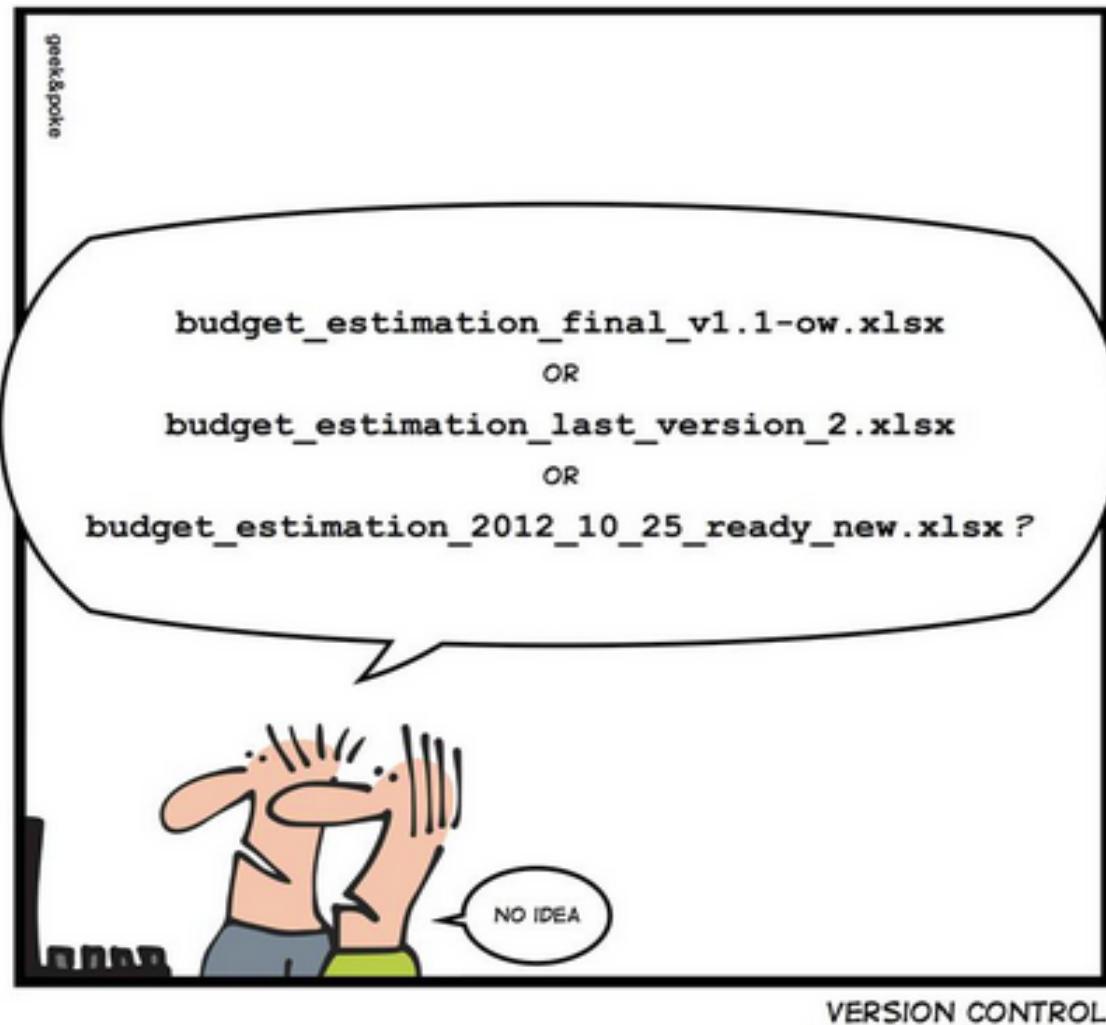




Programa Comercial II

Controle de versão - GIT

SIMPLY EXPLAINED



VERSION CONTROL

O que é Git?

- Gíria britânica que significa “unpleasant person” (pessoa desagradável)
- Torvalds responde: “I'm an egotistical bastard, and I name all my projects after myself. First 'Linux', now 'git'.”



3

O que é Git?

- Segundo os arquivos de readme do projeto "git" can mean anything, **depending on your mood**.
 - random three-letter combination that is pronounceable, and not actually used by any common UNIX command. The fact that it is a mispronunciation of "get" may or may not be relevant
 - "global information tracker": you're in a good mood, and it actually works for you. Angels sing, and a light suddenly fills the room
 - "goddamn idiotic truckload of sh*t": when it breaks



Quando surgiu?

