



# GitLab vs GitHub: Explore suas Principais Diferenças e Semelhanças

Quando se trata do GitLab vs GitHub, qual é a melhor plataforma de desenvolvimento para sua equipe em 2021? Qual é a melhor solução para o seu caso de uso?

Essas são as perguntas-chave que estamos respondendo hoje.

Tanto GitLab quanto GitHub são plataformas de desenvolvimento com um forte foco no sistema Git de código aberto para controle distribuído de versões. [87,2%](#) dos desenvolvedores entrevistados confiam em Git para o controle de versão, colaboração e implementação mais suave.

Git se tornou uma ferramenta fundamental para desenvolvedores em startups, empresas e até mesmo posições governamentais. Você também pode [usar Git em Kinsta](#).

É por isso que escolher a ferramenta certa para sua equipe é tão importante.

Vamos fazer um mergulho profundo nas diferenças do GitLab e GitHub, seus benefícios e desvantagens, e que você deve escolher.

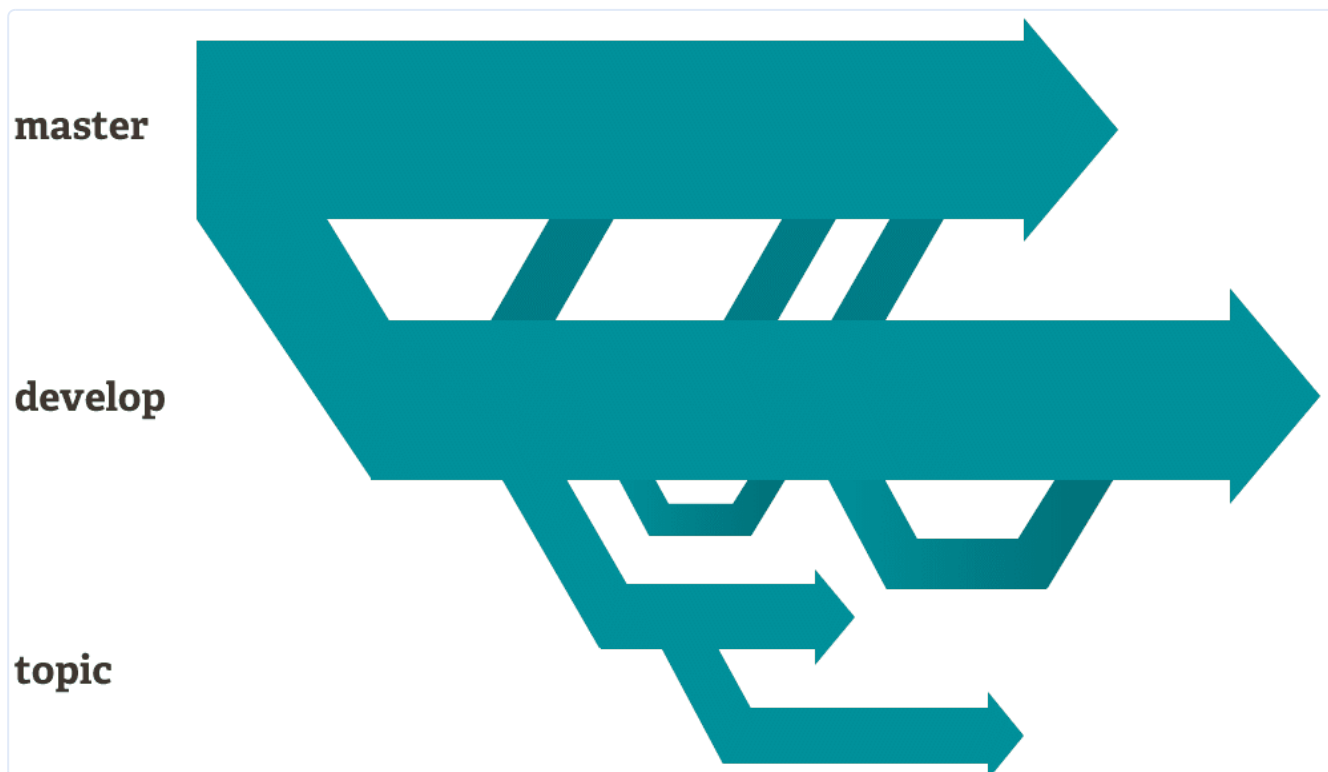
## Noções básicas de Git e Repositórios de Git baseados em nuvens

Se você é um iniciante completo e novo no Git, você deve aprender alguns fundamentos antes de escolher a opção certa. Mas se você for um usuário experiente de Git, sinta-se à vontade para pular esta seção.

Git é um [sistema de controle de versão de código aberto \(VCS\)](#) que permite salvar “snapshots” de um projeto de software. Em essência, ele mantém registro de todas as mudanças que você faz em qualquer programa.

O que o torna diferente do controle da versão tradicional é que você pode trabalhar em diferentes versões de filiais do software e gerenciá-las todas ao mesmo tempo.

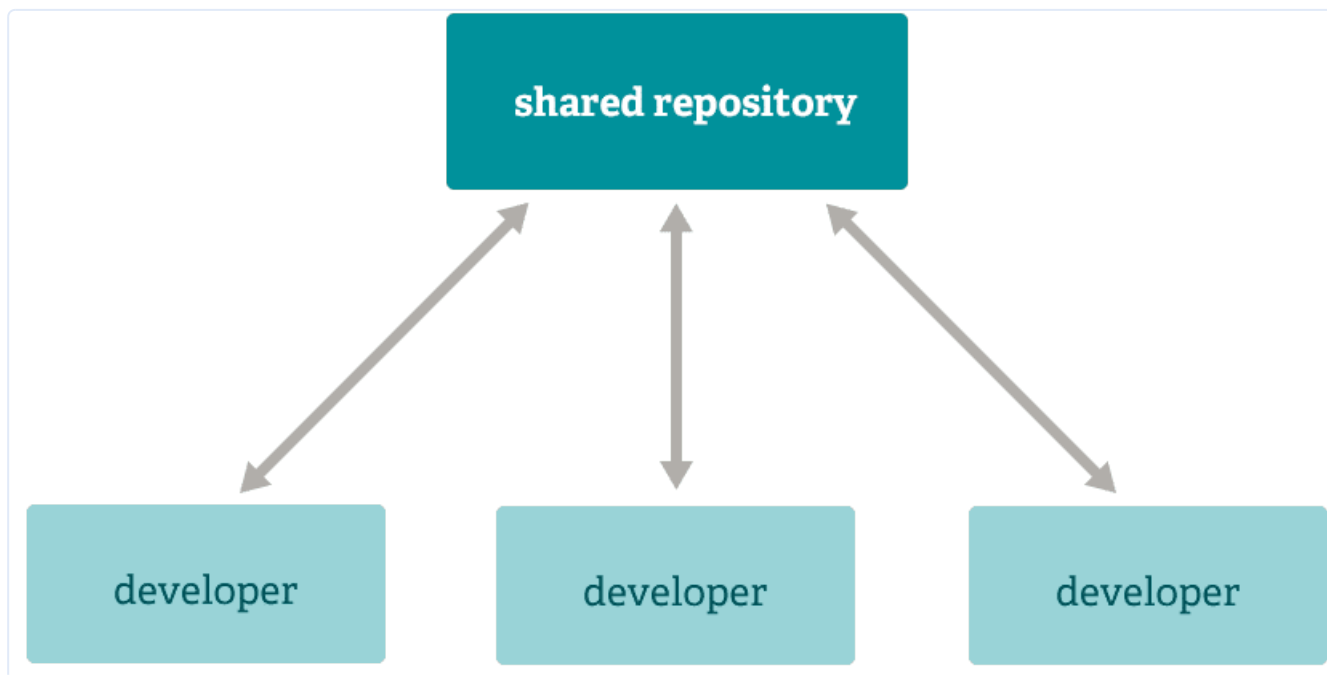
Você pode então testá-los fundindo-os com um branch de teste, e depois com a versão ativa quando você tiver finalizado uma variante melhor.



