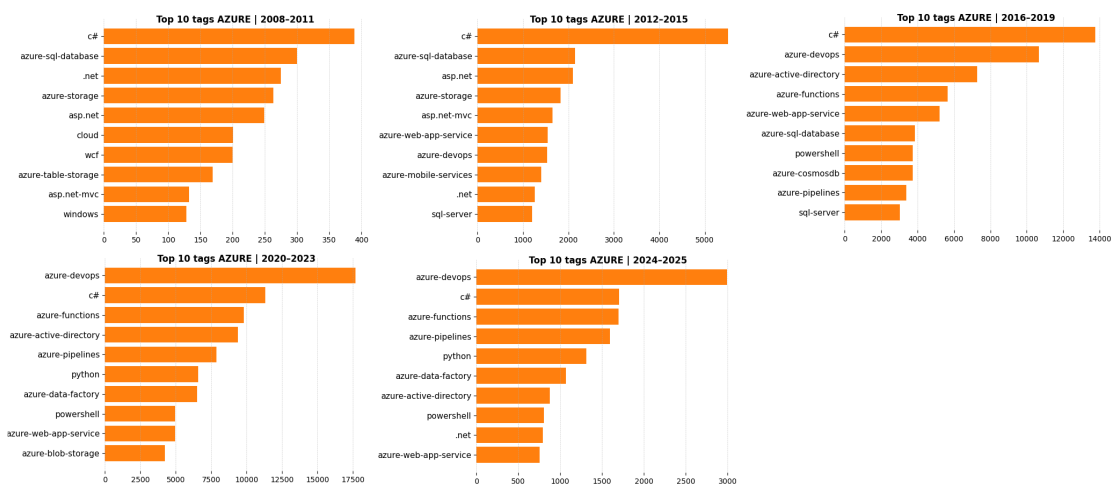


Figura 4. Top 10 tags por período para perguntas relacionadas a Azure.

com suporte a modelos de dados persistentes e ferramentas reconhecidas no ecossistema Python/Java, fortemente ancoradas na arquitetura imposta pelo App Engine.

No período de 2012 a 2015, *google-app-engine* se mantém como a principal *tag*, enquanto *android* e *google-cloud-messaging* assumem posições de destaque, revelando a integração crescente entre GCP e o desenvolvimento móvel. Outras *tags* importantes incluem *google-cloud-datastore*, *java* e *push-notification*, além de serviços como *google-cloud-endpoints* e *google-cloud-storage*, que indicam o uso da nuvem da Google como *backend* para aplicações móveis. Por sua vez, a partir de 2016, observa-se uma mudança significativa com a entrada das *tags firebase* e *google-cloud-firestore*, ambas ligadas ao ecossistema de desenvolvimento móvel e web da Google. Também ganha destaque o *google-cloud-functions*, serviço *serverless* para execução de funções *event-driven*. Linguagens e *frameworks* amplamente utilizados no desenvolvimento *frontend*, como *javascript* e *node.js*, passam a figurar com força crescente.

Entre 2020 e 2023, *google-cloud-firestore* e *firebase* se tornam ainda mais expressivos, ambos ultrapassando amplamente as demais *tags* em volume absoluto. Essas tecnologias são voltadas ao desenvolvimento de aplicações modernas com sincronização em tempo real, autenticação integrada e *backend* gerenciado. Neste mesmo intervalo, surge *flutter*, evidenciando o fortalecimento do ecossistema Google para aplicações multiplataforma. As *tags google-cloud-functions*, *node.js* e *python* também figuram entre as dez *tags* mais populares, consolidando o uso da GCP para soluções web e móveis com arquiteturas modernas. Já no período mais recente (2024–2025), observa-se a continuidade dessa tendência, com *google-cloud-firestore*, *firebase* e *flutter* com destaque. *Tags* associadas ao desenvolvimento *frontend* (*javascript*, *reactjs*) e *backend* (*node.js*, *google-cloud-functions*, *python*) reforçam a integração entre as diversas camadas do desenvolvimento moderno. Esses resultados indicam que GCP se consolidou, ao longo do tempo, como uma plataforma fortemente orientada a desenvolvedores, com ênfase em aplicações em tempo real, multiplataforma e com infraestrutura simplificada.

