INFOCORP CONSULTORIA
DIRETORIA DE QUALIDADE E GESTÃO DE PESSOAS
PROCESSO SELETIVO 2018/1

MATERIAL COMPLEMENTAR

Introdução a GIT com GITHUB

infocorp

Um sistema de controle de versão

UM SISTEMA DE CONTROLE DE VERSÃO?

O que é? Na função prática da Ciência da Computação e da Engenharia de Software, é um software que tem a finalidade de gerenciar diferentes versões no desenvolvimento de um documento qualquer[2.1].

Esses sistemas são comumente utilizados no desenvolvimento de software para controlar versões e histórico de mudanças no código[2.2].

POR QUE É IMPORTANTE?

O controle de versão é importante para registrar mudanças feitas em um arquivo ou um conjunto de arquivos ao longo do tempo de forma que possa ser feita a recuperação[2.3].

Mais informações sobre o conteúdo deste slide você pode encontrar clicando nos links em destaque.

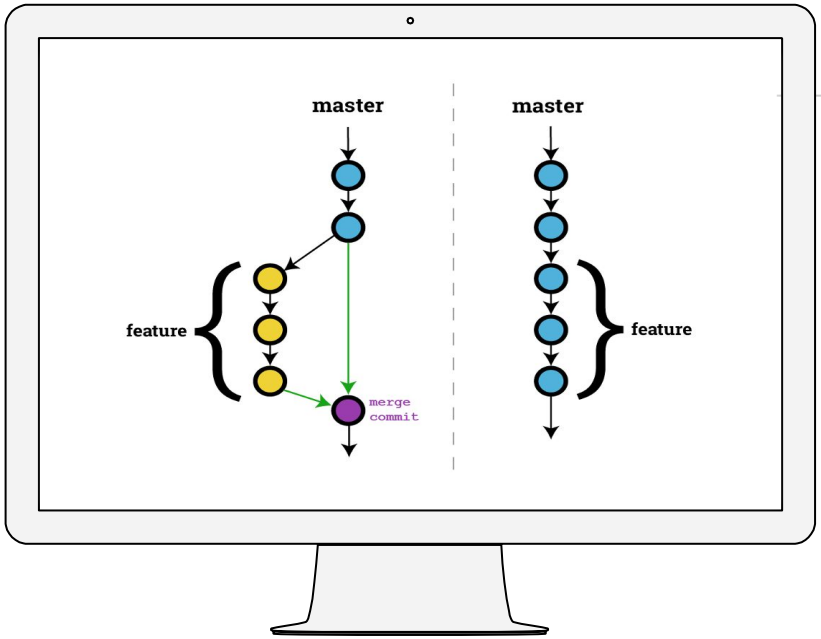
Os principais software utilizados

- CVS
- Mercurial
- Git
- SVN

Todos esses são soluções livres, para instalação e utilização sem custo.

Um resumo sobre CVS

Criado em 1990, o CVS é um sistema de versionamento centralizado, os usuários resgatam os arquivos do servidor[4.1].



Um pouco mais ...

