

Oficina de GIT

Eu

Trabalha em... casa.

Função: dona de casa e desempregado.

Contato:

Twitter: /th1agofm

Github: /thiagofm

E-mail: t@art-is-t.me

Instalação

- Sudo apt-get install nano curl git-core
- Git config - -global user.name "Bill Gates"
- Git config - -global user.email
tiobill@microsoft.com
- Git config - -global core.editor "nano"

Github

- Crie uma conta
- Adicione sua chave pública(SSH) na sua conta criada
- Crie seu repositório no github (ele te dá instruções de como adicionar seus arquivos atuais nele)

Sistemas de controle de versão

GIT

GIT - História

Desenvolvido pelo criador do Linux, Linus Torvalds para substituir o BitKeeper(que era pago), e anteriormente usava o CVS (predecessor do SVN)

GIT - Características

- Desenvolvimento distribuído
- Suporte para desenvolvimento não-linear
- Muito rápido mesmo para projetos extensos
- Compatível com vários protocolos/sistemas

Projetos que usam

- Qualquer projeto sério 😊

Erm... brincadeira.

GIT – Projetos SÉRIOS que usam

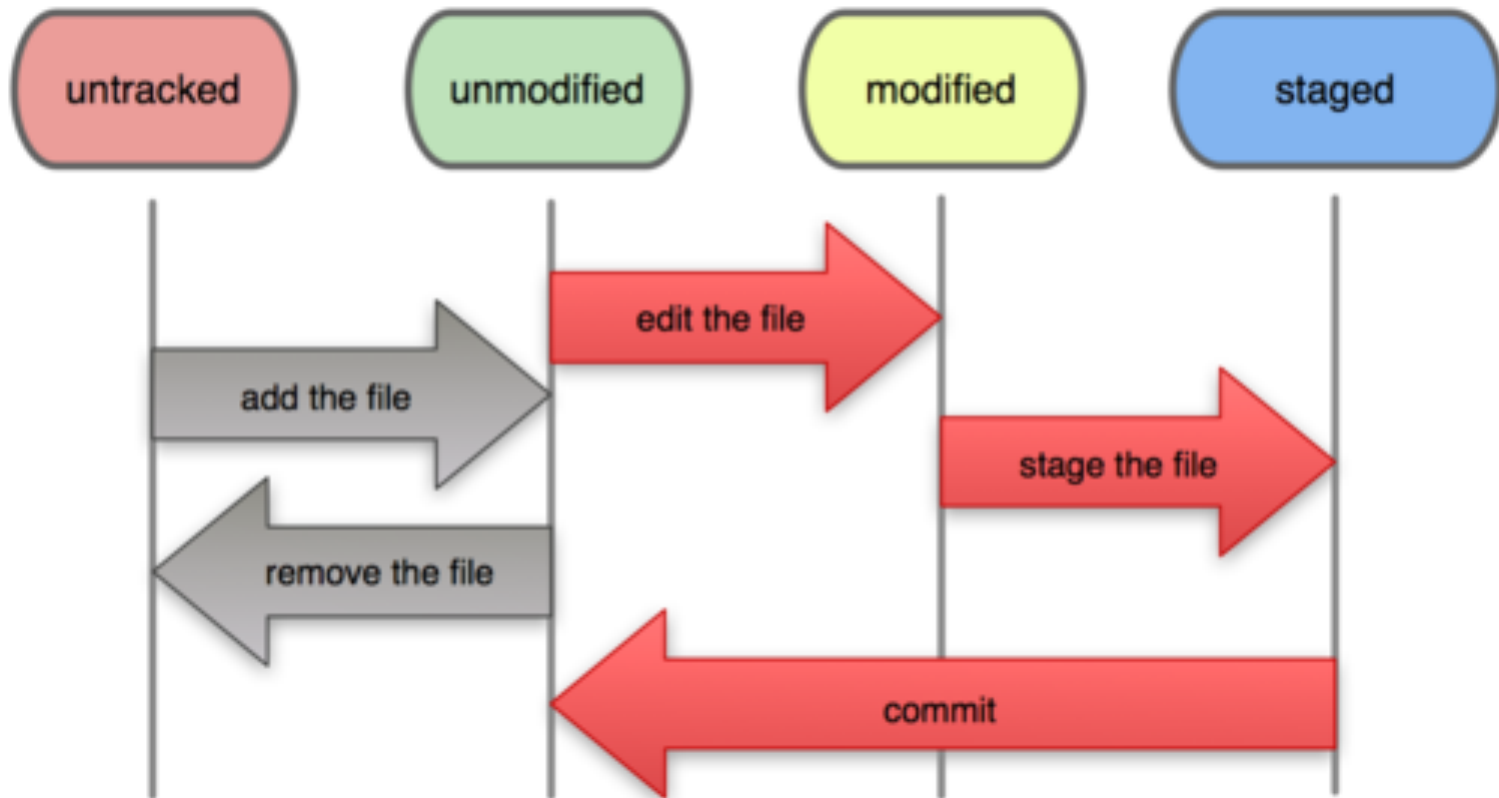
- Linux
- PHP (cakePHP e frameworks...)
- Ruby (ruby on rails)
- Facebook
- Twitter
- Microsoft (windows azure)
- Redhat
- ...