Em síntese, a análise das *tags* por período evidencia que o uso das três plataformas evoluiu de soluções focadas em infraestrutura básica para ecossistemas completos volta-

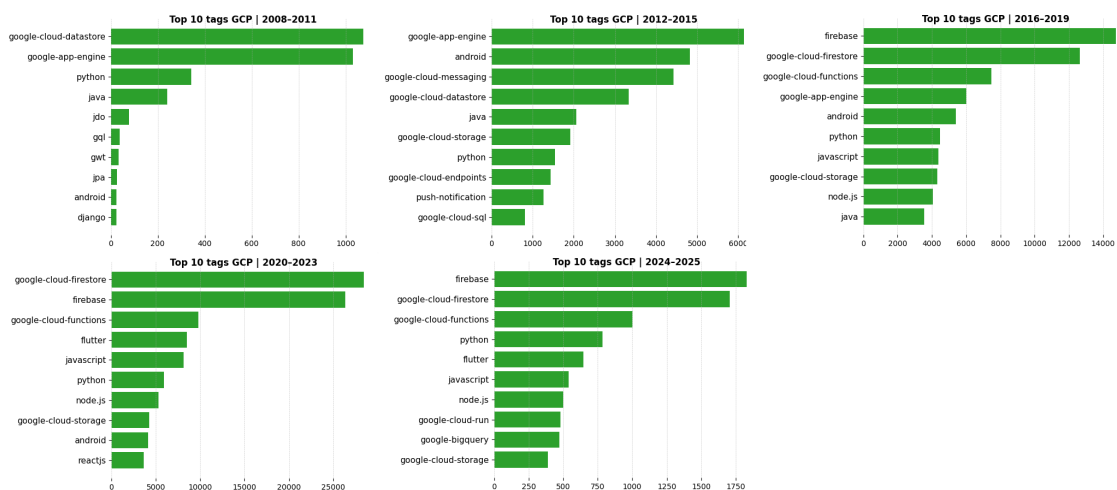


Figura 5. Top 10 tags por período para perguntas relacionadas a GCP.

dos à automação, dados e desenvolvimento moderno. AWS se destacou inicialmente com serviços fundamentais como *ec2* e *s3*, e, ao longo do tempo, se consolidou como uma plataforma versátil, com forte ênfase em escalabilidade, automação e diversidade de serviços, liderada atualmente por *aws-lambda* e ferramentas como *terraform*. Azure, por sua vez, partiu de uma base fortemente integrada ao .NET e se transformou em um ambiente corporativo robusto para integração contínua, gestão de identidade e *pipelines* de dados, com destaque para *azure-devops*, *azure-functions* e *azure-data-factory*. Por fim, GCP trilhou um caminho mais voltado à simplicidade no desenvolvimento, com forte adesão da comunidade móvel e web, consolidando-se com *firebase*, *google-cloud-firestore* e *flutter* como elementos centrais. Embora compartilhem tendências como *serverless* e DevOps, cada nuvem mantém identidade própria: AWS como infraestrutura generalista e altamente escalável; Azure como solução empresarial orientada à automação e dados; e GCP como plataforma leve e produtiva voltada a aplicações multiplataforma em tempo real.

QP3.2: Quais novas tags surgiram em cada ecossistema ao longo dos períodos analisados, e como elas refletem a introdução de novos serviços, tecnologias ou paradigmas?

O surgimento de *tags* no Stack Overflow ocorre por meio de um processo colaborativo que combina proposição espontânea, validação comunitária e adoção orgânica. Inicialmente, usuários com reputação igual ou superior a 1.500 pontos podem propor diretamente uma nova *tag* ao formular uma pergunta; já usuários com menor reputação têm suas sugestões encaminhadas a um painel de revisão. Em seguida, moderadores e membros experientes da comunidade avaliam a proposta quanto à relevância temática, clareza ortográfica e escopo semântico, com o objetivo de evitar ambiguidade ou duplicação com *tags* existentes. Uma vez aprovada, a *tag* passa a integrar o sistema com o suporte de um “tag wiki”, onde editores documentam sua definição, sinônimos associados, casos de uso e boas práticas.

