Atalho git terminal

* **git add .** : Para adicionar todas as modificações realizadas de uma só vez;
* **git commit -m “mensagem do commit”** : usado quando queremos capturar e salvar o estado atual do repositório;
* **git status**: serve para listar todos os arquivos que foram modificados;
* **git push**: utilizado para envio das alterações gravadas no diretório local para o repositório remoto. É utilizado para atualizar o repositório remoto com os commits feitos no repositório local;
* **git restore**: trabalha com a restauração de arquivos e do projeto, voltando para um estado anterior através;
* **git log**: exibe o histórico de commits do repositório. Mostra informações como o autor do commit, data, hora e mensagem associada a cada commit;
* **git log --oneline** : mostra o log de forma resumida, em apenas uma linha;
* **git clone**: clona um repositório para o seu ambiente local. É utilizado para obter uma cópia completa do repositório, incluindo todos os arquivos e histórico de commits. É utilizado quando você deseja obter uma cópia completa de um repositório remoto para começar um novo projeto ou colaborar com um projeto existente;
* **git branch**: cria, lista ou exclui branches. Um branch é uma ramificação do histórico de commits, permitindo trabalhar em paralelo em diferentes versões do projeto;
* **git checkout**: altera o branch atual ou restaura arquivos. É utilizado para alternar entre branches existentes ou restaurar arquivos para uma versão anterior. Atenção: é um comando que tem mais de uma função, se você quiser apenas trocar de branch sem alterar modificações, você pode utilizar o git switch;
* **git checkout -b “informe\_o\_nome\_da\_branch”**: Cria uma nova branch e muda para ela;
* **git switch “nome\_da\_branch”**: troca de uma branch para outra sem realizar alterações de estado. Observação: comando ainda em fase de experimentação;
* **git merge**: mescla alterações de um branch para outro. É utilizado para combinar as alterações feitas em um branch com outro, integrando as modificações;
* **git pull**: atualiza o repositório local com as alterações do repositório remoto. Quando você já possui uma cópia do repositório em seu computador e deseja trazer as últimas alterações feitas por outros colaboradores para sincronizá-las com seu repositório local. É diferente do git clone, que faz um clone do repositório remoto na sua máquina.
* **git fetch**: busca as alterações do repositório remoto, mas não as mescla com o branch atual. É para obter informações mais recentes do repositório e verificar se há mudanças. Diferente do git pull , que já copia as alterações do repositório remoto;
* **git diff**: mostra as diferenças entre arquivos ou commits;
* **git reset**: desfaz commits ou altera o estado do repositório;
* **git remote**: gerencia repositórios remotos;