Alguns recursos

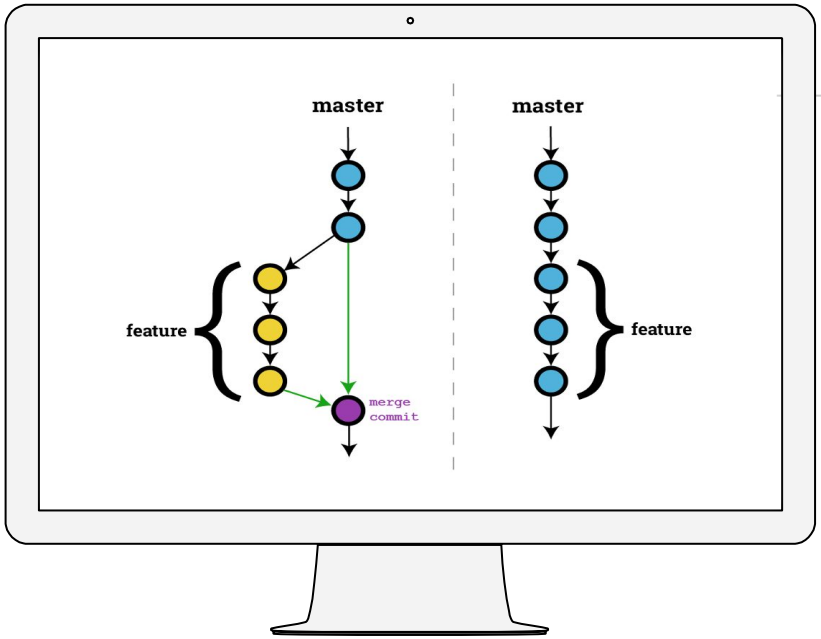
- Armazenamento centralizado
- Histórico universal de revisões
- Análise de diferenças entre versões
- Trabalho cooperativo
- Ramificação

Desvantagens em reação aos outros

- Diretórios não são “versionados”
- Não gerencia cópias e moções de arquivos (Não gerencia cópias e moções de arquivos)
- Revisões independentes para cada arquivo

Um resumo sobre Mercurial

Lançado em 2005, o Mercurial é um sistema de controle de versão distribuído, os usuários possuem cópia do repositório em cada computador[6.1][6.2][6.3].



Um pouco mais ...

Alguns recursos

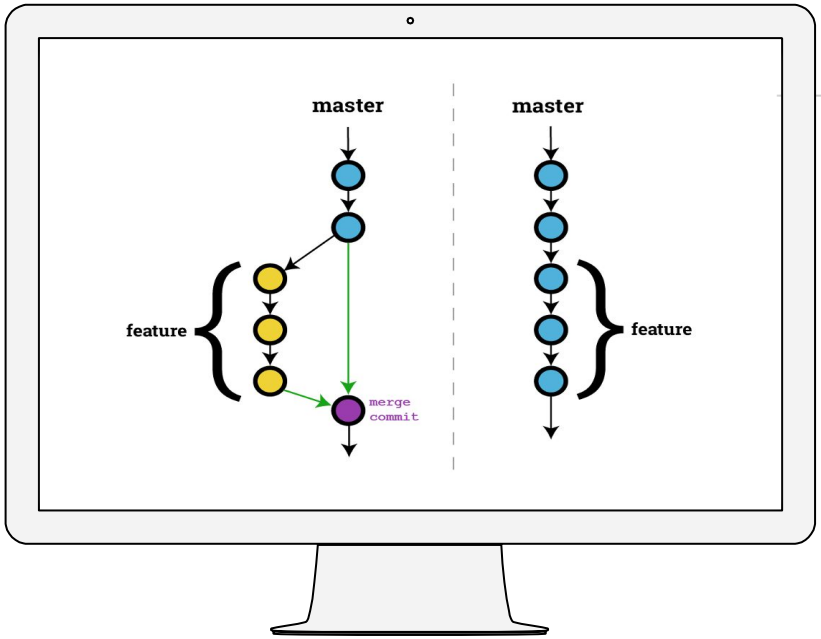
- Armazenamento distribuído
- Multiplataforma
- Interface nativa para interação e visualização dos commits
- Suporte a plugins
- Integração com outras ferramentas via plugins

Desvantagens em reação aos outros

- No geral, é mais lento que o Git
- Não é recomendável para certos perfis de projetos
- Comunidade menos expressiva, já que o número de repositórios que utilizam Mercurial é menor.

Um resumo sobre Git

Criado em 2005, o Git também é um sistema de controle de versão distribuído, os usuários possuem cópia do repositório em cada computador[8.1][8.2].



Um pouco mais ...