— branch Git ([Fonte](#))

Somente com Git você tem esta liberdade para trabalhar em uma grande variedade de versões de uma só vez. É uma obrigação para qualquer equipe que queira fazer um desenvolvimento baseado em características.

Com Git, tudo isso acontece localmente em seu computador. Mas você também pode solicitar e dar acesso a outros desenvolvedores manualmente – sobre uma LAN, por exemplo.



— Repositório compartilhado Git ([Fonte](#))

Ao trabalhar com equipes remotas ou distribuídas maiores, a maioria das empresas recorre a soluções baseadas nas nuvens. Entre no GitLab e no GitHub.

Eles oferecem [armazenamento ilimitado baseado em nuvem](#) para repositórios. Desenvolvedores de todo o mundo podem facilmente colaborar em código. E eles podem fazê-lo sem ter que configurar uma LAN virtual ou compartilhar manualmente os repositórios.

Mas os repositórios de Git baseados em nuvens como GitHub e GitLab vão além da funcionalidade básica de Git. Eles também incluem gerenciamento de projetos, colaboração e ferramentas de implantação para ajudar a melhorar o fluxo de trabalho de desenvolvimento.

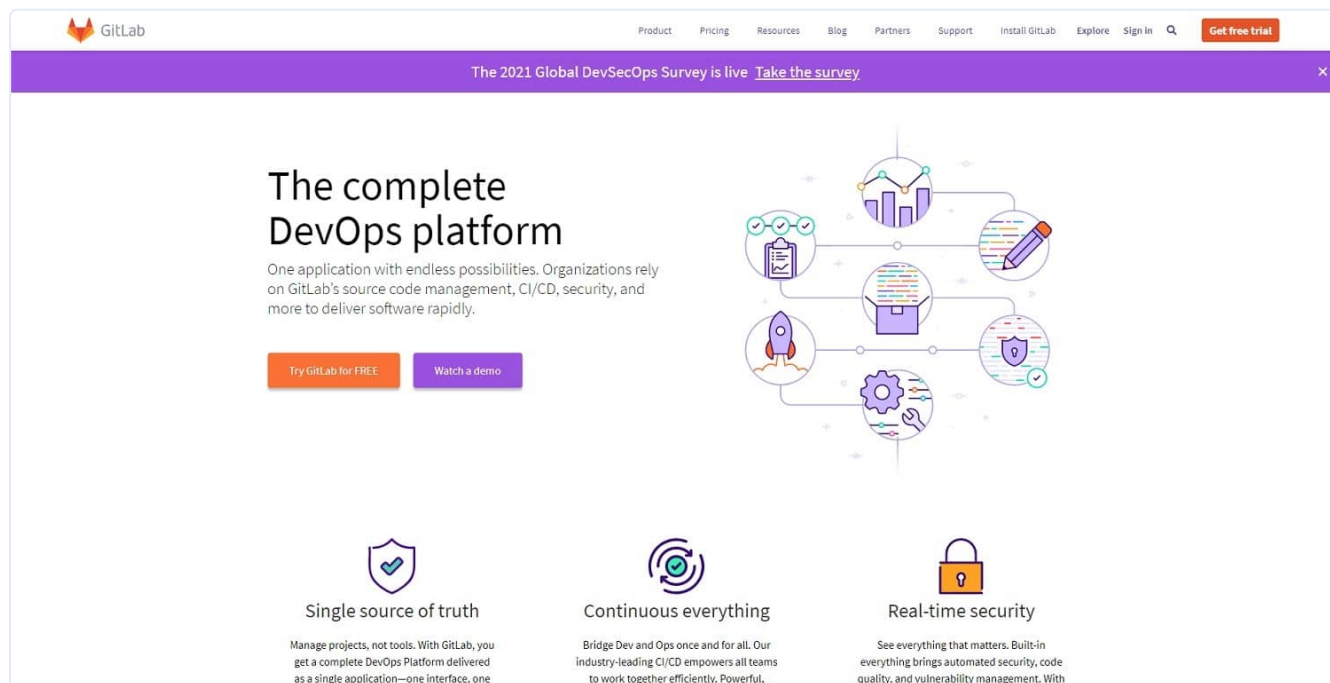
Nosso artigo [Git vs GitHub](#) explica estes aspectos com mais detalhes.

Agora, vamos ver como GitLab e GitHub se empilham um contra o outro.

## **GitLab vs GitHub: O básico**

Vamos começar com o básico, explorando rapidamente cada plataforma e o que cada empresa traz para a mesa.

## O que é o GitLab?



— Página inicial do GitLab

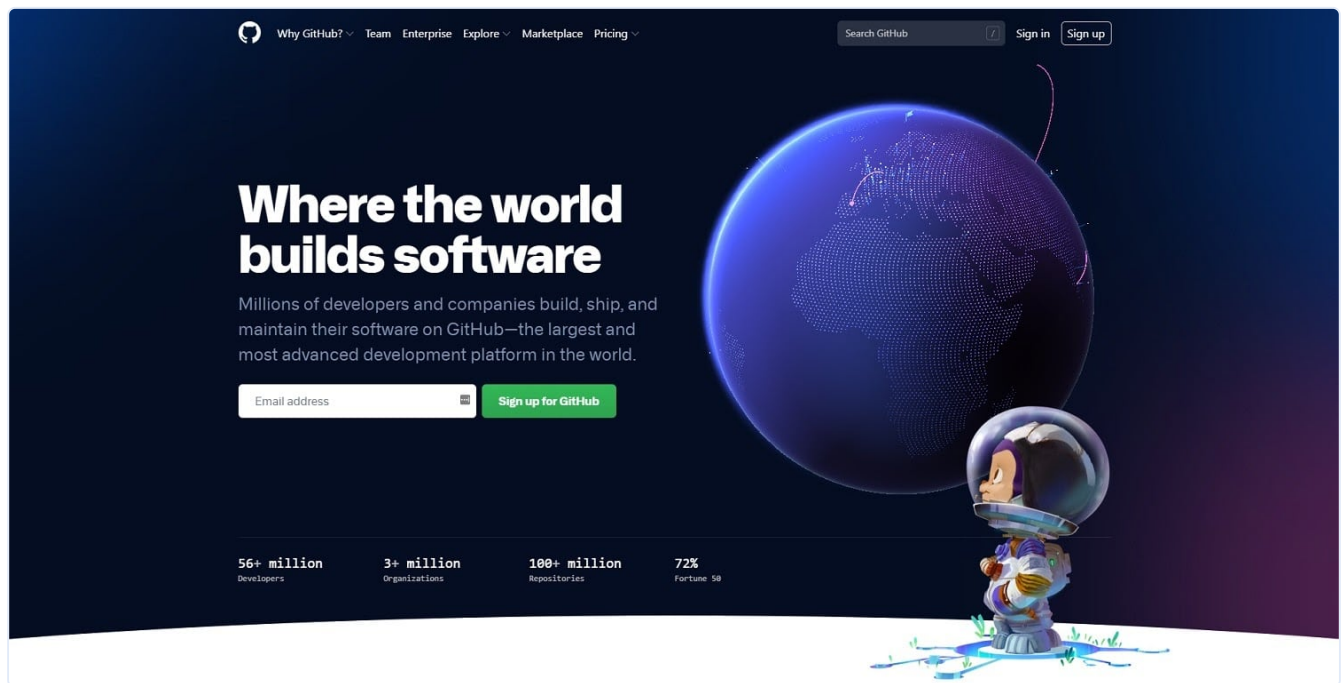
[GitLab](#) é uma plataforma Git e DevOps baseada em nuvem que ajuda os desenvolvedores a monitorar, testar e implantar seu código.

