

Git para Desenvolvedores

SEMABI 2018

Git para Desenvolvedores

Sandro Custódio

TI desde 1993

Servidor Público desde 1995

Na Justiça Federal desde 1999

Anos 90: Programador e Operador de CPD

Anos 2000: Sysadmin e suporte a usuário

Desde 2005: Suporte a Usuários.

Membro do Tchelinux.org desde 2010.

Em 2018: Retornado ao desenvolvimento.

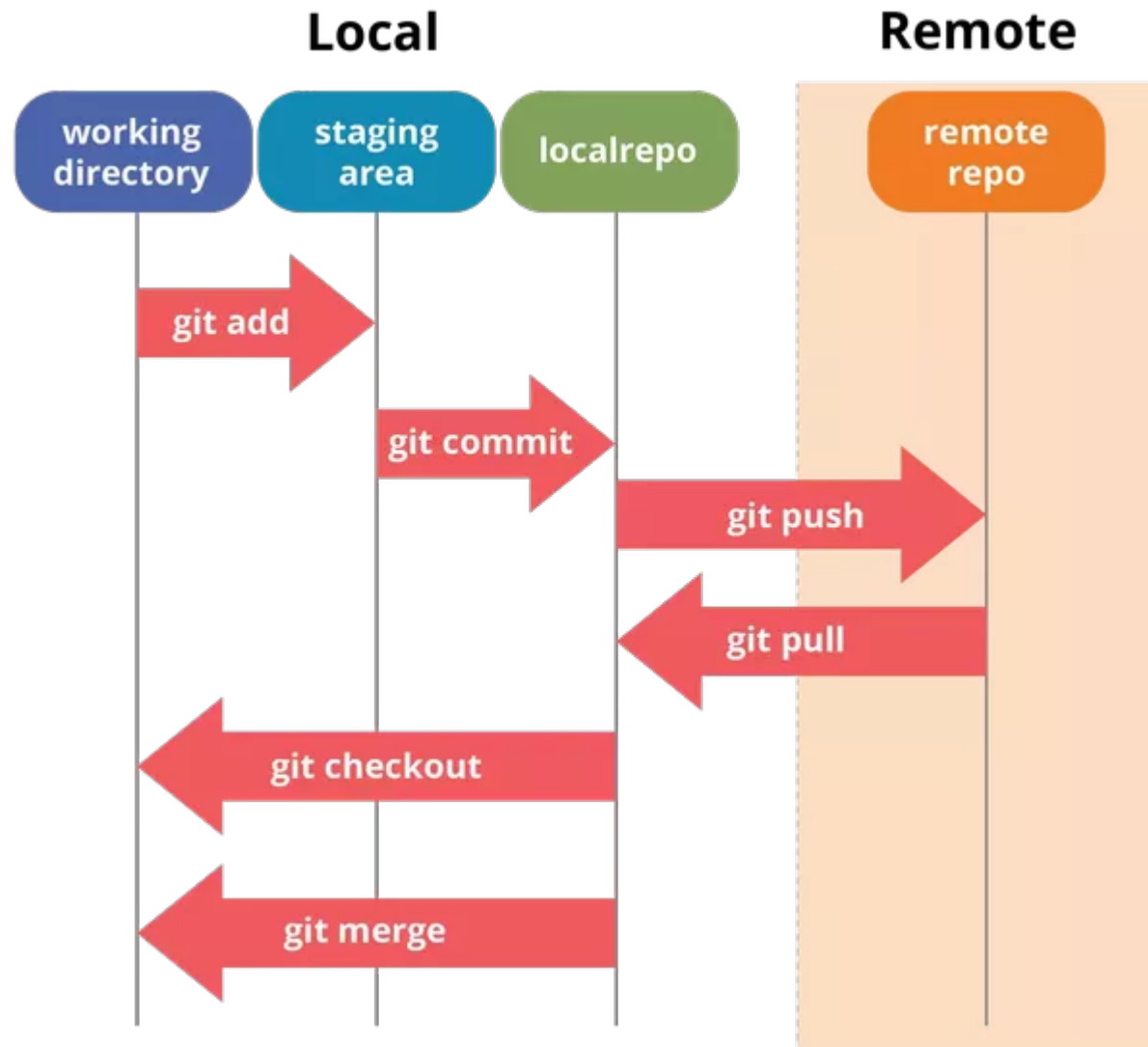
Git – O que é?

- Manter histórico de alterações no código
- Autores (Quem?)
- Data das alterações (Quando?)
- Motivo das alterações (Por que?)