Por que devo usar?

- Trabalhar em equipe
- Desfazer erros
- Ver diferenças entre uma e outra modificação
- ...

Como funciona?

File Status Lifecycle



mão na massa

Adicionando git ao projeto

- Cd diretorio
- Git init

Ver estado do diretório atual do git

- Git status

Adicionar arquivos modificados para um commit

- `Git add .` (adiciona todos arquivos modificados)
- `Git add nomedoarquivo1 nomedoarquivo2 nomedoarquivo3` (seleciona arquivos modificados)

Fazer um commit

- `Git commit -m "Adicionei o README"`
- `Git commit -a -m "Adicionei o README"` (adiciona todos os arquivos modificados e dá um commit)

Ver últimos commits

- `Git log`
- `Git log - -oneline` (formatado com um commit em cada linha)

Ver todas modificações no repositório

- Git reflog

Removendo um arquivo do stage

- `Git reset HEAD nome_do_arquivo1
nome_do_arquivo2`

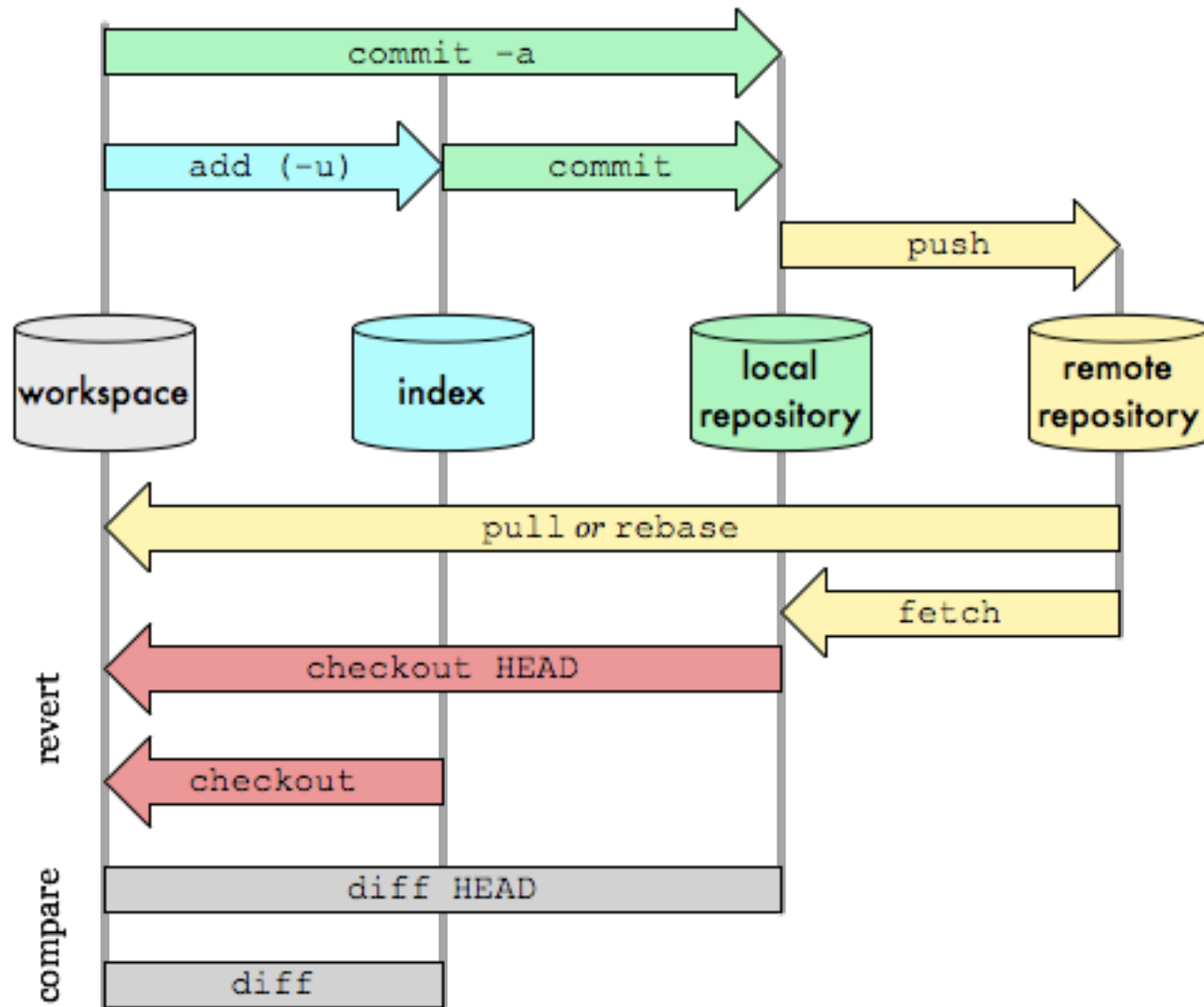
Adicionando um arquivo para ser removido quando o commit ocorrer

- `Git rm nomedoarquivo`
- `Git add -u` (adicionar arquivos também removidos ao staging)

Vamos trabalhar juntos?