Um repositório de Git de nuvem costumava ser o principal ponto de venda do GitLab. Mas a plataforma evoluiu além de suas origens simples. Hoje, GitLab oferece uma ampla gama de recursos [DevOps](#), como integração contínua, segurança, e até mesmo ferramentas de implantação de aplicativos.

GitLab começou como uma alternativa de código aberto auto-hospedado ao GitHub. Agora ele oferece planos SaaS gratuitos e pagos também.

Também oferece [ferramentas essenciais de gerenciamento de projetos](#) para monitorar e controlar os membros da sua equipe. Não é apenas um sistema de controle de versão para o código fonte do seu software.

## O que é GitHub?



— Página inicial do GitHub

[GitHub](#) é uma das plataformas originais de Git baseadas em nuvem que permite aos desenvolvedores hospedar e monitorar suas mudanças de código. Ele também evoluiu para se tornar uma plataforma de desenvolvimento completa.

Mas é menos um pacote “pronto” do que o GitLab, optando por dar aos desenvolvedores a opção de implementar aplicativos e integrações livremente através do mercado GitHub.

O foco está em utilizar a vibrante comunidade GitHub e deixá-los contribuir com a funcionalidade que lhes falta.

## **GitLab e GitHub são a mesma empresa?**

Não, os dois serviços online são oferecidos por empresas completamente diferentes. A GitLab Inc. é uma empresa privada, totalmente remota, com 1.303 funcionários.

GitHub é uma empresa filial da Microsoft, com sede em São Francisco e emprega mais de 1.600 pessoas em todo o mundo.

Embora esta seja uma pergunta freqüente – e uma concepção errônea – as duas empresas não estão relacionadas, exceto que elas são concorrentes. É um pouco como pensar que Coca-Cola e Pepsi são a mesma empresa.

## **A Microsoft também possui o GitLab?**

Não, a Microsoft não é proprietária do GitLab somente GitHub.

GitLab é uma empresa privada com participações minoritárias detidas por vários investidores de capital de risco e fundos de investimento de capital de risco. Em particular, Goldman Sachs e Iconiq Capital são investidores significativos no GitLab.

GitLab costumava hospedar seus serviços no [Microsoft Azure](#), mas mudou-se para a [plataforma Google Cloud](#) depois que a [Microsoft adquiriu o GitHub](#).

## **GitLab vs GitHub: Semelhanças**

Ok, então agora sabemos que apesar de oferecer – à primeira vista – serviços muito semelhantes, as empresas são 100% diferentes.



A seguir, vejamos algumas das semelhanças mais específicas além do nível da superfície do nome e da página inicial. Vamos começar com as características.

## GitLab vs GitHub: Características

Quando se trata das características gerais de alto nível, há poucas diferenças entre os dois serviços. A maior parte do que o GitLab oferece, GitHub também tem no menu, e vice versa.

<b>Destaque</b>	<b>GitLab</b>	<b>GitHub</b>
Git	?	?
Versão auto-hospedada	?	? (Com plano da empresa)
Integração e entrega contínua	?	? (Com aplicativos de terceiros)
Documentação baseada em Wiki	?	?
Pré-visualização de mudanças de código	?	?
Rastreador de problemas	?	?
<a href="#">Revisão do código</a>	?	?
Cessionários de Múltiplos Assuntos	? (Plano pago)	? (Somente Repositório Público em Plano gratuito)
Quadros de gerenciamento de projetos	?	?
Discussões da equipe	?	?
Rastreamento do tempo	?	? (Com App)
Ferramentas de Segurança e Conformidade	?	?
Teste de desempenho de carga	? (Plano pago)	? (Com App)
Teste de desempenho do navegador	? (Plano pago)	? (Com App)
Iterações e Planejamento de Sprint (Incluindo a Carta de Burndown)	? (Plano pago)	? (Com App)
Dependências de Emissão	? (Plano pago)	?

Não há áreas significativas onde um provedor oferece apoio e o outro não. A diferença está em como você pode acessar estes recursos.

Em muitos casos, você precisa usar um aplicativo ou integração de terceiros com o GitHub. Você pode obter o mesmo resultado – o recurso – mas você precisa pular por alguns arcos extras para chegar lá. Cobriremos esta diferença com mais detalhes mais tarde, pois localizamos onde os serviços diferem uns dos outros.

Vamos além da lista de características, e ver mais de perto como as duas correspondem especificamente a Git.

## Controle de versão e funcionalidade Git

A funcionalidade do Git e os comandos básicos permanecem na maioria das vezes os mesmos entre as duas plataformas:

- **Branch** : Criar uma linha independente de desenvolvimento com o comando branch. Ela pode representar uma característica menor, uma versão de desenvolvimento ou o branch mestre.
- **Fork**: Um fork é uma cópia pessoal de um repositório de código específico que qualquer desenvolvedor pode usar e experimentar, mesmo que seja um colaborador externo. Você pode então enviar pedidos para fundir (merge) as mudanças feitas neste fork com outro Branch mais tarde.
- **Pull**: Uma solicitação pull (ou solicitação de fusão no GitLab) é uma apresentação de mudanças de código sugeridas para a branch editada. Uma vez aceita, ela integra automaticamente as mudanças de código na branch em questão. Um [desenvolvedor sênior](#) ou uma equipe de QA testará quaisquer mudanças antes de incorporá-las à branch mestre.
- **Merge (fundir)**: Nome do GitLab para pedidos pull.

Além da renomeação dos pedidos pull para fundir (merge) que pode fazer mais sentido intuitivo – há pouca diferença na real funcionalidade básica Git entre eles.

