

TÉCNICAS DE PROGRAMAÇÃO

Alessandro Ap. Antonio

alessandro.antonio@etec.sp.gov.br

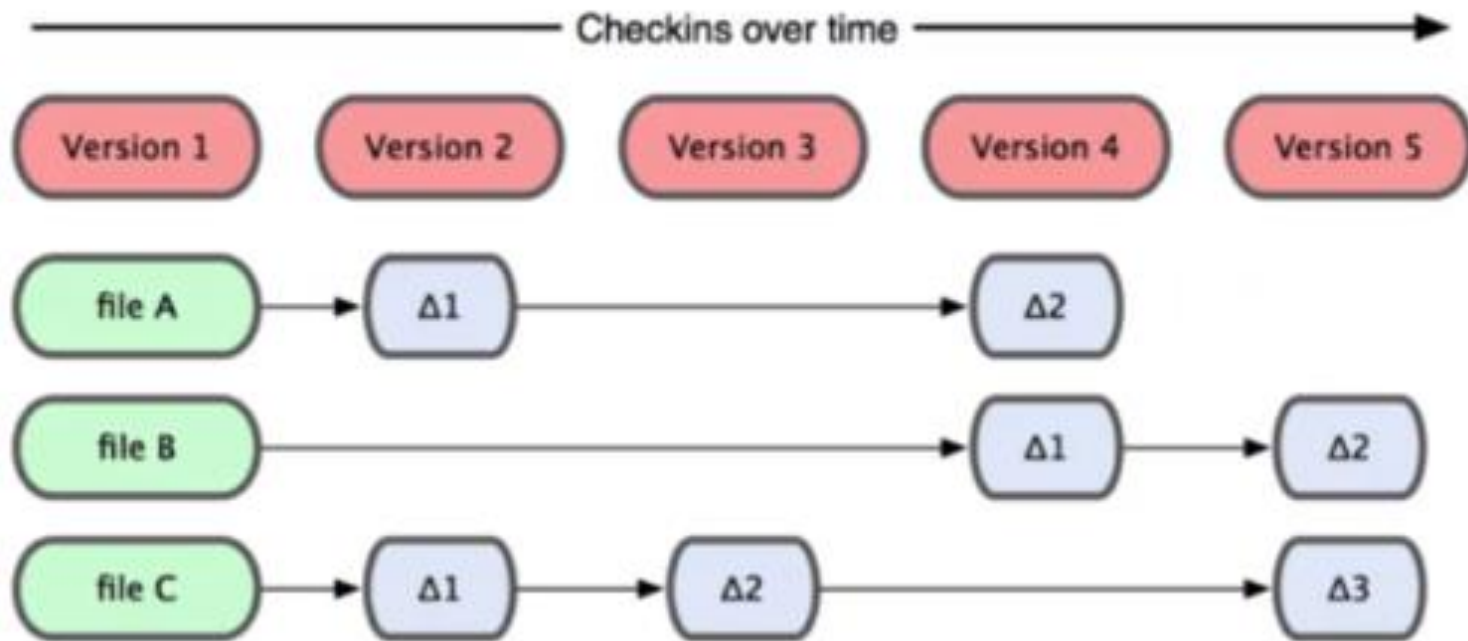
Controle de Versão

Importante:

- Empresas procuram como requisito nos funcionários
- Muitas copias de projetos com poucas alterações
- Espaço em disco
- Apagar arquivo indevidamente sem querer
- Sistema com a finalidade de gerenciar diferentes versões de um documento.
- Responsável por versionar os arquivos de um projeto

