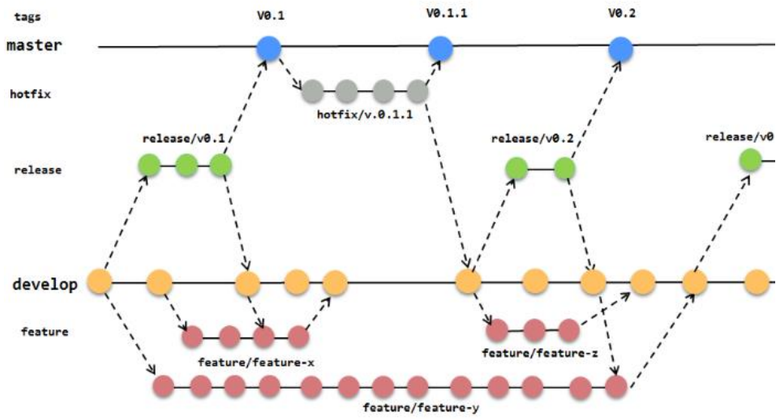
Eduardo Namba
Arquiteto de Soluções Sênior na Via Varejo Online - GPA



Ferramentas de configuração (define quem fez as atualizações)

$ git config –global user.name “[nome/nick do desenvolvedor]“
Define o nome ou o apelido de quem realiza as alterações, o nome é enviado ao GIT e descreve quem realizou a alteração de maneira simples.
$ git config –global user.email “[e-mail do desenvolvedor]“
Define o e-mail de quem realiza as alterações, o e-mail é enviado ao GIT e descreve quem realizou a alteração de maneira simples.
$ git config –global color.ui auto
As interações por linha de comando ficam coloridas com esse comando.

Criar Repositórios (novo repositório ou obtém de um já existente)

$ git init [nome do projeto]
Cria um novo repositório local com um nome já definido
$ git clone [url]
Baixa um projeto e os históricos de versões

Fazendo atualizações para entregar uma funcionalidade

$ git branch
Lista todos os branches (termo para definir versões funcionais)
$ git branch [nome de uma nova branch]
Cria uma nova branch com o nome informado
$ git checkout [nome de uma branch existente]
Altera conteúdo do local de trabalho para o snapshot da branch informada
$ git merge [nome de uma branch existente]
Combina a branch informada com a branch atual do git.
$ git branch -d [nome de uma branch existente]
Remove a branch informada

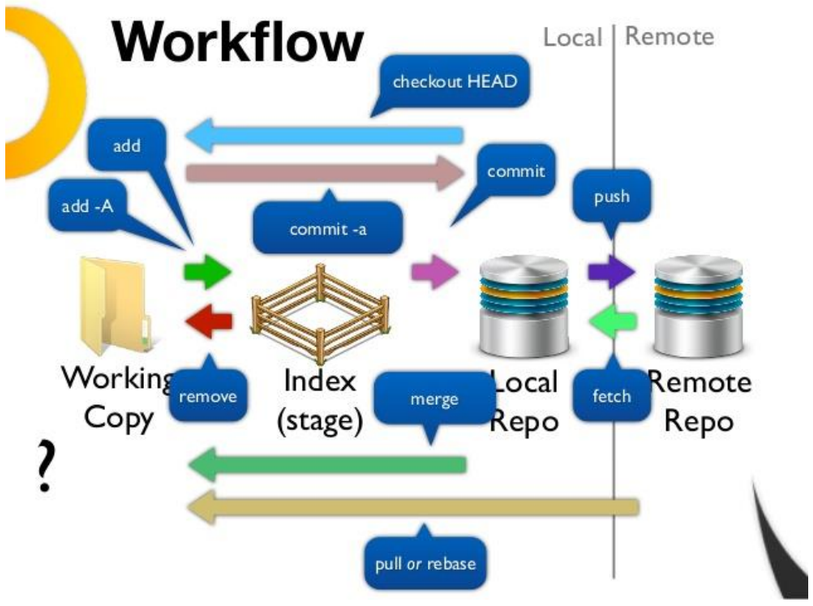
Tentei em poucas linhas descrever os principais comandos no git
Fonte https://training.github.com/kit/downloads/github-git-cheat-sheet.pdf

Fazendo atualização pontual (revisão de alterações commit manual)

$ git status
Lista todas as alterações que precisam ser comitadas
$ git diff
Apresenta as diferenças entre arquivos que não estão staged
$ git add [files]
Adiciona o arquivo para o grupo (staged) que será comitado
$ git diff –staged
Apresenta as diferenças entre os arquivos staging e a ultima versão.
$ git reset [file]
Remove arquivo do grupo (staged) comitado mas preserva conteúdo.
$ git commit -m “[descrição da alteração]“
Gera um snapshot que pode ser utilizado futuramente para roolback por exemplo. Isso não gera uma versão apenas um snapshot, isso é importante para que as alterações sejam documentadas. Essa operação é realizada apenas localmente.

Remover arquivos

$ git rm [file]
Remove o arquivo do local de trabalho e do stage
$ git rm –cached [file]
Remove o arquivo do stage mas não remove do local de trabalho
$ git mv [nome do arquivo original] [novo nome do arquivo]
Altera o nome do arquivo local e na area do stage, pronto para comitar.



Removendo arquivos e diretórios de commit acidental/indevido

*.log
build/
temp-*
Esse deve ser o conteúdo do .gitignore, nesse arquivo ficam todos os diretórios, arquivos ou padrões de arquivos que não devem ser comitados.
$ git ls-files –other –ignored –exclude-standard
Lista todos os arquivos ignorados nesse projeto

Como salvar alterações incompletas sem comitar

$ git stash
Seria como um commit temporário não oficial.
$ git stash pop
Recupera os arquivos que foram salvo pelo stash
$ git stash list
Lista todos os stash realizados e não recuperados
$ git stash drop
Descarta os stash mais recente realizado.

Histórico

$ git log
Lista o histórico das versões da branch atual
$ git log –follow [file]
Lista histórico de versão de arquivo, incluindo se também foi renomeado
$ git diff [nome da primeira branch] [nome da segunda branch]
Apresenta a diferença entre as duas branch
$ git show [nome ou código de um commit]
Apresenta todos os detalhes as mudanças de um commit especifico

Rollback dos commits realizados

$ git reset [nome do commit_1]
Desfaz todos os commits após o commit_1, mantendo as alterações locais
$ git reset –hard [nome do commit_1]
Descarta histórico de mudanças após o commit_1 e atualiza a área de trabalho com a versão do commit_1

Atualizando o servidor com os commits locais

$ git fetch [geralmente utilizam o nome “origin”]
Faz download do histórico do repositório, na imagem seria do “origin”
$ git merge [geralmente utiliza o nome “origin”]/[nome de uma branch]
Combina o branch do origin com a branch local
$ git push [geralmente o nome “origin”] [o nome da branch remota]
Faz upload de todo o conteúdo local para o servidor do Git
$ git pull
Faz o download de todo histórico do repositório incluindo as mudanças

Publicado originalmente em https://www.linkedin.com/pulse/guia-r%C3%A1pido-para-usu%C3%A1rio-de-git-eduardo-namba/?trk=v-feed