Mas quando se trata do fluxo de trabalho recomendado usando estes recursos, elas são bem diferentes. GitLab e GitHub têm visões díspares sobre como é um fluxo de trabalho de codificação ideal, e vamos cobrir essas diferenças em nossa próxima seção.

O sistema de controle de versão e Git estão, naturalmente, no centro do que essas ferramentas têm a oferecer, mas hoje em dia elas se tornaram apenas os alicerces de um serviço completo.

Ambas as plataformas oferecem uma ampla gama de colaboração, revisão de código e ferramentas de gerenciamento de projetos.




## **Colaboração, Código e Gerenciamento de Projetos**

Ambas as plataformas incluem ferramentas embutidas de colaboração e [revisão de código](#). Isso faz parte do kit essencial incluído em ambas as versões gratuitas.

Por exemplo, você pode visualizar e discutir pedidos pull (ou fundir) em tempo real, fazer alterações e obter uma visão completa das diferenças de código.

# Discuss code

10	-	# Layouts
11	+	# Grid system
12	+	Layouts are built on a 12 column grid.

**rubyjazzy** 1 day ago  
Good call, this is more specific!  
 3  1


## Comments


On GitHub, conversations happen alongside your code. Leave detailed comments on code syntax and ask questions about structure inline.

## Review requests

If you're on the other side of the code, requesting peer reviews is easy. Add users to your pull request, and they'll receive a notification letting them know you need their feedback.

Request a review ×

 **sophshep** Sophie Shepherd

 **mdo** Mark Otto

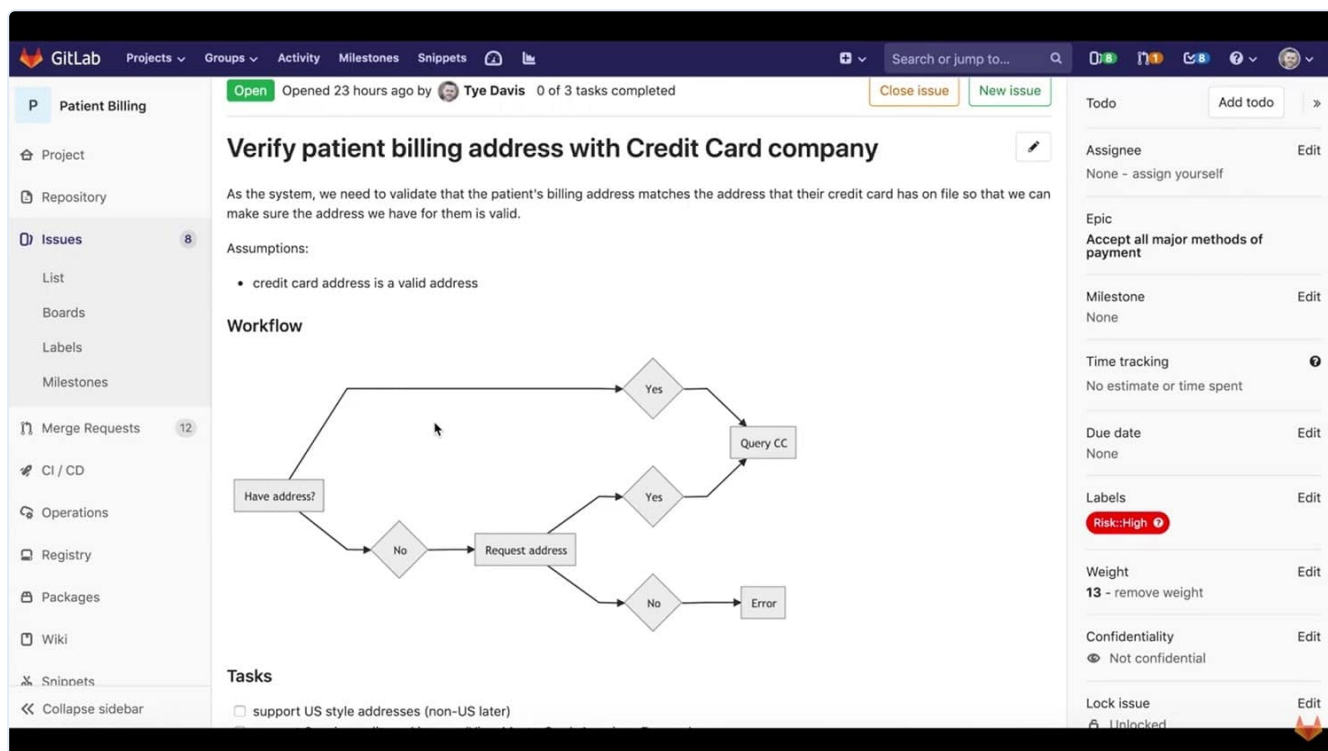
— Ferramentas de revisão de código GitHub

Estas ferramentas destacam todas as mudanças e facilitam a revisão do código para quem quer que esteja revendo. Pode ajudar a detectar erros precocemente, reduzindo a chance de erros fatais ou [falhas de segurança](#).

Enquanto a captura de tela acima destaca o GitHub, o GitLab oferece ferramentas igualmente poderosas de colaboração e revisão de códigos.

Ambas as plataformas também incluem ferramentas essenciais de gerenciamento de projetos.

Por exemplo, no GitLab, você pode criar uma hierarquia lógica entre as questões, adicionar listas de tarefas e descrições, e atribuir diferentes desenvolvedores a diferentes branch.



— Ferramentas de gerenciamento de projetos do GitLab

Também dá a seus gerentes acesso a gráficos de progresso como gráficos queimados e outras ferramentas de visualização.

No GitHub, você também pode categorizar questões, pedidos e notas em projetos específicos. Além disso, você pode definir e rastrear marcos e medir a produtividade com vários gráficos e relatórios.

Há também mais de 500 aplicativos de gerenciamento de projetos e integrações disponíveis no mercado GitHub.

## **Páginas GitHub vs Páginas GitLab**

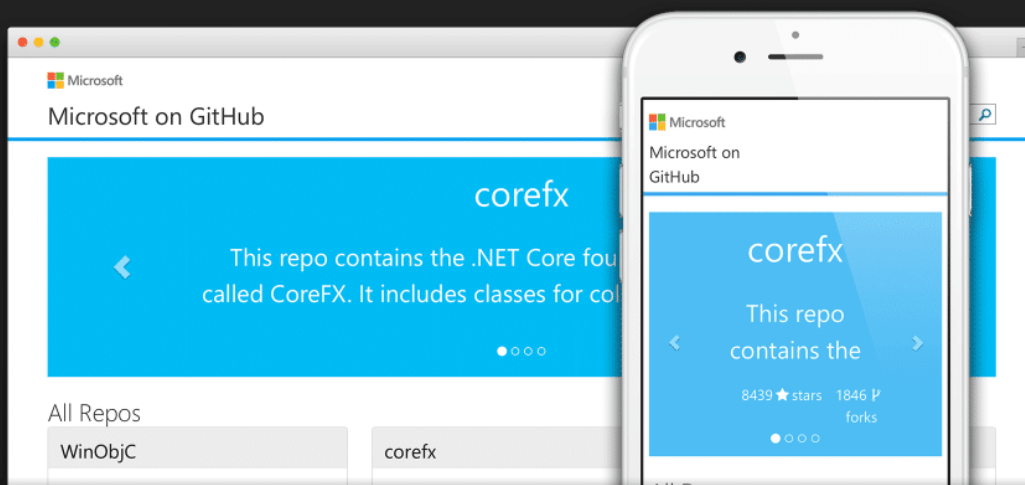
Tanto GitHub como GitLab oferecem [páginas web estáticas](#) gratuitas com informações sobre seu projeto e repositório de software.