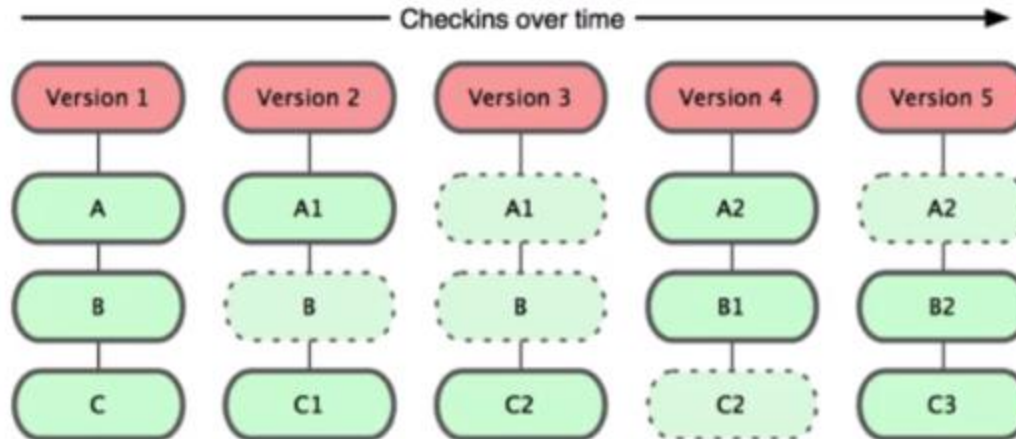
Outros Sistemas

Trabalha com diferença entre os arquivos



Sistema Git

Guarda o status dos arquivos



História

BitKeeper guardava o controle de versão do Linux.

Quebra de Contrato (Linux era isento de pagamentos)

Linus Torvalds criou o seu próprio sistema de controle de versão com as melhorias necessárias para a sua necessidade.

Melhorias:

- Velocidade
- Design simples.
- Suporte robusto a desenvolvimento com branches paralelos.
- Totalmente distribuído.
- Capaz de líder eficientemente com grandes projetos como o kernel do Linux.

GitHub

Serviços de Web compartilhado para projetos que utilizam o Git para versionamento.

GIT e GITHUB são diferentes

GIT → controle de versão

GITHUB → local na web para disponibilizar/armazenar o código fonte

Instalação

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

Configuração

Nome de Usuário

E-mail

Guarda as informações em 3 lugares:

Git_config

Sistema /etc/gitconfig

Usuário ~/.gitconfig ou ~/.config/git/config

Projeto .git/config

```
git config --global user.name "Aluno"
```

```
git config --global user.email "aluno@etec.sp."
```

```
git config --global core.editor <comando>
```

```
git config user.name     (user.email)
```

```
git config --list
```


Iniciando um repositório

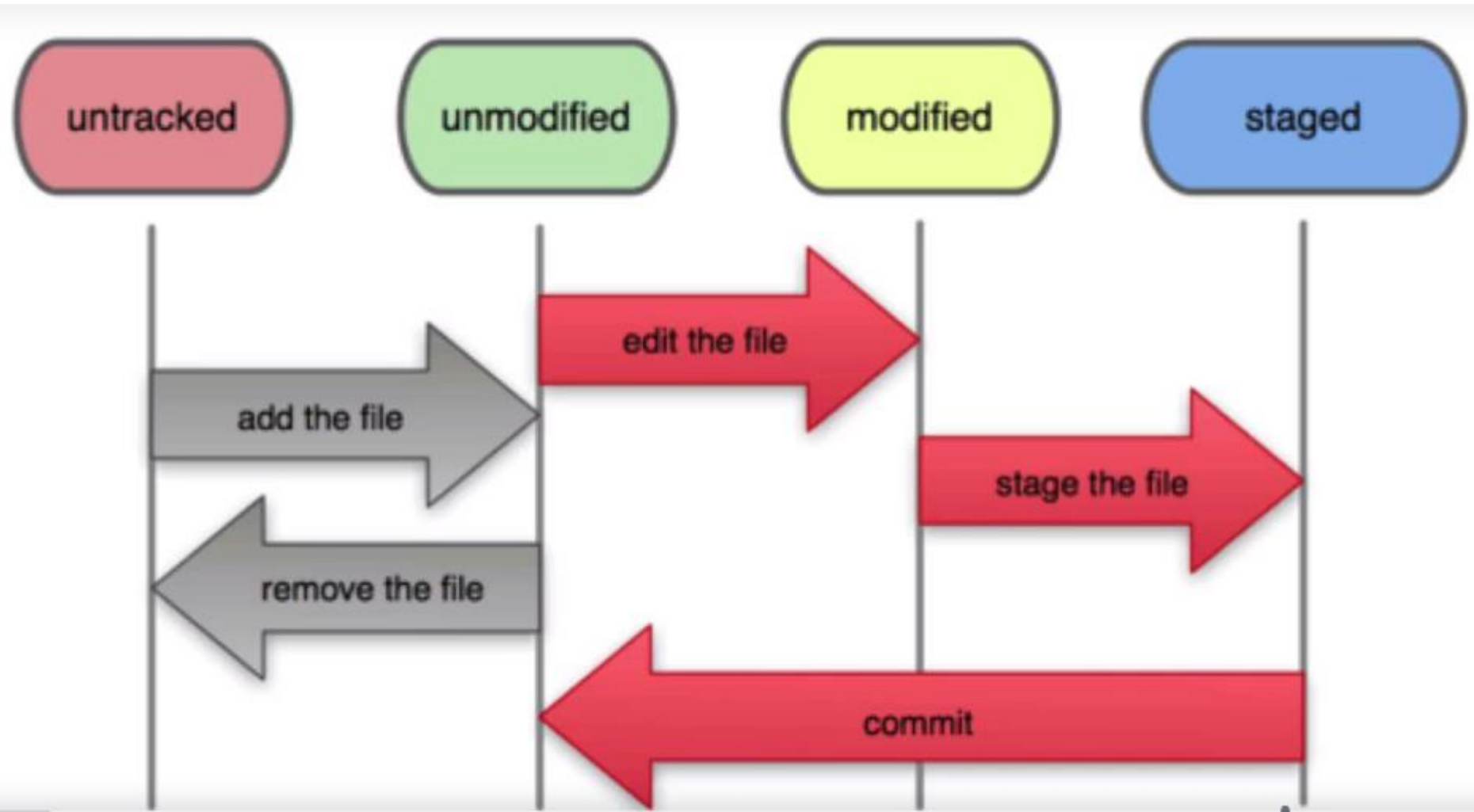
Inicializando um repositório

```
git init
```