Alguns recursos

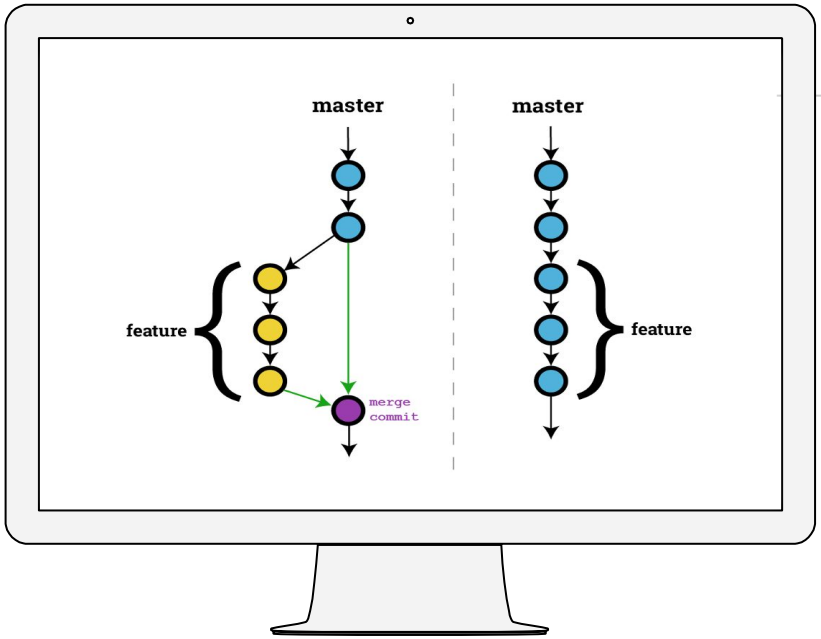
- Armazenamento distribuído
- Rapidez
- Autonomia
- Confiabilidade
- Redução de custo com servidor

Desvantagens em reação aos outros

- Maior complexidade

Um resumo sobre SVN

Criado em 2000, veio para ser o substituto do CVS, como o Git também é um sistema de controle de versão distribuído, os usuários possui cópia do repositório em cada computador[10.1][10.2].



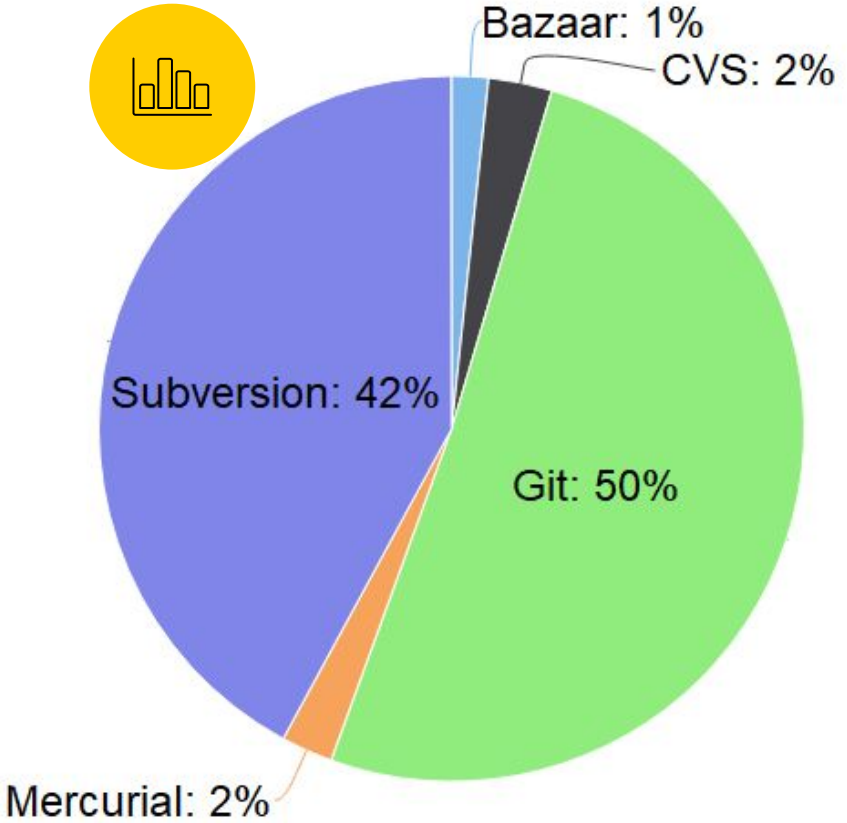
Um pouco mais ...