Version	Original release date	Latest version	Release date
0.99	2005-07-11	0.99.9n	2005-12-15
1.0	2005-12-21	1.0.13	2006-01-27
1.1	2006-01-08	1.1.6	2006-01-30
1.2	2006-02-12	1.2.6	2006-04-08
1.3	2006-04-18	1.3.3	2006-05-16
1.4	2006-06-10	1.4.4.5	2008-07-16
1.5	2007-02-14	1.5.6.6	2008-12-17
1.6	2008-08-17	1.6.6.3	2010-12-15
1.7	2010-02-13	1.7.12.4	2012-10-17
1.8	2012-10-21	1.8.5.6	2014-12-17
1.9	2014-02-14	1.9.5	2014-12-17
2.0	2014-05-28	2.0.5	2014-12-17
2.1	2014-08-16	2.1.4	2014-12-17
2.2	2014-11-26	2.2.3	2015-09-04
2.3	2015-02-05	2.3.10	2015-09-29
2.4	2015-04-30	2.4.11	2016-03-17
2.5	2015-07-27	2.5.5	2016-03-17
2.6	2015-09-28	2.6.6	2016-03-17
2.7	2015-10-04	2.7.4	2016-03-17
2.8	2016-03-28	2.8.4	2016-06-06
2.9	2016-06-13	2.9.3	2016-08-12
2.10	2016-09-02	2.10.1	2016-10-03

Legend: Old version Older version, still supported Latest version Latest preview version



Como surgiu e quem escreveu as primeiras linhas?

- Bitkeeper denunciou alguns desenvolvedores do kernel do Linux por realizar engenharia reversa
- Com isso, após a liberação do kernel versão 2.6.12-rc2, Torvalds, em abril de 2005, começou a escrever o controle de versão para manter o código do kernel
- Escrito em Perl, C e vários shell scripts



Onde está o Git hoje?

- Mantido por quase 1000 programadores
- Repositório contendo o código fonte completo do Git:
 - <https://github.com/git/git>



Quem utiliza?



- <https://git-scm.com/>



Sites para manter seus repositórios

- <https://bitbucket.org/>
- <https://github.com/>



Instalando o Git

- Linux
 - `sudo apt-get install git gitk`
- Windows
 - <http://msysgit.github.com>
- Mac OS x
 - <http://git-scm.com/download/mac>



