

Introdução ao Git

1. O que é Git?

O Git é um projeto de código aberto maduro e com manutenção ativa desenvolvido em 2005 por Linus Torvalds, o famoso criador do kernel do sistema operacional Linux. Git é um sistema de controle de versão de arquivos. Através deles podemos desenvolver projetos em que diversas pessoas podem contribuir simultaneamente no mesmo, editando e criando novos arquivos e permitindo que os mesmos possam existir sem o risco de suas alterações serem sobrescritas. O Git permite que um arquivo possa ser editado ao mesmo tempo por pessoas diferentes.

Outro fator importante do git é a possibilidade de criar, a qualquer momento, vários snapshots do seu projeto, ou branch.

2. Existem outros sistemas de controle de versão?

Sim, existem sistemas de controle de versão centralizados, como o Subversion, e distribuídos, como o Mercurial

3. Quais as vantagens de se utilizar um sistema de controle de versão como o Git?

As características brutas de desempenho do Git são muito fortes quando comparadas a muitas alternativas. Fazer o commit de novas alterações, branches, mesclagem e comparação de versões anteriores – tudo é otimizado para desempenho. Os algoritmos implementados no Git aproveitam o conhecimento profundo sobre atributos comuns de árvores de arquivos de código-fonte reais, como costumam ser modificados ao longo do tempo e quais são os padrões de acesso.

O Git foi projetado com a integridade do código-fonte gerenciado como uma prioridade. O conteúdo dos arquivos, bem como os verdadeiros relacionamentos entre arquivos e diretórios, versões, tags e commits, todos esses objetos no repositório do Git são protegidos com um algoritmo de hash de criptografia seguro chamado SHA1. Isso protege o código e o histórico de alterações contra alterações acidentais e maliciosas e garante que o histórico tenha rastreabilidade total.

Um dos principais objetivos de design do Git é a flexibilidade. O Git é flexível em vários aspectos: suporte a vários tipos de fluxos de trabalho de desenvolvimento não lineares, em eficiência em projetos pequenos e grandes e em compatibilidade com muitos sistemas e protocolos existentes.

4. Qual o comando para indicar que uma pasta deve ser gerenciado pelo Git?

`"git init"`

5. Ao adicionar/editar arquivos dentro de uma pasta gerenciada pelo Git, quais os comandos necessários para adicionar estas mudanças, de fato, ao Git?

- `git add <arquivos...>` - Este comando adiciona o(s) arquivo(s) num lugar que chamamos de INDEX, que funciona como uma área do git no qual os arquivos possam ser enviados ao Github. É importante saber que ADD não está adicionando um arquivo novo ao repositório, mas sim dizendo que o arquivo (sendo novo ou não) está sendo preparado para entrar na próxima revisão do repositório.
- `git commit -m "comentário qualquer"` - Este comando realiza o que chamamos de "commit", que significa pegar todos os arquivos que estão naquele lugar INDEX que o comando add adicionou e criar uma revisão com um número e um comentário, que será vista por todos.
- `git push` Push (empurrar) - é usado para publicar todos os seus commits para o github. Neste momento, será pedido a sua senha.

- `git status` - Exibe o status do seu repositório atual

6. Qual a função do comando "checkout"?

O comando "checkout" serve para mudarmos para uma outra *branch*.

7. Pesquise e explique o que é uma "branch"?

Uma branch é uma ramificação no *git*, um ponteiro para as alterações feitas nos arquivos do projeto. É útil em situações nas quais queremos adicionar um novo recurso ou corrigir um erro, gerando uma nova ramificação garantindo que o código instável não seja mesclado nos arquivos do projeto principal. Depois de concluir a atualização dos códigos da ramificação, você pode mesclar a ramificação com a principal, geralmente chamada de *master*.

É semelhante aos snapshots que fazemos em VM's no Virtual Box, por exemplo, que permite guardar trabalho já feito antes de avançar para um passo importante, para que o trabalho feito anteriormente não se perca.

8. Diversas plataformas existem no mercado para suportar o uso de Git via Web. Procure algumas destas plataformas. Procure informações como preço, número de pessoas, se oferece o serviço gratuitamente, se possui repositórios privados. Obs: No mínimo 3 plataformas.