Os desenvolvedores em qualquer uma das plataformas podem então facilmente bifurcar o projeto para começar a fazer suas mudanças e experimentar com eles.

GitHub Pages

# Websites for you and your projects.

Hosted directly from your [GitHub repository](#). Just edit, push, and your changes are live.



— Páginas de GitHub

GitHub tem um tutorial básico de instalação, incluindo alguns temas básicos para ajudá-lo a criar seu site.

Nem o GitHub nem o GitLab oferecem processamento dinâmico do lado do servidor, como você precisaria com arquivos PHP ou ASP. Você só pode publicar conteúdo estático do site como HTML e CSS.

Embora isto signifique que você não pode mover seu site WordPress diretamente para GitHub ou GitLab, você ainda *pode* ter [controle de versão no WordPress](#) se você tiver a

configuração adequada. Com Kinsta, você pode mover diretamente do seu repositório Git usando SSH.

## **Plano Gratuito com Repositórios Ilimitados**

Ambos os serviços oferecem planos gratuitos com repositórios ilimitados (públicos e privados). Eles também limitam o acesso a recursos avançados de gerenciamento, segurança e conformidade em suas opções gratuitas.

Estas últimas características estão no topo de qualquer lista de prioridades empresariais, mas pequenas equipes e freelancers não vão sentir muita falta delas.

Com tudo considerado, as duas opções são bastante semelhantes. É claro, isso provavelmente não ajuda muito na escolha de uma em detrimento da outra.

Então vamos cavar um pouco mais fundo e explorar algumas das diferenças entre estas plataformas de Git.

## **GitLab vs GitHub: Principais diferenças**

Como você provavelmente adivinhou pelos nomes deles, GitLab e GitHub têm mais em comum do que não.


Mas isso não significa que eles estão sem suas diferenças. Nesta seção, vamos cobrir as principais discrepâncias entre as duas plataformas.

### **Integração contínua**

Talvez a diferença mais significativa na experiência do usuário seja a dedicação do GitLab à integração contínua e ao fluxo de trabalho DevOps.

As ferramentas CI GitLab permitem construir, implementar e executar códigos automaticamente sem a necessidade de atualizações manuais ou integrações pesadas e personalizadas.




Product Pricing Resources Blog Partners Support Install GitLab Explore Sign in
Get free trial

- **Detect errors as quickly as possible:** Fix problems while fresh in developers minds
- **Reduce integration problems:** Smaller problems are easier to digest and solve
- **Avoid compounding problems:** Allows teams to develop faster, with more confidence

### Seamless path to CD

- **Ensure every change is releasable:** Test everything, including deployment, before calling it done
- **Lower the risk of each release:** Make releases "boring"
- **Deliver value more frequently:** Reliable deployments mean more releases
- **Tight customer feedback loops:** Fast and frequent customer feedback on changes

### Advantages of using GitLab for CI

- **Integrated:** CI is built into GitLab, enabling a single conversation from ideation, to planning, to deployment (and beyond)
- **Open source:** CI, along with CD, is a part of both the open source GitLab Community Edition and the proprietary GitLab

— GitLab para CI

Se você já utiliza uma [solução PaaS](#) como a Kubernetes hospedada, a integração GitLab permitirá que você implante, encene, teste e escale seus aplicativos automaticamente.

Trabalhando somente dentro do GitLab, sua equipe de desenvolvimento pode se concentrar em fazer pequenas atualizações ao vivo várias vezes por dia, em vez de transformá-las em grandes lançamentos.

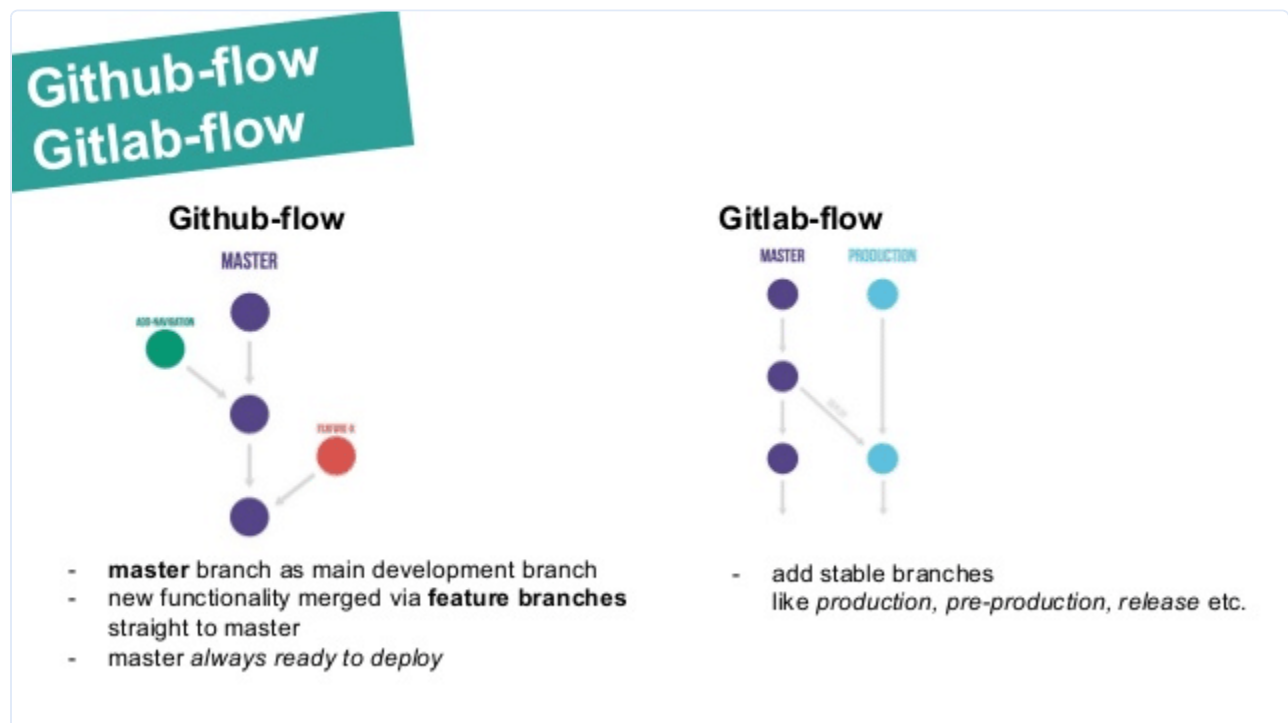
Uma das coisas cruciais que faz isso acontecer é a ferramenta de teste automatizado que automaticamente varre o código em busca de potenciais riscos de segurança.