Git Básico

Comandos Locais

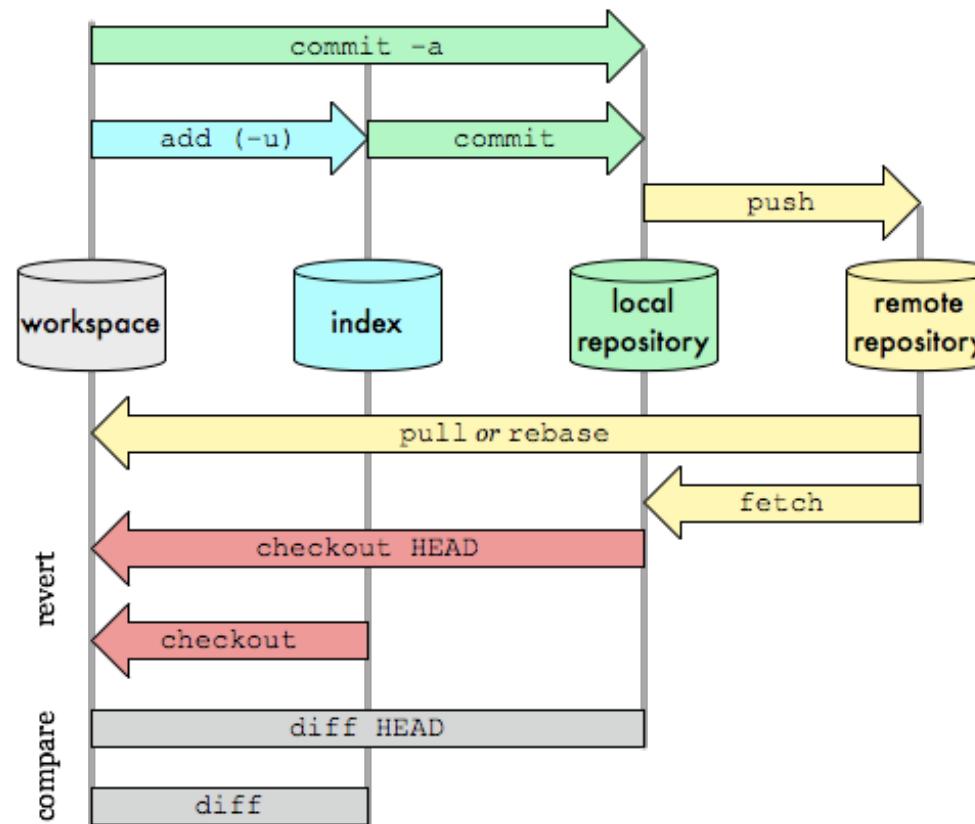


12

Git Local

Git Data Transport Commands

<http://osteale.com>



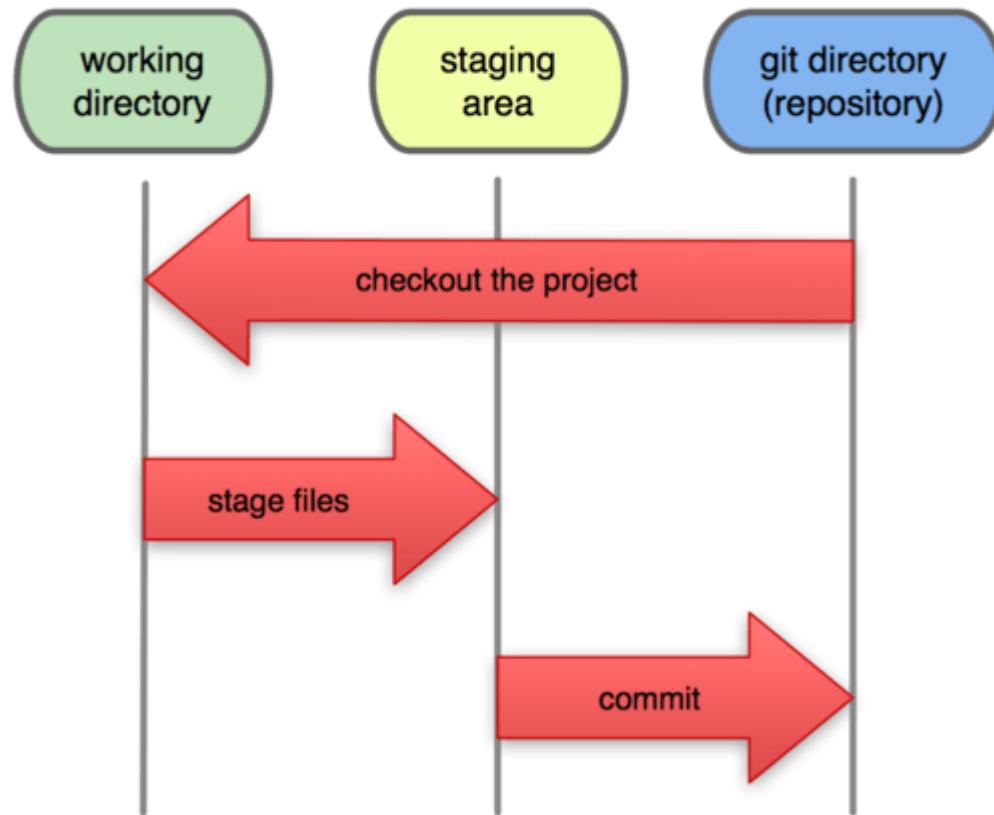
Os três estados

- Git tem três principais estados:
 - **Committed**
 - As alterações estão salvas no banco de dados, estão no repositório.
 - **Modified**
 - O arquivo foi alterado mas não foi realizado o commit. O arquivo está no diretório de trabalho.
 - **Staged**
 - O arquivo foi marcado como modificado e entrará no próximo commit. O arquivo está na área de estágio.



Estados

Local Operations



15

Configuração Inicial

- O Git necessita de uma configuração inicial, sem essa configuração ele não fará nenhuma alteração.
- As configurações básicas são:
 - Nome do usuário
 - **git config --global user.name "Seu Nome"**
 - E-mail do usuário
 - **git config --global user.email seuemail@seuemail**



Estado do repositório

- É sempre útil verificar o estado do repositório antes de alguma operação.
- O comando retorna o nome do branch, se tem arquivos alterados, arquivos para serem adicionados entre outras informações.
- Verificar o estado de um repositório
 - **git status**

Adicionar arquivos

- Se você quer deixar um arquivo sob controle de versão, deve adicionar ele para que o Git possa rastreá-lo.
- Adicionar um único arquivo
 - **git add [arquivo]**
- Adicionar todos os arquivos do projeto
 - **git add .**
 - Obs: Podem ser utilizados os curingas do Linux.



