**Santander Coders 2023**

**Aluno: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Data: \_\_/\_\_/\_\_**

**Principais comandos GIT**



**1.1 Criar novo repositório local:**

**> git init**

**Ou seja, o arquivo/pasta agora poderá ser salvo através das possibilidades da linguagem git, trazendo versionamentos, árvores de desenvolvimento para seu projeto.**

**1.2 Clonar um repositório existente no servidor:**

**I. Pegar link no site da** [**www.git.hub.com**](http://www.git.hub.com) **e usar no comando abaixo**

**> git clone link**

**EXEMPLO ABAIXO:**

**> git clone** [**www.git.hub.com/webnamorada.html**](http://www.git.hub.com)

**II. Para acessar um diretório existente no PC, copiamos o caminho do diretório e utilizamos o comando ‘’cd‘’ para entrar no diretório através do terminal de comando.**

**EXEMPLO ABAIXO:**

**> cd C:\Windows**

**1.3 Verificar estado e alterações feitas em seus arquivos:**

**> git status**

**\*Obs.: Para obter um detalhamento das modificações feitas, usamos o comando:**

**> git diff**

**Os arquivos removidos estarão em vermelho.**

**Os arquivos adicionados estarão em verde.**

**1.4 Adicionar arquivos para área (Staged); ou seja, passar do estado de modificado (em edição) para conclusão (todas modificações feitas) e assim, por final, salvar a versão atual no servidor global / local através da função denominada ‘’ commit ‘’.**

**> git add**

**\*Obs.:**

**I. Para adicionar um arquivo específico, usamos:**

**> git add nomedoarquivo.tipodearquivo**

**EXEMPLO ABAIXO:**

**> git add site1.txt**

**II. Para adicionar um diretório específico (pasta), usamos:**

**> git add diretorio**

**EXEMPLO ABAIXO:**

**> git add Programação/VSCode**

**III. Para descartar as mudanças feitas e não seguir para a etapa de ‘’commit’’, utilizamos o comando:**

**> git restore arquivo/diretorio**

**1.5 Enviar versão atual dos arquivos para servidor global / local:**

**> git commit arquivo ou diretório –m ‘’ descrição das alterações de forma simplificada ‘’**

**EXEMPLO ABAIXO:**

**> git commit –m ‘’ modificação dos contatos do site ‘’**

**1.6 Visualizar o histórico de alterações/commits:**

**> git log**

**\*Obs.:**

**I. Para obter um histórico detalhado, utilizamos:**

**> git log –stat**

**1.7 Para criar uma nova ramificação no seu projeto, podendo desenvolver ideias diferentes, usamos:**

**> git branch nomedaramificacao**

**\*Obs.:**

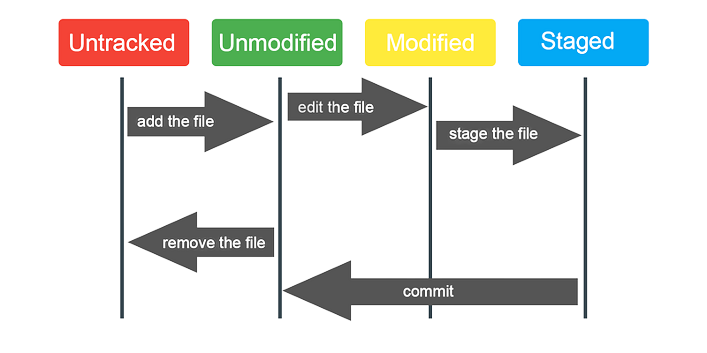
**I. Para alternar entre as ramificações, lembrando que (por padrão), o git sempre apontará para master, usamos:**

**> git checkout nomedabranch**

**II. Ao criar ramificações, nós separamos a sequência de versionamento, de certa forma, porém é possível uni-las novamente; em primeiro momento, temos que acessar a branch que queremos trazer as informações, ou seja, se estou na branch secundária e quero trazer algo para principal, uso o >git checkout branch1 e, posteriormente, utilizo o comando:**

**>git merge branch2**

**1.8 Abaixo temos a sequência do funcionamento do Git:**



**Fonte:**

[**https://gist.github.com/leocomelli/2545add34e4fec21ec16**](https://gist.github.com/leocomelli/2545add34e4fec21ec16)