Enquanto você pode estabelecer um fluxo de trabalho CI/CD com GitHub, você precisa trabalhar com uma ferramenta CI de terceiros como TravisCI ou CircleCI. Você não tem a mesma plataforma de recursos de CI que interagem diretamente com seu repositório GitHub.

## GitHub Flow vs GitLab Flow

Embora a tecnologia e as características subjacentes da Git sejam praticamente idênticas, o fluxo de trabalho recomendado não é o mesmo. Nem de perto.

GitHub enfatiza a velocidade, enquanto GitLab enfatiza a confiabilidade. Essa é a essência da diferença entre as duas plataformas de repositório de nuvens.



— Fluxo GitHub vs fluxo GitLab ([Fonte](#))

GitHub defende uma abordagem de [desenvolvimento rápida e focada em recurso](#) para **fundir** (incluir) novas branches com a branches mestre. Este fluxo de trabalho é perfeito para equipes e projetos Agile menores.

A branch mestre está sempre pronto para ser implantado, garantindo que você possa rapidamente restabelecer o status quo se algo der errado. Você pode voltar para a versão anterior em segundos.

No fluxo de trabalho do GitLab, você cria múltiplos branches estáveis além do master, geralmente pelo menos a produção e pré-produção. Isso significa um processo de teste em múltiplas etapas onde uma única revisão de código após a solicitação de fusão não é suficiente.

Para equipes menores, ele pode retardar o processo até que um novo recurso esteja viva na produção. Mas se você tiver uma equipe de QA dedicada, é uma maneira de deixar sua equipe RandD trabalhar livremente em novas funcionalidades sem se preocupar em testar cada minuto de mudança de código.

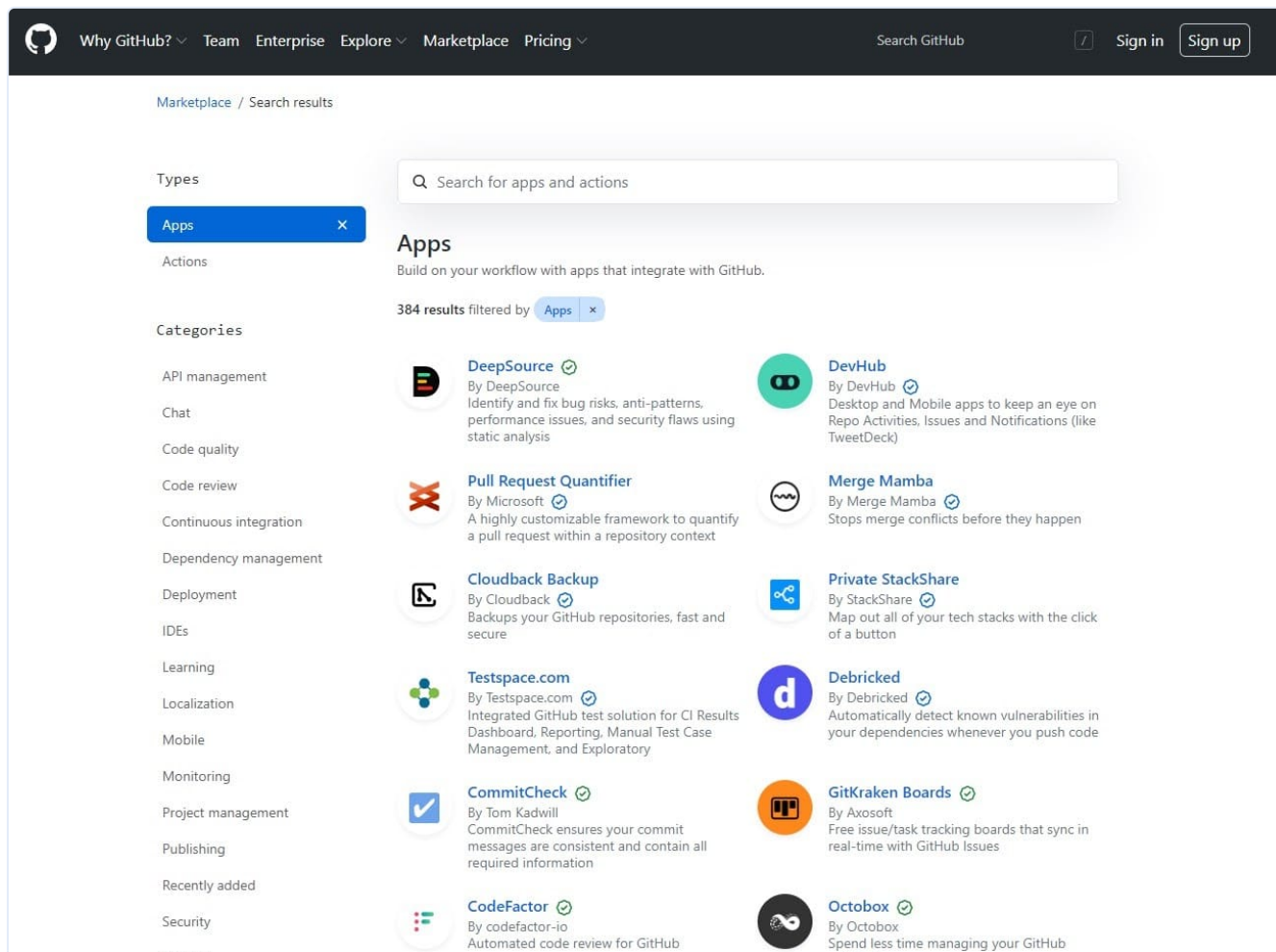
Alguns concluem que isto faz do GitLab a plataforma GitLab mais segura, mas o GitHub também suporta o mesmo fluxo de trabalho. É que ele não o recomenda oficialmente.

## **Plataforma completa vs Mercado**

GitLab adotou a abordagem de uma plataforma completa e embalada, em vez de dar a todos a opção de construir sua plataforma com diferentes aplicativos.

Isso não significa que é completamente vazio de integrações. GitLab suporta integrações com Asana, Jira, [Microsoft Teams](#), [Slack](#), Gmail, e mais de 30 outros aplicativos e plataformas.

Mas enquanto você tem opções, elas são muito mais limitadas do que se você for com GitHub. GitHub atualmente tem 374 aplicativos gratuitos em seu mercado, com um alto número de opções pagas para iniciar.



— GitHub marketplace

Mas você tem que lembrar que é através destas integrações de terceiros que GitHub oferece *algumas das* características out-of-the-box do GitLab.

Por exemplo, GitLab oferece recursos como integração contínua, rastreamento de tempo e backups por padrão. Com GitHub, porém, você precisaria de um aplicativo para essa funcionalidade, como o aplicativo gratuito CI Jenkins.

*GitHub* Ações automatizadas tornam possível o CI/CD sem ferramentas de terceiros, mas você não terá o mesmo nível de supervisão.