Remover arquivos

- Se um arquivo for removido ele não será mais rastreado pelo Git. Um arquivo pode ser removido somente do controle de versão ou do sistema e do controle de versão.
- Remove o arquivo da área de trabalho e da área de trabalho
 - **git rm <file>**
- Remove o arquivo do controle de versão mas não da área de trabalho
 - **git rm --cached [file]**



Mover ou renomear arquivos

- Um arquivo pode ser movido ou renomeado, para isso o comando utilizado é o git mv
- Altera o nome/local do arquivo e prepara (insere na área de estágio) para um commit
 - **git mv <file_old> <file_new>**

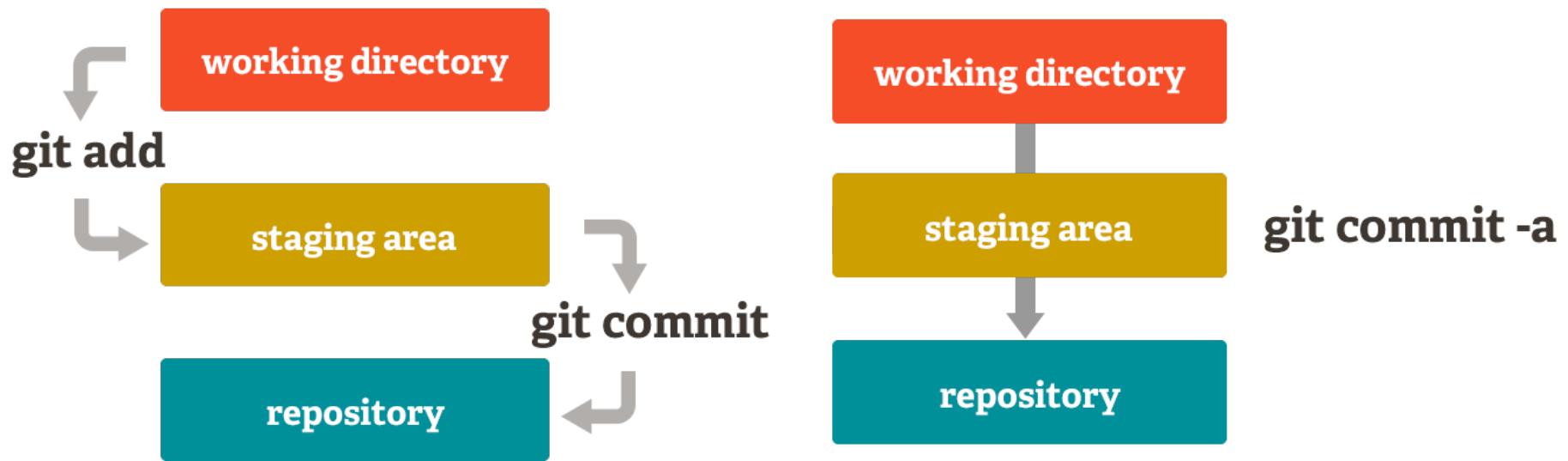


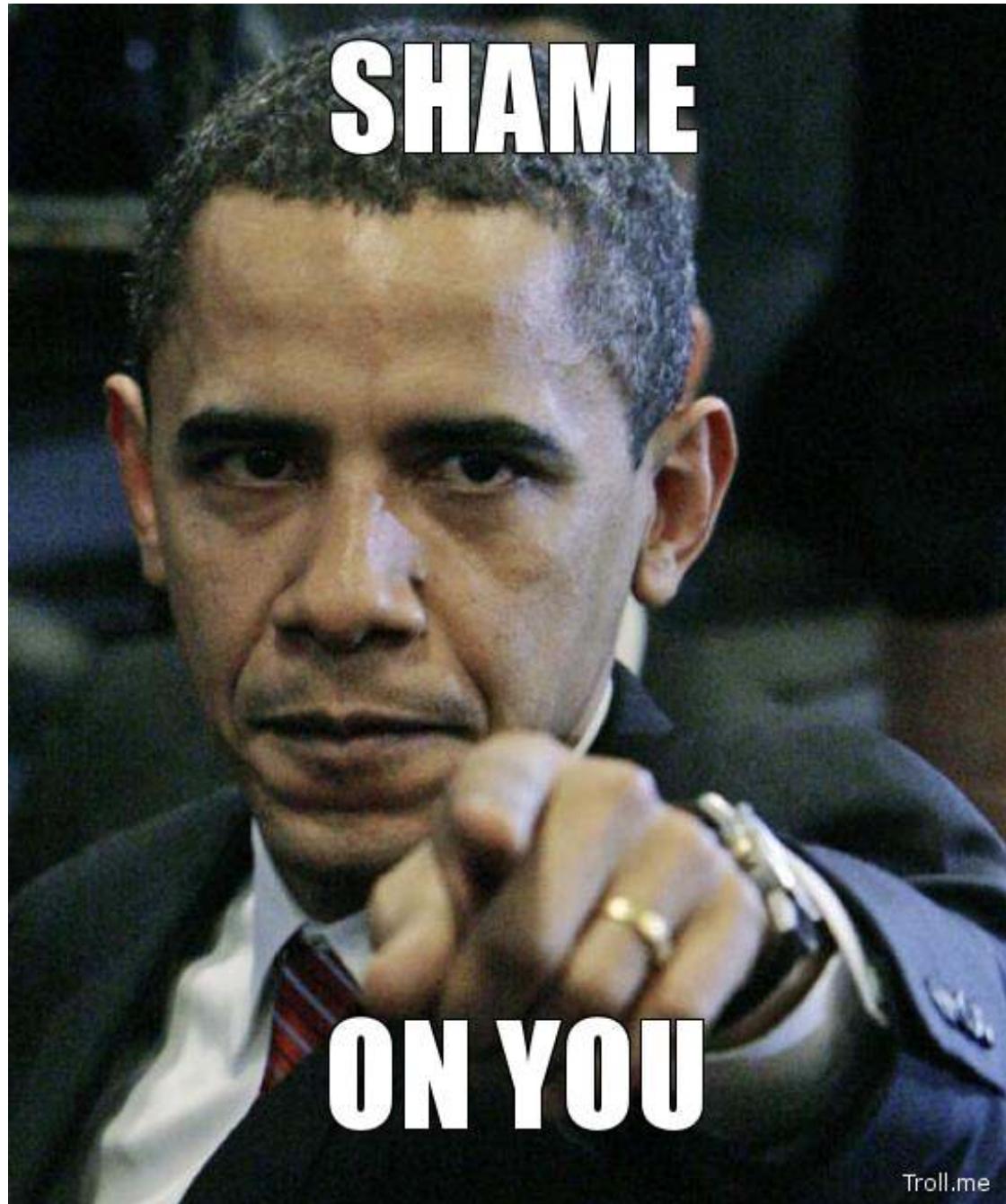
Commit

- Commit é a alteração que coloca um ou mais arquivos no repositório (committed), é o que realmente salva a alteração.
- Para fazer um commit é necessário adicionar o arquivo que contém a alteração, enviar o arquivo do diretório de trabalho para a área de estágio.
- Se a alteração for referente a somente um arquivo, adicione-o e faça o commit. Após adicione outro arquivo e faça outro commit. Se a alteração for referente a vários arquivos pode adicionar todos ao mesmo tempo.
- Commit da área de estágio
 - **git commit -m "Mensagem de commit"**
- Commit de todas as alterações
 - **git commit -am "Mensagem de commit"**



Diferenças do Commit