- O GitLab é a escolha número um para substituir o GitHub. É o mais próximo em termos de uso e fácil de migrar. Muitos projetos de código aberto já estão usando o GitLab incluindo 100 mil empresas e instituições como Bayer, NASA, Sony, Comcast e Nasdaq. Repositórios privados gratuitos.

<p>Livre</p> <p>Desenvolva com uma equipe de qualquer tamanho</p> <p>\$ 0</p> <p>/ usuário / mês</p> <p>Comece agora</p> <p>Inclui</p> <p>Todos os estágios do ciclo de vida DevOps</p> <p>Traga seus próprios corretores de CI</p> <p>Traga seu próprio ambiente de produção</p> <p>400 CI / CD minutos</p> <p>Compare recursos</p>	<p>Bronze / Starter</p> <p>Controle o que entra em produção</p> <p>\$ 4</p> <p>/ usuário / mês</p> <p>(USD, cobrado anualmente a \$ 48)</p> <p>Compre Agora</p> <p>Todos os benefícios do Free +</p> <p>Gerenciamento de projeto de equipe única</p> <p>Mais controle sobre seu código</p> <p>Suporte no próximo dia útil</p> <p>2.000 minutos CI / CD</p> <p>Compare recursos</p>	<p>Silver / Premium</p> <p>Planeje várias equipes</p> <p>\$ 19</p> <p>/ usuário / mês</p> <p>(USD, cobrado anualmente em \$ 228)</p> <p>Compre Agora</p> <p>Todos os benefícios do Bronze / Starter +</p> <p>Gerenciamento de projetos entre equipes</p> <p>Controles de integridade de código</p> <p>Suporte multirregional</p> <p>Suporte prioritário</p> <p>10.000 minutos CI / CD</p> <p>Compare recursos</p>	<p>Gold / Ultimate</p> <p>Proteger e monitorar a produção</p> <p>\$ 99</p> <p>/ usuário / mês</p> <p>(USD, cobrado anualmente em \$ 1188)</p> <p>Compre Agora</p> <p>Todos os benefícios de Silver / Premium +</p> <p>Gestão de portfólio em toda a empresa</p> <p>Segurança avançada de aplicativos</p> <p>Insights de nível executivo</p> <p>Automação de conformidade</p> <p>Usuários convidados grátis</p> <p>50000 minutos CI / CD</p> <p>Compare recursos</p>
---	---	--	--

- O BitBucket é da Atlassian, dona do Trello. Totalmente integrado a outras ferramentas de gerenciamento Jira, HipChat e Confluence. Escolha certa para grandes empresas que já usam essa plataforma. É gratuito para projetos com até cinco membros.

	Free	Standard	Premium
	Começar	Faça o teste	Faça o teste
	0 US\$ <small>Para sempre</small>	3 US\$ <small>Preço inicial/usuário/mês</small>	6 US\$ <small>Preço inicial/usuário/mês</small>
Limite de usuários	até 5 usuários	ilimitado	ilimitado
Minutos de compilação	50 min/mês	2500 min/mês	3500 min/mês
Git Large File Storage	1 GB	5 GB	10 GB
Repositórios privados ilimitados	✓	✓	✓
Integração com Jira Software	✓	✓	✓
Integração Trello	✓	✓	✓
Implementação/integração contínuas	✓	✓	✓
Revisores de solicitações pull ilimitados	✓	✓	✓
Code Insights	3 integrações	ilimitado	ilimitado
Verificações de mesclagem	✓	✓	✓
Verificações mescladas aplicadas			✓
Permissões de implementação			✓
IP Whitelisting			✓
Verificação em duas etapas obrigatória			✓
Suporte Standard	✓	✓	✓

- SourceForge é outro nome popular em código aberto. Muitas distribuições e projetos do Linux fornecem downloads através do SourceForge. O veterano de 1999 passou por redesign, ganhou novas ferramentas e também um importador direto do GitHub.
- Launchpad é uma plataforma com visual um pouco obsoleto da Canonical, a empresa-mãe do Ubuntu. Tem sido fundamental para o rastreamento de bugs e projetos relacionados a distro. Ela está nos bastidores há anos e não ganhou popularidade.
- O Apache Allura é uma opção bastante flexível da Apache Software Foundation, uma organização sem fins lucrativos. A plataforma suporta linguagem Git, além de outras, cria páginas wiki internas e tickets para documentação e correção de bugs do códigos