Listando o conteúdo do diretório

```
ls -la
```

Ciclo de vida dos status dos arquivos



Status do Repositório

`git status`

```
ProjetoGit (master)  
$ git status  
On branch master  
No commits yet  
  
nothing to commit (create/copy files and use "git add" to track)
```

Crie um arquivo com o nome Readme.md

Github

Status do Repositório

`git status`

```
ProjetoGit (master)
$ git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)

    README.md

nothing added to commit but untracked files present (use "git add" to track)
```

Adicionando o arquivo

`git add README.md`

`git status`

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   README.md
```

Status do Repositório

Alterando o arquivo Readme.md

Github

Primeiro Projeto usando controle de Versão

```
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

        new file:   README.md

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working directory)

        modified:   README.md
```

Status do Repositório

```
git add Readme.md
```

```
git commit -m "Criação do Projeto"
```

```
ProjetoGit (master)
$ git commit -m "Criação do Projeto"
[master 038c65b] Criação do Projeto
1 file changed, 2 insertions(+)
create mode 100644 Readme.md
```

```
git status
```

Arquivo Readme.md

Github

Primeiro Projeto usando controle de Versão

Ajuda: [Git](<https://git-scm.com/doc>)

```
git status
```

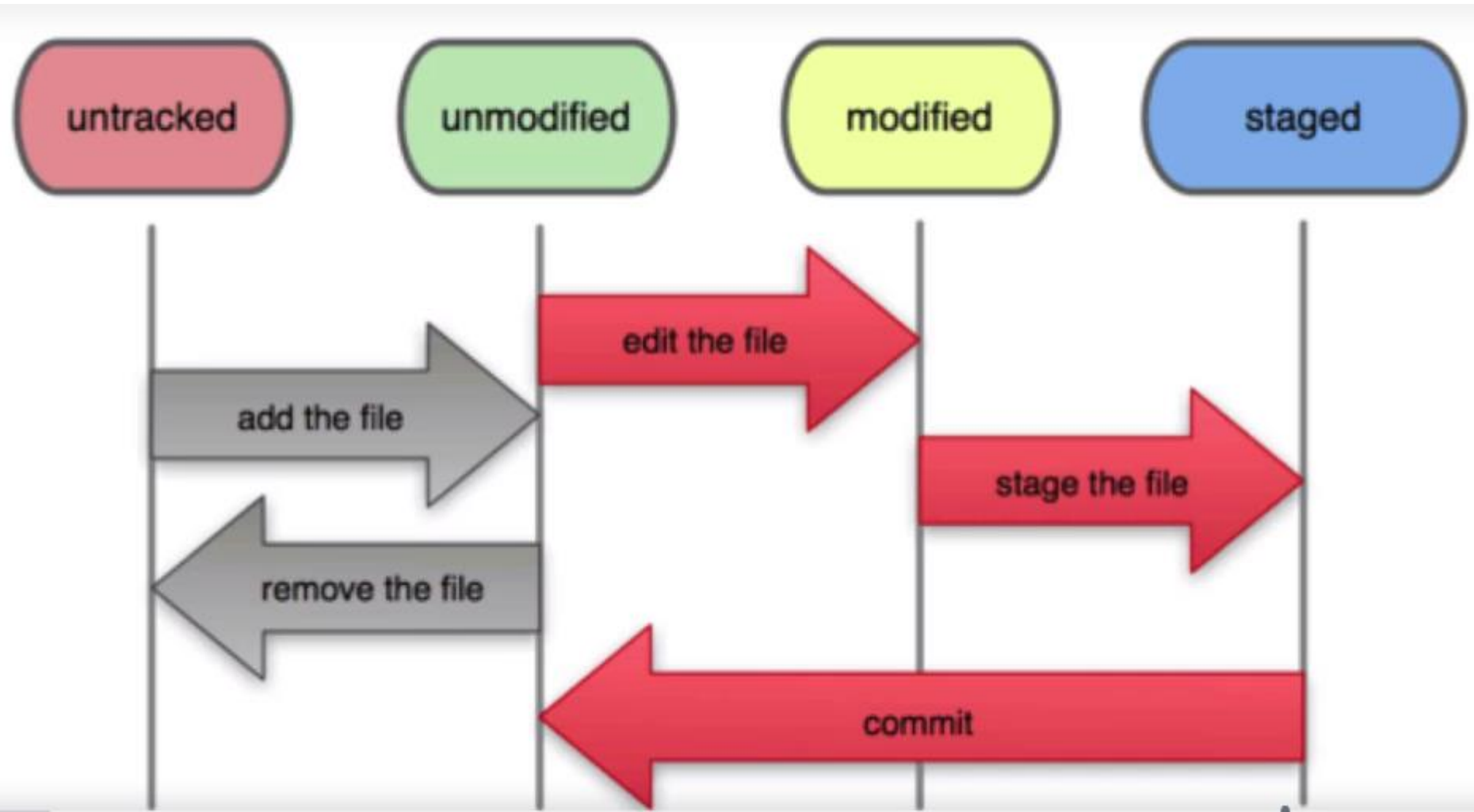
```
git commit -m "Vai dar Erro"
```

```
git add Readme.md
```

```
git status
```

```
git commit -m "Adicionado um link de Ajuda"
```

Ciclo de vida dos status dos arquivos



Configurações Opcionais

Configurando o git no VS Code

Arquivo → Preferencias → Settings

Pesquise por **shell**

terminal > integrated > shell : windows

Altere o caminho para: **C:\Program Files\Git\bin\bash.exe**

Não salvar as credenciais no Windows