Hall of SHAME

[REDACTED] pushed to branch [verif](#) at SMDH / zr16rh08
4a1ce471 novo testcase
2631be30 novo testcase
[Compare 73e55578...4a1ce471](#)

[REDACTED] pushed to branch [verif](#) at SMDH / zr16rh08
74df5deb modificado para acabar a simulação

SMDH / zr16rh08

Commit [287bfc28143b5c69e7c3dfc51272a8182d1a9d97](#)
Authored by [REDACTED] 11 days ago
1 parent [90da01c5](#)
by [digphy](#)

update files

Showing 3 changed files with 118 additions and 256 deletions

[REDACTED] pushed to branch [bringup](#) at SMDH / rh_drv
543e638c edição no topo

[REDACTED] pushed to branch [bringup](#) at SMDH / rh_drv
91271757 edição código



Hall of SHAME

[REDACTED] pushed to branch [REDACTED] at SMDH / zr16_lp
a76360d2 Editando covergroups

[REDACTED] pushed to branch [REDACTED] at SMDH / zr16_lp
64bb8cbd Editando covergroups

[REDACTED] pushed to branch [REDACTED] at SMDH / zr16_lp
a72a204e Editando covergroups

[REDACTED] pushed to branch [REDACTED] at SMDH / zr16_lp
19c462e5 Editando covergroups

[REDACTED] pushed to branch [REDACTED] at SMDH / zr16_lp
37edfef8 Editando covergroups

[REDACTED] pushed to branch [REDACTED] at SMDH / zr16_lp
35bad81e Editando covergroups

[REDACTED] pushed to branch [REDACTED] at SMDH / zr16_lp
14de54fb Editando covergroups

[REDACTED] pushed to branch [REDACTED] at SMDH / zr16_lp
368b0a4a Editando covergroups