Além da análise das tecnologias mais utilizadas em cada período discutidas anteriormente, a Tabela 4 apresenta uma perspectiva complementar ao destacar as *tags* emergentes. Essas *tags* não são, necessariamente, as mais frequentes, mas indicam o surgimento de novas tecnologias, práticas e ferramentas dentro de cada ecossistema.

Tabela 4. Novas tags por período e plataforma.

Período	AWS	GCP	Azure
2008–2010	—	—	—
2011–2013	node.js, amazon-dynamodb, amazon-elastic-beanstalk, ios, carrierwave	google-cloud-endpoints, python-2.7, app-engine-ndb, google-cloud-sql, android-c2dm	asp.net-mvc-4, azure-mobile-services, node.js, visual-studio-2012, acs
2014–2016	aws-api-gateway, ansible, amazon-cognito, swift, spring-boot	firebase-cloud-messaging, firebase, google-cloud-dataproc, apache-spark, kubernetes	azure-cosmosdb, azure-resource-manager, azure-application-insights, visual-studio-2015, asp.net-core
2017–2019	aws-amplify, aws-glue, aws-appsync, serverless, terraform-provider-aws	flutter, firebase-security, angularfire2, firebase-admin, vue.js	azure-aks, azure-databricks, databricks, azure-cosmosdb-sqlapi, asp.net-core-2.0
2020–2022	aws-glue-spark, opensearch, aws-databricks, mwaa, cicd	google-cloud-vertex-ai, flutter-streambuilder, vuejs3, google-secret-manager, flutter-futurebuilder	azure-bicep, .net-6.0, .net-5, azure-devops-pipelines, blazor-webassembly
2023–2025	amazon-bedrock, langchain, large-language-model, next.js13, .net-8.0	google-gemini, langchain, flutter-flow, large-language-model, google-cloud-colab-enterprise	azure-openai, .net-8.0, openai-api, azure-synapse-analytics, langchain

Em relação a AWS, observa-se que as *tags* emergentes iniciais estavam associadas ao armazenamento e bancos de dados não relacionais. Ao longo do tempo, surgem ferramentas voltadas à automação de infraestrutura, segurança e integração de dados, como AWS Glue, API Gateway e AWS AppSync, indicando aumento da complexidade dos sistemas. No período mais recente (2023–2025), destacam-se tecnologias ligadas à IA generativa, como Amazon Bedrock, LangChain e Large Language Models. Na GCP, as *tags* emergentes indicam ênfase no desenvolvimento de aplicações web e móveis, com destaque para o crescimento do Firebase a partir de 2014. Tecnologias como Flutter, Vertex AI, Gemini e FlutterFlow refletem a expansão da plataforma para aplicações inteligentes e ambientes de desenvolvimento moderno. Por fim, Azure apresenta trajetória fortemente conectada ao ambiente corporativo, com *tags* associadas à migração de sistemas .NET, gestão de infraestrutura e integração de sistemas complexos, incluindo Databricks, Pipelines e Application Insights. A partir de 2023, surgem soluções como Azure OpenAI, LangChain e Synapse Analytics, reforçando o foco em automação empresarial baseada em IA.

A distribuição temporal das *tags* emergentes revela ciclos de inovação tecnológica. Entre 2011 e 2013, surgem serviços voltados à mobilidade e API. Entre 2014 e 2016, destacam-se ferramentas de infraestrutura como código. A partir de 2017, ampliam-se tecnologias voltadas ao processamento em larga escala e arquiteturas orientadas a eventos. No período de 2020 a 2025, observa-se a ascensão da IA generativa (*amazon-bedrock*, *azure-openai*, *vertex-ai*, *langchain*) e de ferramentas de desenvolvimento rápido e visual (*flutterflow*, *colab-enterprise*), indicando um novo ciclo de inovação centrado em modelos de linguagem e plataformas *low-code*.