Alguns recursos

- Armazenamento centralizado
- Maior consistência entre os seus recursos
- Uma revisão para o repositório todo
- Escolha do protocolo de transmissão

Desvantagens em reação aos outros

- Custos com servidor centralizado



Comparação de Repositórios

A Infocorp optou por utilizar o Git em seus projetos, após analisar suas inúmeras vantagens.

Um pouco mais afundo

Vamos começar com um pouco da história do Git

O software

O Git é um sistema de controle de versão com ênfase na velocidade. O git foi inicialmente projetado e desenvolvido por Linus Torvalds para o desenvolvimento de kernel Linux, mas foi adotado por muitos outros projetos.

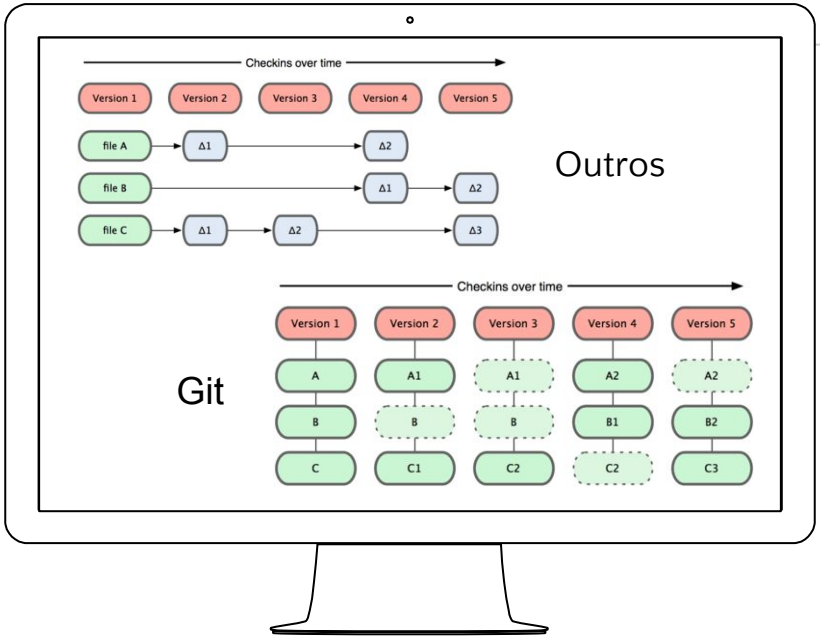
Cada diretório de trabalho do Git é um repositório com um histórico completo das revisões.

O Git é um software livre, ou seja, você pode instalar e usar no seu computador sem custo algum[15.1].

A *maior diferença entre o Git e outro VCS está na forma como o Git trata os dados. Vamos ver algumas noções básicas*

Snapshots, e não diferenças

Outros sistemas costumam armazenar dados como mudanças em uma versão inicial de cada arquivo, o Git não pensa nem armazena informações dessa forma. Ao invés disso o Git considera que os dados são como um snapshots, que são **captura de algo em um determinado instante, como em uma foto** de um mini-sistema de arquivo.