[REDACTED] pushed to branch [REDACTED] at SMDH / zr16_lp
ac0e9ba2 Editando covergroups

SMDH / zr16_lp

Commit 18266ef566ec3da8f6276109add94112639d2ccb

Authored by [REDACTED] about a year ago

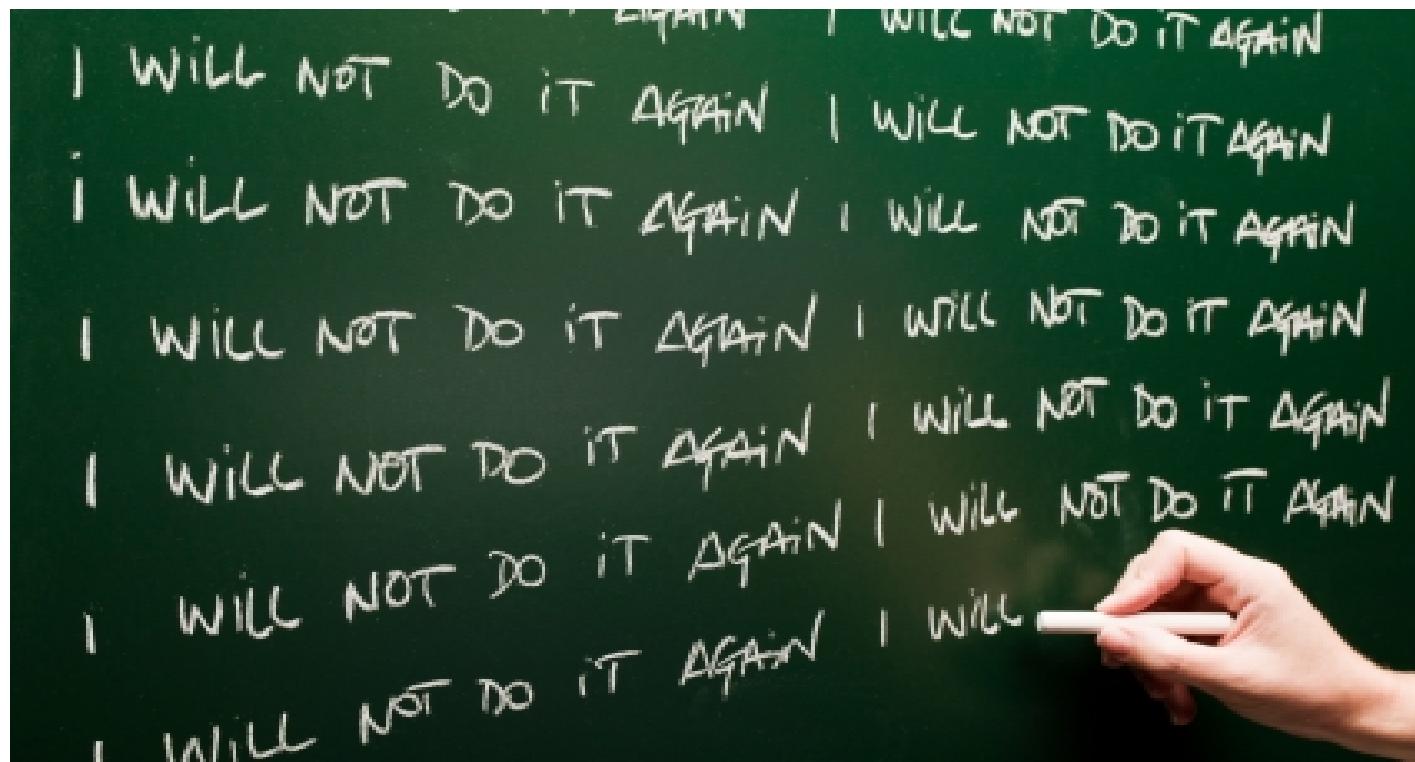
1 parent 360337e9

▶ master ...

vaiGDS

Showing 1 changed file

Hall of SHAME





Hall of FAME

SMDH / rh_drv

Commit [c5debb474a0dfc8dfaf751ee8694d0d5c3aeb003](#)

Authored by  Jorge Noval about 18 hours ago

1 parent [a4009e4f](#)

 sram

#2117 Testes simulação elétrica SRAM com corners

SMDH / zr16_lp

Commit [0a3ff12f52f0befae98d5c3cbc162b757a0fbea4](#)

Authored by  Lucas Teixeira 3 months ago

1 parent [427c7daf](#)

 master ...