Git Data Transport Commands

<http://osteele.com>



Branchs: o que são? Por que existem?

Criando um novo branch

- `Git branch nomedobbranch` (somente cria o branch)
- `Git checkout -b nomedobbranch` (cria o branch e vai para ele)

Mudando para outro branch

- `Git checkout nomedobranh`

Removendo um branch

- `Git branch -d nomedobranh`

Ok, terminei a feature que estava fazendo no meu branch e agora quero juntar ela ao branch master, como faço?

- `Git checkout master`
- `Git merge nome domeu branch`

Beleza, entendi... mas como minha equipe recebe minhas modificações?

Repositorios remotos

- `Git remote nomedorepo endereçodorepo`

Beleza, mas onde posso hostear meu repositório para ter esse endereço de repo?

- Use algum serviço como o github(grátis para repositórios públicos) ou bitbucket(grátis para repositórios privados).

Github

- Crie uma conta
- Adicione sua chave pública(SSH) na sua conta criada
- Crie seu repositório no github (ele te dá instruções de como adicionar seus arquivos atuais nele)

Recebendo modificações de um repositório remoto

- `Git pull nomedorepo nomedobbranch`

Enviando modificações para o repositório remoto

- `Git push nomedorepo nomedobbranch`
- `Git push -u nomedorepo nomedobbranch` (diz para o github qual é seu repositório upstream, ou seja, o centralizado do Github por exemplo... permitindo você usar `git push` e `pull` sem especificar o `nomedorepo` e `nomedobbranch` posteriormente)

Git fetch

- No git pull, recebemos as modificações logo no nosso repositório e ele junta tudo com o estado atual do nosso repositório(faz o merge).
- No git fetch, ele só baixa as modificações, esperando você dar um merge.
- Resumindo, um git pull é equivalente a se dar um git fetch e git merge.

Como saber o que mudou de um commit X a um commit Y?

- `Git diff <version hash commit X> <version hash commit Y>`

Como usar o Github para contribuir
com open source e reportar bugs?

Reportando bugs

- Entre no repositório desejado
- Crie uma issue com uma descrição muito boa com uma maneira de reproduzi-lo(caso possível)

Descobri um bug que sei consertar,
como faço?

Faça um pull request 😊

- Dê um fork no repositório 'mestre'
- Faça o commit no seu repositório local
- De um push para seu repositório do github
- Escolha o commit e envie-o como um pull request

Dicas:

- Se atenha ao coding style do repositório
- Faça boas descrições de tudo, o maintainer (cara que administra o repositório) não é responsável em arrumar o problema.
- Não poste dúvidas nas issues, use o stackoverflow ou derivados para isso.
- De um pull request para cada bug, erro ou feature, preferencialmente com um único commit squasheado(mais disso adiante).

Arquivo .gitignore

A remove possibilidade de adicionar os arquivos/diretórios listados ao commit.

Sessão: Fiz merda, e agora?

Doutor, fiz uma merda em um arquivo e não committei-o ainda, mas gostaria da versão antes de eu fazer a merda.

- `Git checkout nome do arquivo`

Doutor, as modificações que colocamos em produção hoje estão uma droga e o chefe está na nossa orelha para darmos um rollback ou a cobra vai fumar.

- `Git revert <version hash>`

Doutor, criei 19 commits tentando fazer algo funcionar em produção por pressão do chefe... e agora a lista de commits está muito feia.

- `Git rebase <hash version do commit anterior ao primeiro dos 19 commits> -l`
 - Abrirá o editor configurado no git(vim, nano...) após o commando.
 - Escreva squash em todos os commits que quer juntar no lugar de pick... deixe UM commit com pick.

Doutor, fui dar esse rebase passado ai,
mas deu merda e ele não deixa eu dar
outro rebase e diz que “Interactive
rebase already started”

- `Git rebase - -abort`

Voltar a uma determinada versão do repositório

- `Git checkout <version hash>`

É “só” 😊.

Podem entrar em contato comigo por
e-mail ou interagir no twitter
(@thi1agofm) caso tenham dúvidas
de algo que eu saiba.

Nota: o google manja muito mais que
eu.