Git – O que é?

- Facilita a colaboração (trabalho em equipe).
- Auxilia na identificação e resolução de problemas.
- Histórico das m..., digo, dos problemas.
- Permite fechar versões.

Git – Conceitos



Git – Conceitos

- Working directory
 - Arquivos e pastas do seu projeto.
- Staging area
 - Arquivos ou pastas modificados e selecionados para o próximo commit.
- Repositório local
 - Pasta local do git (.git) onde ficam os arquivos que contém os vários commits realizados.
- Repositório remoto
 - Pasta ou serviço remoto onde seu projeto está armazenado, nos mesmos moldes do repositório local

Git – Conceitos

- Working directory
 - Armazena arquivos e pastas do seu projeto.
 - Contém personalização, ignorando arquivos e pastas desnecessárias ao projeto.
 - Nesta camada se decide o que entrará no projeto e o que será ignorado do controle de versão dos arquivos.
 - Principais comandos:
 - \$ git status
 - \$ git add <arquivos>

Git – Conceitos

- Staging area
 - Antes do commit, os arquivos são selecionados para ser comitados
 - Comandos:
 - \$ git add <arquivo>
 - \$ git add . # adiciona todos os arquivos modificados

Git – Conceitos

- Repositório local
 - Pasta local do git (.git) onde ficam os arquivos que contém os vários commits realizados.
 - Armazena cada alteração de cada arquivo controlado do projeto.
 - Contém todos os commits de todos os participantes do projeto em todos os micros.
 - Comando básico:
 - \$ git commit ...
 - \$ git commit -m “Descricao (sem acentos) dos commit”

Git – Conceitos

- Repositório remoto
 - Pasta ou serviço remoto onde seu projeto está armazenado, nos mesmos moldes do repositório local.
 - Geralmente é um projeto num site de hospedagem de projetos git.
 - <https://github.com>
 - <https://gitlab.com>
 - Repositório numa pasta da rede local.
 - Repositório numa pasta de um pendrive.

Git – Conceitos

- Todos os repositórios (locais e remoto) contêm todos os conteúdos de todos os commits.
- Basta um repositório ok para todos serem restaurados.
- Todos os micros envolvidos ficam sempre sincronizados (atualizados).

Git - Configuração Inicial

```
$ git config --global user.name "Fulano de Tal"
```

```
$ git config --global user.email "asd@asd.com"
```

```
$ cd pasta_do_projeto
```

```
$ git init
```

Git - Comandos

Ver situação/status

```
$ git status
```

Adicionando arquivos para o próximo commit

```
$ git add <arquivos>
```

Comitando no repositório local

```
$ git commit -m "Primeiro commit do projeto."
```

Git - Comandos

Vendo diferenças

```
$ git diff ...
```

Vendo histórico de commits

```
$ git log ...
```

Vendo graficamente o histórico

```
$ gitk
```

Git - Comandos

removendo um arquivo do repositório local

... remover do working directory ...

\$ git add . # vai avisar da remoção

\$ git rm (nome arquivo) # retira do projeto

\$ git commit -m "Arquivo xyz removido"