**Nota:** Mesmo que o aplicativo GitHub em si seja gratuito, o serviço de terceiros ainda custa dinheiro.

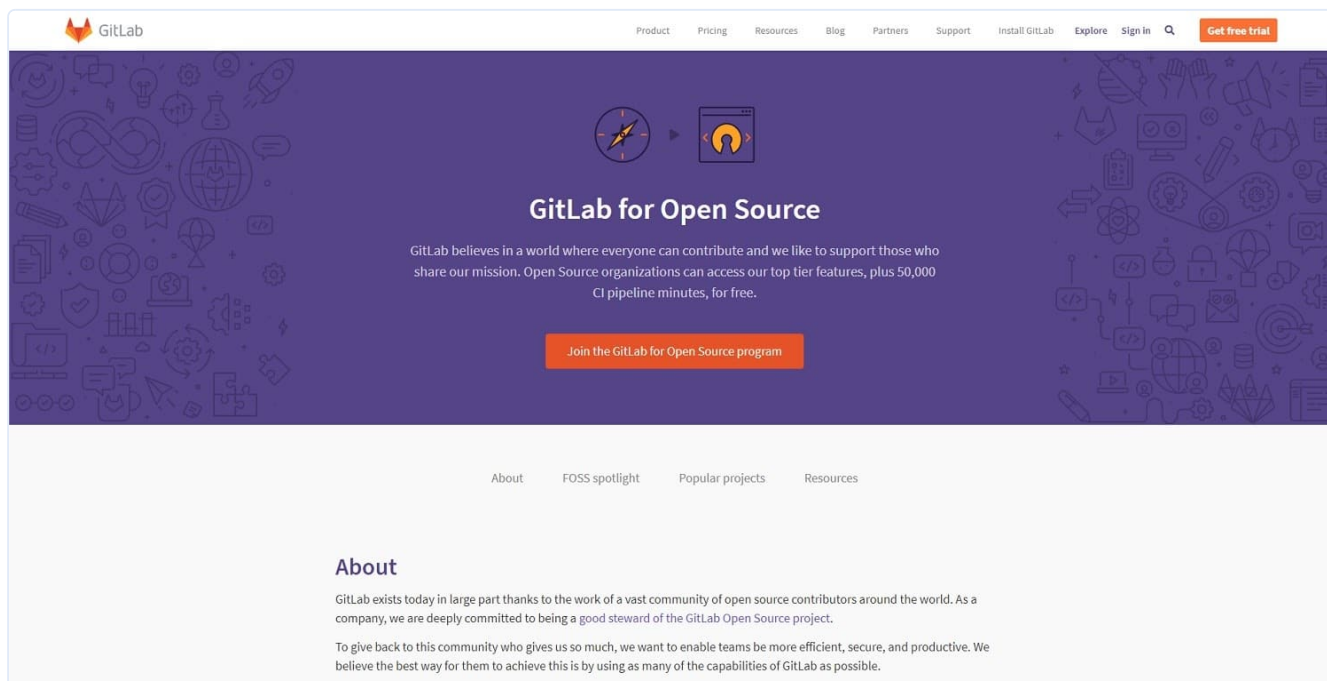
## **Instalação auto-hospedada para servidor privado**

O GitLab começou como uma plataforma de código aberto auto-hospedada. Dito isto, você ainda tem a opção de hospedar uma versão privada do GitLab em suas máquinas virtuais.

E você pode fazer isso com um plano GitLab gratuito, enquanto com GitHub, ele só está disponível para os planos empresariais GitHub.

## **Código Aberto**

GitLab oferece recursos gratuitos de plano empresarial e até 50.000 minutos CI para organizações ou equipes menores que trabalham com software de código aberto.



— GitLab para Código Aberto

O próprio GitLab é software de código aberto, e a versão auto-hospedada é gratuita para qualquer pessoa usar.

Isso não quer dizer que o GitHub não seja uma boa opção para código aberto, pois oferece acesso instantâneo à maioria desses desenvolvedores. GitHub também os incentiva a contribuir com o código aberto toda sexta-feira, expandindo regularmente o número de recursos públicos disponíveis.

As equipes de desenvolvedores da GitHub também contribuem para projetos open source especificamente Git LFS e outros projetos relacionados a Git.

## Planos e preços pagos: GitHub Enterprise vs GitLab Ultimate

Embora ambos os planos gratuitos ofereçam repositórios públicos e privados ilimitados, há uma notável disparidade nas etiquetas de preço dos planos pagos.

O plano Premium do GitLab começa em **\$19** por usuário/mês. Ele dá aos gerentes acesso ao gerenciamento do projeto e aos controles de integridade do código.

The screenshot shows the GitLab pricing page with the heading "Choose a plan that suits your needs" and a subtext "All plans have unlimited private or public repositories". There are three plan cards:

- FREE**: A complete DevOps platform, \$0. Includes: All stages of the DevOps lifecycle, Bring your own CI runners, Bring your own production environment, Static application security testing, 400 CI/CD minutes. Button: "Start now".
- Premium**: DevOps with project management, code integrity controls, and productivity analytics, \$19/user/month (billed annually at \$228 USD). Includes: Cross-team project management and roadmaps, Code integrity controls and approvals, Code and productivity analytics, Disaster recovery, 24/7 support. Button: "Buy now".
- Ultimate**: Full DevSecOps with portfolio management, advanced security testing, and compliance, \$99/user/month (billed annually at \$1188 USD). Includes: Vulnerability management, Company wide portfolio management, Advanced application security, Dependency scanning, Compliance automation, Free guest users, 50,000 CI/CD minutes. Button: "Buy now".

Below the Premium card, it says "All the benefits of Free +". Below the Ultimate card, it says "All the benefits of Premium +".

— Preços do GitLab

Da mesma forma, os planos pagos para GitHub também se concentram em ferramentas avançadas de colaboração (para repositórios privados), bem como recursos de segurança e conformidade.

Mas há uma grande diferença entre as duas plataformas no preço.

O plano da Equipe GitHub começa em **\$4** por usuário/mês.

The screenshot shows the GitHub Pricing page. At the top, there's a navigation bar with links like 'Why GitHub?', 'Team', 'Enterprise', 'Explore', 'Marketplace', and 'Pricing'. Below this, the heading 'Plans for all developers' is centered. A prominent blue banner states 'NEW GitHub is now free for teams' with a 'Try GitHub Free' button. Below the banner, four pricing plans are listed in columns: 'Free', 'Team', 'Enterprise', and 'GitHub One'. Each plan includes a list of features and a price. The 'Free' plan is \$0/month, 'Team' is \$4 per user/month, 'Enterprise' is \$21 per user/month, and 'GitHub One' has a 'Learn more' button. A 'Contact Sales' button is also present for the Enterprise and GitHub One plans.