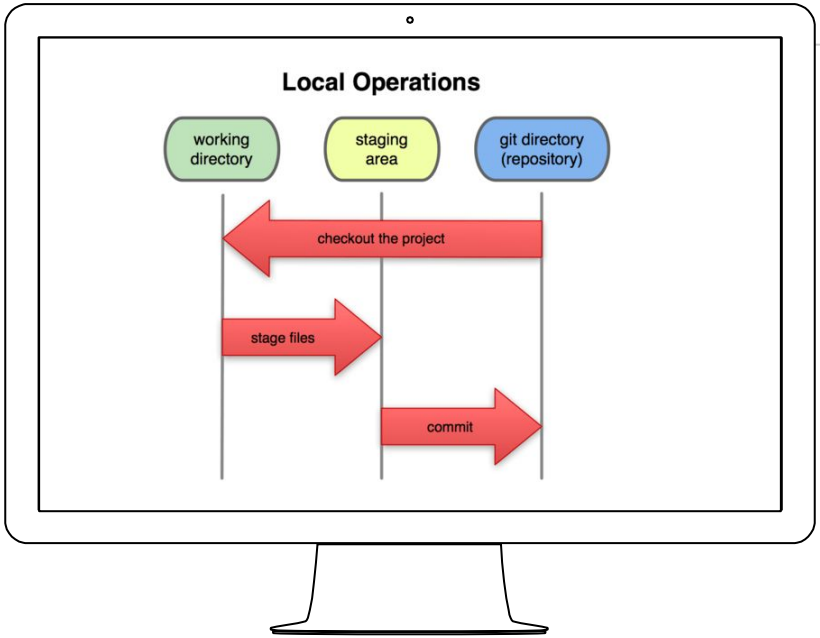


Os três estados

Git faz com que seus arquivos sempre estejam em um dos três estados fundamentais: consolidado (committed), modificado (modified) e preparado (staged).

O workflow básico do Git pode ser descrito assim:

1. Você modifica arquivos no seu diretório de trabalho.
2. Você seleciona os arquivos, adicionando snapshots deles para sua área de preparação.
3. Você faz um commit, que leva os arquivos como eles estão na sua área de preparação e os armazena permanentemente no seu diretório Git.



Principais comandos

Agora vamos ver os principais comandos Git. Lembrado que devem ser feitos pelo terminal, console, GitBash.

1

Git init

Este comando inicia um repositório

2

Git add <arquivo>

Adiciona um arquivo para ser monitorado pelo repositório

3

Git status

Verifica o status dos arquivos e em que estado estão

4

Git reset HEAD <arquivo>

Volta ao estágio anterior ao adicionamento

5

Git commit -m “Mensagem”

Grava as mudanças dos arquivo

6

git checkout -b branch

Cria uma ramificação no repositório

7

git checkout master

Volta para o branch master

8

git merge nome-do-branch

"Jogando" o branch criado no branch master

9

git clone url-do-projeto

Clona um repositório remoto

10

git pull

Sincroniza com o repositório remoto

10

git push origin master

Envia o projeto para o repositório remoto

Processo de configuração para

Windows
Link

Linux
Link

Próximo slide

Alguns links úteis

Começando com o Git:

- Fazer download do Git.
- Guia de instalação em algumas plataformas.
- Um guia prático aqui.

Agora vamos ao Github

O QUE É O GITHUB?

Criado em 2008 o Github é um plataforma de hospedagem de código usando o Git. O Git permite que programadores, utilitários ou qualquer usuário contribuam com projetos em qualquer lugar no mundo[33.1].

POR QUE USAR O GITHUB?

Existem outros sites onde podemos colocar nossos projetos, mas o que torna o Github interessante são recursos de rede social.

No Github podemos criar repositórios públicos e privados para os projetos, seguir outros desenvolvedores, baixar projetos, modificar projetos, receber atualizações de modificação de projetos etc.

Crie sua conta no Github, acesse https://github.com/join .

Creditos

Agradeço a todos aqueles que contribuíram para a construção desse material, você também pode contribuir, basta entrar em contato (próximo slide):

- Template da apresentação por SlidesCarnival

Obrigado!

Alguma pergunta ou sugestão ?

Você pode me encontrar por meio de:

- +55 65 9 9609-5847
- jorgerodrigues9@outlook.com
- suporte@infocorp.ic.ufmt.br