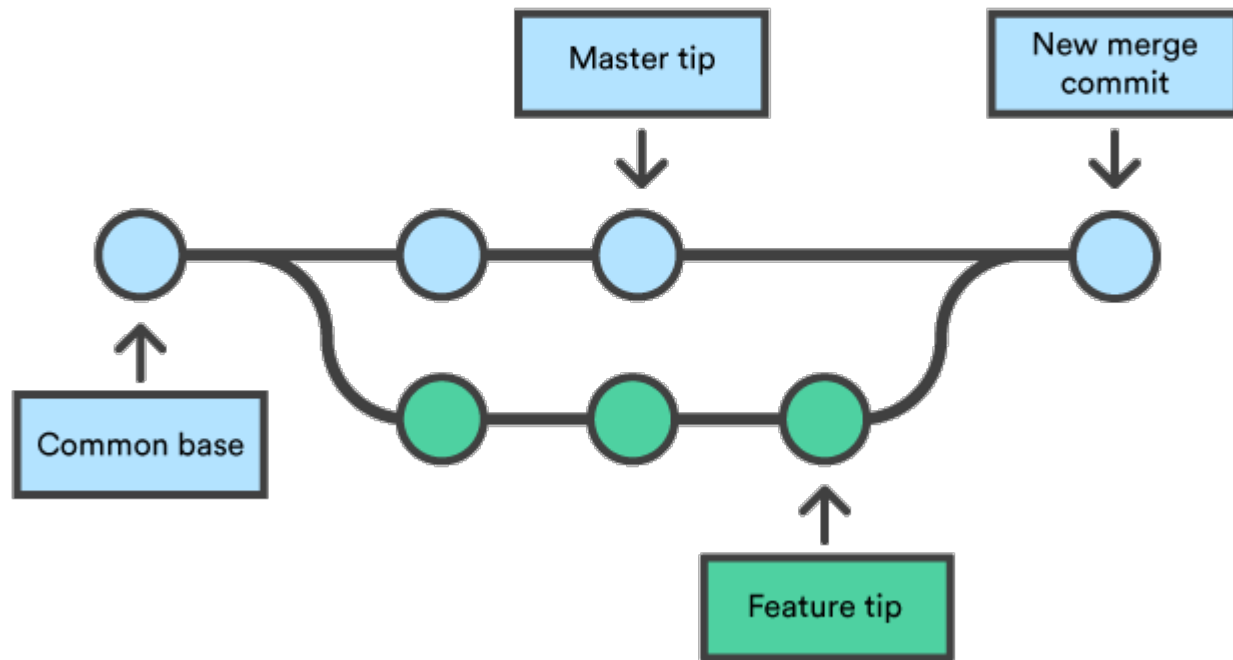
Git - Comandos

Tags (dar nome específico ao commit)

\$ git tag ... # vai avisar da remoção

\$ git show v1.0 # mostra tag

Branches



Git - Branch

cria o branch

\$ git branch teste

passa para o branch teste (sai do master)

\$ git checkout teste

cria o branch teste já fazendo o checkout

\$ git checkout -b teste

Git - Branch

para sair do branch teste e voltar ao master

\$ git status

ou informar o que foi feito até o momento como "funcoes basicas"

\$ git commit -a -m "ponto de commit para poder voltar depois"

\$ git checkout master

depois, para voltar ao teste e continuar o trabalho

\$ git checkout teste

Git - Branch

trazendo as alterações do branch de teste para o branch master (unifincando no master)

vai para o master

\$ git checkout master

traz as alteracoes de teste para master

\$ git merge teste

Git - Branch

caso não precise mais trabalhar no branch teste, podemos apaga-lo

```
$ git branch -d teste
```

Git – Merge

Um exemplo...

em caso de conflito ao fazer merge

\$ git checkout -b teste2 # cria novo merge e faz checkout nele

... alterar um arquivo em determinadas linhas ...

\$ git commit -a -m "bug em x e y corrigido, agora com parametro \$ID"

... enquanto isso, outro colega no branch master ...

(git checkout master)

... altera o mesmo arquivo quase nas mesmas linhas ...

\$ git commit -a -m "bug em x e z corrigido, agora com verif. de \$valor"

... quando voce tenta fazer merge (unificar) com master ...

(git checkout master)

\$ git merge teste2 # ele faz o merge mas vai informar que há conflitos

... no código fonte onde há conflitos, o git insere ambas as versões das linhas com os devidos comentários ...

... e voce edita diretamente no código fonte para deixar apenas a versão que vai ficar ...

commit dos arquivos já sem os conflitos (isto já no master, após o merge)

\$ git commit -a -m "bug em x, y e z corrigido, agora com parametro \$ID e verif. de \$valor"

Git - Repositórios

da primeira estação de trabalho, clonar o repositório

\$ git clone file:///vmwindows/pasta/nomeprojeto nomelocaldoprojeto # exemplo usando vm windows

... ou ...

\$ git clone /caminho/ate/a/pasta nomelocaldoprojeto # formato generico

\$ git push origin master # envia arquivos para o repositorio remoto padrão (origin) (pegando do branch master)