Free	Team	Enterprise	GitHub One
Basics for teams and developers	Advanced collaboration and support for teams	Security, compliance, and flexible deployment for enterprises	All of our best tools, support, and services
<ul style="list-style-type: none"><li>Unlimited public/private repositories</li><li>Unlimited collaborators</li><li>2,000 Actions minutes/month (Free for public repositories)</li><li>500MB of GitHub Packages storage (Free for public repositories)</li><li>Community Support</li></ul>	<ul style="list-style-type: none"><li>Unlimited public/private repositories</li><li>Required reviewers</li><li>3,000 Actions minutes/month (Free for public repositories)</li><li>2GB of GitHub Packages storage (Free for public repositories)</li><li>Code owners</li></ul>	<ul style="list-style-type: none"><li>Everything included in Team</li><li>SAML single sign-on</li><li>50,000 Actions minutes/month (Free for public repositories)</li><li>50GB of GitHub Packages storage (Free for public repositories)</li><li>Advanced auditing</li></ul>	<ul style="list-style-type: none"><li>Everything included in Enterprise</li><li>Community-powered security</li><li>Actionable metrics</li><li>24/7 support</li><li>Continuous learning</li></ul>
\$0/month	\$4 per user/month	\$21 per user/month	Learn more
Join for free	Continue with Team	Contact Sales	Contact Sales

— GitHub preços

GitLab Ultimate lhe dará um retorno de **\$99** por usuário/mês contra **\$21** por usuário/mês para GitHub Enterprise.

GitLab posiciona seu plano gratuito como um concorrente do plano da Equipe GitHub e seu plano Premium como sua contraparte do plano Enterprise.

Mas em termos de segurança e ferramentas de conformidade, você vai precisar do plano mais caro do GitLab para espelhar as características do GitHub.



# GitLab vs GitHub: Qual é a melhor opção?

Infelizmente, como a maioria das coisas na vida, não há aqui uma “melhor escolha para todos” clara. Depende da sua equipe, maior organização, objetivos e casos de uso únicos.

Em termos de funcionalidade out-of-the-box e auto-hospedagem, o GitLab sai por cima. Mas GitHub está em uma posição única como líder de mercado. Tem um número muito maior de integrações e aplicativos.

- **Para os repositórios privados auto-hospedados**, o GitLab é a melhor escolha. Você pode começar sem ter que pagar nenhuma taxa de assinatura mensal.
- Ambas as opções são ótimas para **projetos de código aberto**, com o GitLab oferecendo recursos premium gratuitamente e o GitHub oferecendo acesso a um grupo maior de desenvolvedores de código aberto ativos.
- Para **projetos comerciais**, GitHub oferece um plano empresarial significativamente mais barato.
- Se você está procurando entrar no **DevOps**, o GitLab é a escolha mais ponderada e provavelmente mais barata. (Você pode começar de graça sem depender de nenhum outro serviço pago).

Depende também das ferramentas que suas equipes já utilizam e dominam. Se o GitLab não oferece uma integração com uma ferramenta que você usa regularmente, mas o GitHub oferece, isso fará dele a escolha óbvia para você.

E como GitHub oferece mais de 300 integrações do que GitLab, este não é um cenário rebuscado.

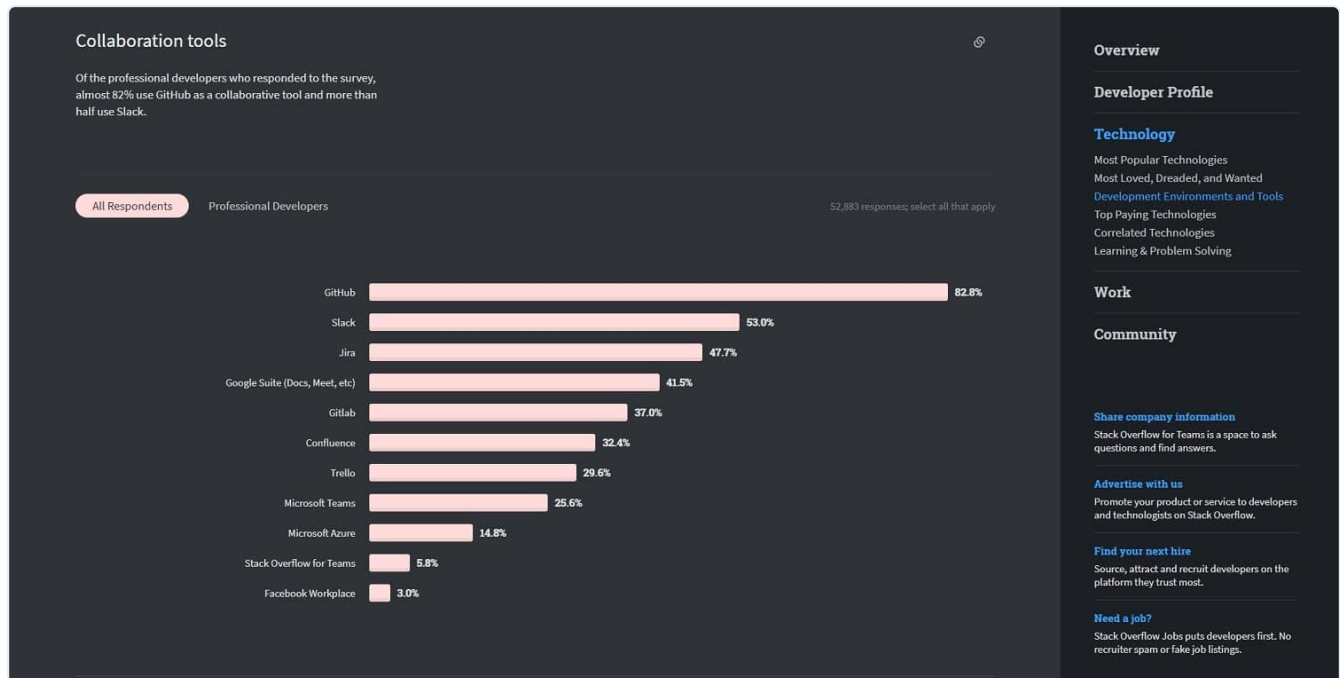
## Por que o GitHub é mais popular que o GitLab?

Uma das principais razões pelas quais GitHub é mais popular do que GitLab é que GitHub teve um avanço. Ele foi lançado em 2008, enquanto GitLab começou como um projeto de código aberto em 2011.

Mas com recursos integrados de integração contínua e uma opção gratuita para servidores privados e auto-hospedados, o GitLab está começando a se recuperar. De acordo com uma pesquisa, GitLab [ganhou uma participação de mercado de 4,6%](#) de 2018-19, enquanto

GitHub caiu 0,4%.

Mas mesmo assim, na última pesquisa de desenvolvimento Stack Overflow 2020, 82,8% dos entrevistados dizem usar GitHub, e apenas 37% dizem usar GitLab.



— Uso de ferramentas de colaboração em código ([Fonte](#))

Embora GitHub ainda seja a escolha padrão para novos desenvolvedores, isso está mudando lentamente à medida que GitLab se desenvolve em uma plataforma especializada com características e benefícios únicos.

## Resumo

Como desenvolvedor em 2021, você não pode se dar ao luxo de ignorar ferramentas como Git ou grandes provedores de repositório de nuvens como GitHub e GitLab.

A grande maioria dos desenvolvedores, equipes e empresas utilizam estas ferramentas para garantir que cada código seja revisado cuidadosamente antes de entrar em funcionamento, e que a contribuição de ninguém desapareça.

Embora as diferenças entre as ferramentas pareçam mínimas na superfície, a principal diferença é esta: O GitLab oferece uma plataforma DevOps abrangente. Em contraste, GitHub oferece uma profunda extensibilidade através de seu mercado de aplicativos.