Os resultados convergem com a literatura ao indicarem que a computação em nuvem evolui em ondas temáticas. Em linha com o estudo [G. Zhang, Y. Chen e Li, 2020], identificamos fases sucessivas associadas à maturação tecnológica, com evidência mais granular da transição para contêineres e arquiteturas *serverless*. De forma semelhante ao estudo [Sharma e Srivastava, 2016], observamos que os debates acompanharam a consolidação dos modelos SaaS, PaaS e IaaS, incorporando desafios práticos de documentação e padronização. Em relação às projeções do estudo [Gill et al. 2019], nossos resultados confirmam a relevância de novas tecnologias, destacando o protagonismo

de serviços de dados e aprendizagem de máquina (*machine learning*). Por fim, em sintonia com o estudo [Alzide, 2024], evidenciamos que a transformação digital também se manifesta nas práticas cotidianas dos desenvolvedores, oferecendo uma análise experimental e longitudinal que complementa abordagens conceituais ao detalhar como comunidades técnicas vivenciam essas transições tecnológicas.

5. Considerações Finais

Este estudo reporta indícios de um padrão evolutivo nos ecossistemas de nuvem (AWS, Azure e GCP) ao longo de quase duas décadas de interações. A análise segmentada por períodos revelou “ondas temáticas” alinhadas aos ciclos de inovação tecnológica:

- **2008–2012:** Predomínio do provisionamento de máquinas virtuais, armazenamento e autenticação, refletindo a transição inicial para IaaS e dúvidas concentradas em disponibilidade e segurança;
- **2013–2016:** Consolidação de serviços gerenciados (PaaS) e orquestração por *templates*, com deslocamento do foco operacional para padronização de ambientes;
- **2017–2019:** Expansão da automação, containerização e *serverless*, incorporando infraestrutura como código e *pipelines* CI/CD como pilares da cultura DevOps;
- **2020–2022:** Avanço da engenharia de dados e MLOps, consolidando as nuvens como suporte a *workloads* analíticos e aprendizado de máquina;
- **2023–2025:** Integração com soluções *IA-driven*, RAG (*Retrieval-Augmented Generation*, em português Geração Aumentada por Recuperação) e preocupações com latência e governança de modelos, caracterizando a expansão de *copilots* e serviços cognitivos.

As análises também indicam trajetórias distintas entre as plataformas. AWS evoluiu de infraestrutura básica para um ecossistema orientado à automação e orquestração. Azure se consolidou como plataforma corporativa, com ênfase em identidade, *compliance* e governança. GCP se destacou pela produtividade do desenvolvedor, *serverless* e inovação em engenharia de dados e IA. Apesar dessas diferenças, ciclos como *serverless*, MLOps e IA generativa transcendem plataformas, impulsionando transformações contínuas nas práticas de desenvolvimento e estratégias dos fornecedores. Além disso, sob a perspectiva da teoria socio-técnica, os resultados mostram que a evolução das plataformas ocorre por reconfigurações contínuas de complementaridades, papéis e governança. As “ondas temáticas” observadas indicam que as interações comunitárias não são periféricas, mas parte central da adaptação e resiliência dos ecossistemas, refletindo transformações que emergem da interação entre tecnologia, provedores, comunidade, governança e regulação.

Este estudo apresenta limitações metodológicas e experimentais. A principal se refere ao uso do Stack Overflow como única fonte de dados. Embora historicamente central para a comunidade de desenvolvedores, a migração gradual para plataformas como Slack, Discord e GitHub Discussions pode limitar sua representatividade recente. Como mitigação, adotamos segmentação temporal para capturar tendências de longo prazo, reconhecendo a necessidade de integração de múltiplas fontes em estudos futuros. Outra limitação envolve o ruído inerente aos dados. Mesmo após pré-processamento e filtragem, podem persistir questões redundantes ou com baixo conteúdo técnico. Há ainda ameaças relacionadas à extração e interpretação, como inconsistências em metadados e mudanças

na plataforma ao longo do tempo. Para reduzir esses riscos, realizamos validações cruzadas entre períodos, verificação da integridade das extrações e uso de métricas de coerência dos tópicos, reconhecendo, contudo, que tais ameaças não podem ser completamente eliminadas. Novos estudos incluem o uso de modelos semânticos mais sofisticados, como BERTopic e Top2Vec, e replicações com dados de outras comunidades.