Adicionadas condições de pull up/down extraídos do esquemático elétrico e análise dos PADs

Showing 1 changed file with 5 additions and 5 deletions

Hall of FAME

SMDH / rh_drv

Commit dc18d8cd8ed7fba2be01d66e708146f63a0229e4

Authored by  **Alex Silveira** 15 days ago

1 parent 96e9d482

 bringup

Adicionado esquematico e layout com novo barramento de 44 pinos

SMDH / zr16rh08

Commit 97e32a84fd620686b1a58b7d1836153da494f4e6

Authored by  **Joel Oliveira** 12 days ago

1 parent 55904d70

 verif

testes e hexas para a simulacao do spi



Ver o histórico

- O histórico do projeto pode ser visualizado pelo terminal, diretamente no Gitlab, ou com ferramentas específicas para log como o GtiK. Para um controle maior do histórico é aconselhável utilizar o Gitlab, para uma consulta rápida o terminal com comandos Git pode suprir a necessidade.
- Mostra o histórico do projeto
 - **git log**
- Lista o histórico para apenas um arquivo
 - **git log --follow [file]**
- Mostra o histórico compactado em uma linha
 - **git log --oneline**
- Mostra alterações de um commit específico
 - **git show <commit>**



Diferença de arquivos

- Para ver as diferenças entre versões pode ser utilizado o terminal, o Gitlab ou ferramentas específicas para diff. Para visualizar pequenas alterações ou poucos arquivos pode ser utilizado o terminal com comandos do Git, para um controle maior o Gitlab é mais eficiente.
- Mostra as diferenças entre a área de trabalho e o último commit
 - **git diff**
- Mostra as diferenças entre a área de estágio e o último commit
 - **git diff --cached**
- Mostra a diferença entre dois commits específicos
 - **git diff <ID1> <ID2>**



Desfazer alterações

- Se algo de errado for feito, como uma mensagem de commit errada, ou um arquivo não foi marcado para entrar no commit, Git tem comandos para alterar mensagens, adicionar arquivos e refazer o commit.
- Altera a mensagem do último commit
 - **git commit --amend**
- Move os arquivos da área de estágio para a área de trabalho
 - **git reset HEAD**
- Move um arquivo da área de estágio para a área de trabalho
 - **git reset HEAD <file>**
- **Descarta (ELIMINA)** as mudanças de um aquivo da área de trabalho
 - **git checkout HEAD <file>**



Reverter

- Uma dos benefícios de utilizar um controle de versão é poder reverter para um commit anterior, ou seja, voltar para um estado onde o projeto estava funcionado.
- Reverte para o commit ID e prepara para um novo commit
 - **git checkout <ID> <file>**



Git Remoto

Comandos



git

34

Git Remoto

- Re却itórios remotos sã versões do projeto que estã hospedados no servidõ.
- Projetos com várias pessoas envolve gerenciar esses re却itórios, enviar versões para o servidõ, obter atualizações do servidõ e compartilhar os códigos com a equipe.
- Gerenciar re却itórios remotos inclui saber como adicionar re却itórios, remover os que nã mais necessários e gerenciar os vários branches remotos.
- Um re却itório remoto é o que o comando clone baixa para o computador local.



Clonar repositório

- Clonar um repositório significa pegar todo o repositório com o todas as informações e o histórico completo.
- Clonar todo o repositório
 - **git clone [url]**



Informações sobre repositórios remotos

- Após clonar um repositório, pode-se ver informações sobre ele com o comando `remote`. As informações disponíveis são sobre o endereço ssh, branch, nome...
- O nome de um repositório remoto é sempre **origin**.
- Informações sobre o repositório remoto
 - **git remote -v**
- Inspecionar um repositório remoto
 - **git remote show origin**