noutra estação de trabalho

git clone /caminho/ate/a/pasta nomelocaldoprojeto

git pull origin master # na primeira vez

noutras vezes

git fetch origin (nome branch) #

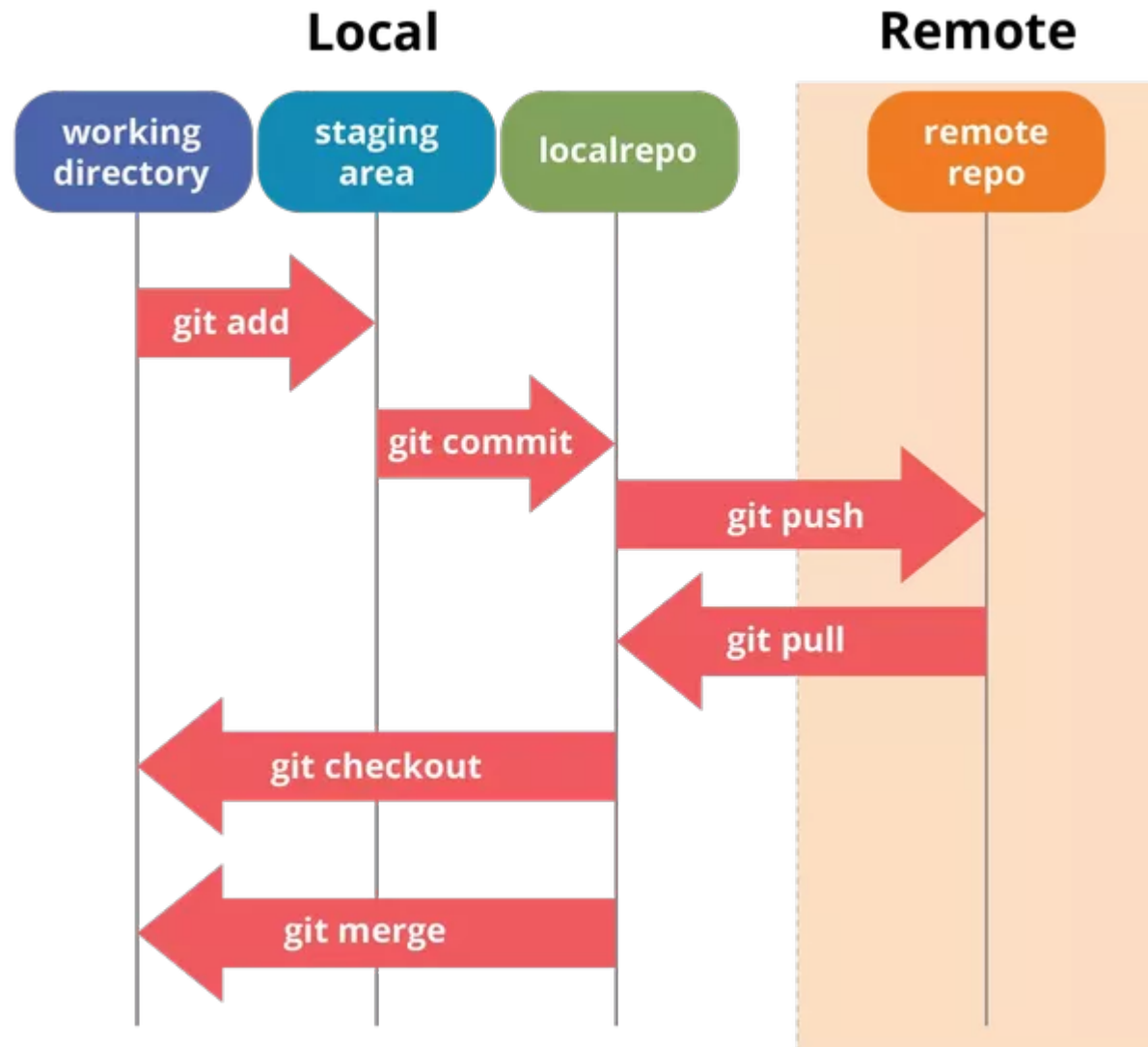
linkando maquina nova

git init

git remote add origin /media/custodio/backup_siteTau/siteTau.git/

git pull origin master

Git - Repositórios



Git - Repositórios

clonar o repositório

```
$ git clone http://github.com/projeto  
nome_projeto
```

... ou ...

```
$ git clone /caminho/para/o/projeto nome_projeto
```

Git - Repositórios

\$ git init

adicionando repositório padrão

\$ git remote add origin /caminho/para/o/projeto

atualizando com o repositório remoto

\$ git pull origin master

enviando alterações para o repositório remoto

\$ git push origin master

Git - Repositórios

atualiz apenas os HEADs, não ors arquivo

\$ git fetch origin (nome branch)

(vou ficar devendo uma explicação decente sobre a utilidade este comando...)

Git - Repositórios

criando o repositório central numa pasta de rede ou num pendrive

\$ cd nome_da_pasta

\$ git init --bare

Git para Desenvolvedores

Sandro Custódio

membro do Tchelinux.org desde 2010

<http://slideshare.net/sandrocustodio>