Executar o comando como Administrador no repositório git.

```
git config --system --unset credential.helper
```

```
git config --list
```

Ligando repositório local a um remoto

Crie um novo repositório com o nome **Tecnicas**

Trocar para Https

```
git remote add origin https://github.com/AlessandroEtec/Tecnicas.git  
git push -u origin master
```

```
ProjetoGit (master)  
$ git push -u origin master  
Enumerating objects: 15, done.  
Counting objects: 100% (15/15), done.  
Delta compression using up to 8 threads  
Compressing objects: 100% (10/10), done.  
Writing objects: 100% (15/15), 1.31 KiB | 268.00 KiB/s, done.  
Total 15 (delta 3), reused 0 (delta 0)  
remote: Resolving deltas: 100% (3/3), done.  
To https://github.com/AlessandroEtec/Tecnicas.git  
* [new branch]      master -> master  
Branch 'master' set up to track remote branch 'master' from 'origin'.
```

Clonando um repositório

Clonar um repositório é uma pratica utilizada quando:

- trabalha em equipe
- utiliza varias computadores para o desenvolvimento do mesmo projeto.
- Copiar um projeto para estudar o seu código

<https://github.com/AlessandroEtec/Tecnicas>

```
git clone
```

<https://github.com/AlessandroEtec/Tecnicas.git>

```
ProjetoGit_Clone
```

Logs

Exibindo todos os logs.

```
git log
```

Exibindo os logs de um autor.

```
git log --author="NomeDoAutor"
```

exibindo o log de forma resumida

```
git shortlog
```

Quantidade de logs

```
git shortlog -sn
```

```
git show 57eaa5355ed10bfb69367c8cc10fe10402ce7
```

Diff

O Comando diff exibe as alterações feitas no arquivo antes de commit

Arquivo Readme.md

```
# Github
```

```
Primeiro Projeto usando controle de Versão
```

```
! [Logo] (https://github.com/favicon.ico)
```

```
Ajuda: [Git] (https://git-scm.com/doc)
```

```
git diff
```

```
git commit -am "Teste diff"
```