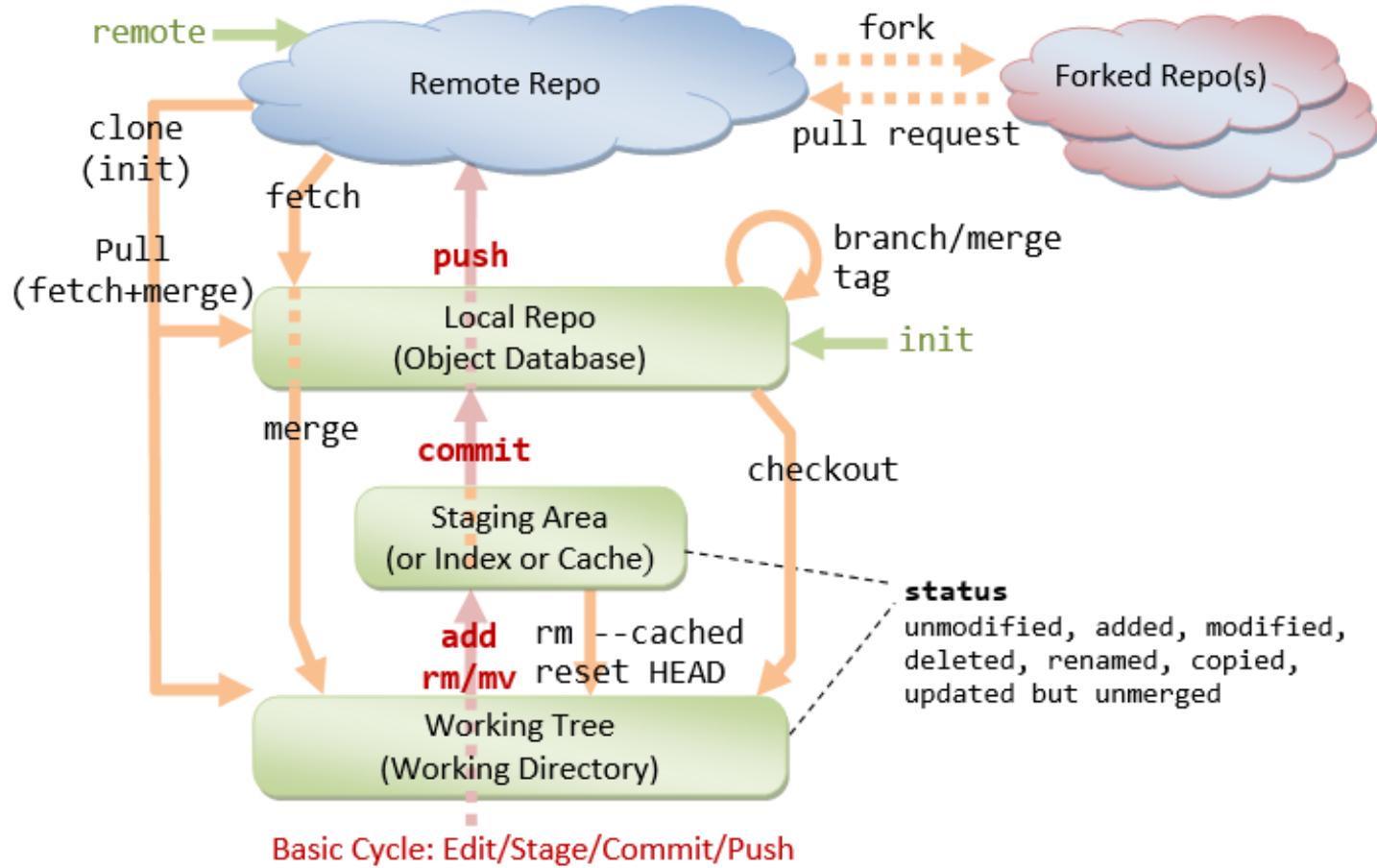


Enviar alterações para um repositório remoto.

- O comando commit faz alterações locais, para enviar essas alterações ao servidor remoto é necessário fazer um push. O comando push envia as alterações locais para o repositório remoto.
- Enviar alterações para um branch remoto
 - **git push origin codif**
 - **git push origin 1898**



Git Local - Remoto



Branches

- Branch significa divergir da linha principal de desenvolvimento e continuar o trabalho sem bagunçar com a linha principal.
- A forma que o Git trabalha com branch é uma de suas grandes vantagens.
- Branches são muito leves e rápidos de serem criados.
- Git encoraja a utilizar branch e merge no fluxo de trabalho, mesmo que isso ocorra várias vezes por dia.
- Entender essa ferramenta lhe dará um poder maior para utilizar Git que pode mudar a forma de trabalho.



Trabalhando com branches

- Criar um novo branch
 - **git branch new-branch**
- Entrar em um branch
 - **git checkout branch-name**
- Criar um branch e entrar nele
 - **git chechckout -b new-branch**
- Listar branches
 - **git branch**
 - **git branch -v**
- Listar branches remotos
 - **git branch -r**
- Criar um branch local baseado no remoto
 - **git branch -b branch-name origin/branch-name**
- Remover um branch local
 - **git branch -D branch-name**
- Remover um branch remoto
 - **git push origin :branch-name**
- Enviar um branch para o servidor
 - **git push origin branch-name**



Informações sobre Git

- Livro:
 - <http://git-scm.com/book/en/v2>
- Cheat Sheet:
 - <http://ndpsoftware.com/git-cheatsheet.html>
- Git Reference:
 - <http://git-scm.com/docs>