Trabalhos futuros podem expandir este estudo por meio da incorporação de modelos preditivos de tendências tecnológicas, análises prescritivas sobre mecanismos de governança e engajamento de desenvolvedores, bem como investigações sobre implicações organizacionais, como estratégias *multi-cloud*. Estudos com maior granularidade temporal ou orientados a eventos também podem aprofundar a compreensão sobre como anúncios tecnológicos influenciam a atividade das comunidades, contribuindo para explorar a teoria socio-técnica no contexto de computação em nuvem e IA.

6. Agradecimentos

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Código de Financiamento 001, CNPq (Proc. 316510/2023-8), FAPERJ (Proc. E-26/204.404/2024) e UNIRIO.

Declaração de uso de IA generativa: A primeira autora declara que utilizou ferramentas de IA generativa (ChatGPT, da OpenAI) exclusivamente para revisão textual e apoio na formatação, sem interferência na análise dos resultados ou nas conclusões do estudo.

Referências

- Alzide, Safana (2024). “Cloud Computing: Evolution, Challenges, and Future Prospects”. Em: *Journal of Information Technology, Cybersecurity, and Artificial Intelligence* 1.1, pp. 52–63. DOI: 10.70715/jitcai.2024.v1.i1.007.
- Armbrust, Michael et al. (2010). “A view of cloud computing”. Em: *Commun. ACM* 53.4, pp. 50–58. DOI: 10.1145/1721654.1721672.
- Bajaj, Kartik, Karthik Pattabiraman e Ali Mesbah (2014). “Mining questions asked by web developers”. Em: *Proceedings of the 11th Working Conference on Mining Software Repositories*. MSR 2014. Hyderabad, India: Association for Computing Machinery, pp. 112–121. DOI: 10.1145/2597073.2597083.
- Barbosa, Olavo et al. (2013). “A systematic mapping study on software ecosystems from a three-dimensional perspective”. Em: *Software Ecosystems*. Cheltenham, UK: Edward Elgar Publishing. DOI: 10.4337/9781781955628.00011.
- Barua, Anton, Stephen W. Thomas e Ahmed E. Hassan (2014). “What are developers talking about? An analysis of topics and trends in Stack Overflow”. Em: *Empirical Software Engineering* 19, pp. 619–654. DOI: 10.1007/s10664-012-9231-y.
- Blei, David M., Andrew Y. Ng e Michael I. Jordan (2003). “Latent dirichlet allocation”. Em: *J. Mach. Learn. Res.* 3, pp. 993–1022.
- Burtch, Gordon, Dokyun Lee e Zhichen Chen (2024). “The consequences of generative AI for online knowledge communities”. Em: *Scientific Reports* 14, p. 10413. DOI: 10.1038/s41598-024-61221-0.
- Buyya, Rajkumar, James Broberg e Andrzej Goscinski (2011). *Cloud Computing: Principles and Paradigms*. John Wiley & Sons. DOI: 10.1002/9780470940105.

- Buyya, Rajkumar, Chee Shin Yeo et al. (2009). “Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility”. Em: *Future Generation Computer Systems* 25.6, pp. 599–616. DOI: <https://doi.org/10.1016/j.future.2008.12.001>.
- Deng, Guohua, Donglin Chen e Mengdi Yao (2015). “Value structure analysis for cloud service ecosystem”. Em: *International Journal of Services Technology and Management* 21.4-6, pp. 228–237. DOI: [10.1504/IJSTM.2015.073922](https://doi.org/10.1504/IJSTM.2015.073922).
- Fontão, Awdren et al. (2018). “Supporting governance of mobile application developers from mining and analyzing technical questions in stack overflow”. Em: *Journal of Software Engineering Research and Development* 6, 8:1–8:34. DOI: [10.1186/s40411-018-0052-6](https://doi.org/10.1186/s40411-018-0052-6).
- Gartner (2023). *Magic Quadrant for Cloud Infrastructure and Platform Services*.
- Gill, Sukhpal Singh et al. (2019). “Transformative effects of IoT, Blockchain and Artificial Intelligence on cloud computing: Evolution, vision, trends and open challenges”. Em: *Internet of Things* 8, p. 100118. DOI: <https://doi.org/10.1016/j.iot.2019.100118>.
- Manikas, Konstantinos e Klaus Marius Hansen (2013). “Software ecosystems – A systematic literature review”. Em: *Journal of Systems and Software* 86.5, pp. 1294–1306. DOI: <https://doi.org/10.1016/j.jss.2012.12.026>.
- Marston, Sean et al. (2011). “Cloud computing — The business perspective”. Em: *Decision Support Systems* 51.1, pp. 176–189. DOI: <https://doi.org/10.1016/j.dss.2010.12.006>.
- Mell, Peter e Timothy Grance (2011). *The NIST Definition of Cloud Computing*. Rel. técn. NIST SP 800-145. National Institute of Standards and Technology.
- Messerschmitt, David G. e Clemens Szyperski (2003). *Software Ecosystem: Understanding an Indispensable Technology and Industry*. MIT Press. DOI: [10.7551/mitpress/6323.001.0001](https://doi.org/10.7551/mitpress/6323.001.0001).
- Muthusami, R et al. (2024). “Investigating topic modeling techniques through evaluation of topics discovered in short texts data across diverse domains”. Em: *Scientific Reports* 14.1, p. 12003. DOI: [10.1038/s41598-024-61738-4](https://doi.org/10.1038/s41598-024-61738-4).
- Popović, Krešimir e Željko Hocenski (2010). “Cloud computing security issues and challenges”. Em: *The 33rd International Convention MIPRO*, pp. 344–349.
- Rimal, Bhaskar Prasad, Eunmi Choi e Ian Lumb (2009). “A Taxonomy and Survey of Cloud Computing Systems”. Em: *2009 Fifth International Joint Conference on INC, IMS and IDC*, pp. 44–51. DOI: [10.1109/NCM.2009.218](https://doi.org/10.1109/NCM.2009.218).
- Sharma, Viney e Gur Mauj Saran Srivastava (2016). “Evolution and present status of cloud computing: a comprehensive analysis”. Em: *Int. J. Bus. Inf. Syst.* 22.2, pp. 123–142. DOI: [10.1504/IJBIS.2016.076243](https://doi.org/10.1504/IJBIS.2016.076243).
- Stack Exchange (2025). *Stack Exchange Data Dump - 2025-03-31*. https://archive.org/details/stackexchange_20250331. April 2025.
- Treude, Christoph e Markus Wagner (2019). “Predicting good configurations for GitHub and stack overflow topic models”. Em: *Proceedings of the 16th International Conference on Mining Software Repositories. MSR '19*. Montreal, Quebec, Canada: IEEE Press, pp. 84–95. DOI: [10.1109/MSR.2019.00022](https://doi.org/10.1109/MSR.2019.00022).
- Wang, Shaowei, David Lo e Lingxiao Jiang (2013). “An empirical study on developer interactions in StackOverflow”. Em: *Proceedings of the 28th Annual ACM Sympo-*

- sium on Applied Computing*. SAC '13. Coimbra, Portugal: Association for Computing Machinery, pp. 1019–1024. DOI: 10.1145/2480362.2480557.
- Wareham, Jonathan, Paul B. Fox e Josep Lluís Cano Giner (2014). “Technology Ecosystem Governance”. Em: *Organization Science* 25.4, pp. 1195–1215. DOI: 10.1287/orsc.2014.0895.
- Zhang, Ge, Yun Chen e Gaoyong Li (2020). “The Evolution and Emerging Trends of Cloud Computing Adoption Research: Visual Analysis of CiteSpace Based on WOS Papers”. Em: *Proceedings of the 2020 3rd International Conference on Signal Processing and Machine Learning*. SPML '20. Beijing, China: Association for Computing Machinery, pp. 40–47. DOI: 10.1145/3432291.3433641.
- Zhang, Qi, Lu Cheng e Raouf Boutaba (2010). “Cloud computing: state-of-the-art and research challenges”. Em: *Journal of Internet Services and Applications* 1, pp. 7–18. DOI: 10.1007/s13174-010-0007-6.
- Zolduoarrati, Elijah, Sherlock A. Licorish e Nigel Stanger (2024). “Harmonising Contributions: Exploring Diversity in Software Engineering through CQA Mining on Stack Overflow”. Em: *ACM Trans. Softw. Eng. Methodol.* 33.7, 179:1–179:54. DOI: 10.1145/3672453.