COMANDOS

OBRIGATÓRIOS

1 - git config

Quando você está utilizando o Git pela primeira vez ou com uma instalação nova, esse comando é fundamental para configurar sua identidade de usuário, inserindo informações como nome e email que serão empregadas em cada commit.

Exemplo:

$ git config –global user.name “Seu nome”
$ git config –global user.email “Seu email”

2 - git init

Esse é o comando que você irá utilizar para criar um novo projeto de git. O comando irá criar um repositório novo em branco e, a partir daí, será possível armazenar seu código fonte, alterar, etc.

Exemplo:

$ git init

Se você já possui um repositório anterior ou deseja criar um repositório com um nome em específico, você pode passar o nome como parâmetro do comando:

$ git init <O nome do seu repositório>

3 - git clone

Esse comando Git cria uma cópia exata de um repositório já existente.

O git clone já executa um comando git init internamente e verifica todo o conteúdo do projeto, então depois de clonado um projeto já é possível fazer pull requests.

Exemplo:

$ git clone <URL do seu projeto>

4 - git status

Em resumo, o git status é uma ferramenta essencial para revelar o que foi modificado, adicionado ou excluído em seus arquivos desde o último commit. Ele fornece informações valiosas para que você tome ações assertivas e mantenha seu projeto organizado.

Exemplo:

$ git status

5 - git add

Esse comando Git adiciona os arquivos especificados de código ao seu repositório, sejam arquivos novos ou arquivos anteriores que foram alterados. Oferece diferentes possibilidades de sintaxe.

Exemplo:
$ git add . (esse comando adicionar o(s) arquivo(s) dentro do mesmo diretório que está sendo chamado)

$ git add * (esse comando irá adicionar todos os arquivos novos e/ou modificados ao repositório)

$ git add <arquivo modificado> (esse comando irá adicionar o(s) arquivo(s) especificados)

6 - git commit

Um commit adiciona as alterações mais recentes do código-fonte para o repositório, tornando essas alterações parte da revisão principal do repositório.
Cada commit possui um identificador único, chamado de hash SHA, que possibilita referenciá-lo posteriormente. Além disso, os commits ficam registrados permanentemente no histórico do projeto, o que possibilita revisitá-los ou revertê-los posteriormente.
A mensagem para a execução de um commit é muito importante para identificar posteriormente o que aquele commit contém.

Exemplo:
$ git commit -m “seu comentário”

7 - git branch

Uma branch (ou ramo) no desenvolvimento de software é essencialmente uma linha de desenvolvimento separada que permite aos desenvolvedores trabalharem em recursos específicos ou correções de bugs sem interferir diretamente no código principal do projeto.

Exemplos:

$ git branch (lista todas as ramificações)

$ git branch <nome_do_branch> (cria um branch com o nome especificado)

$ git branch -d <nome_do_branch> (deleta o branch com o nome especificado)

8 - git checkout

Esse comando Git pode ser utilizado para trocar de uma branch (ramificação) para outra.

Exemplo:

$ git checkout <nome_do_branch>

Também é possível criar uma branch a partir de uma outra e já ir para esta nova branch com um único comando, basta você estar na branch que quer usar como base para a sua nova branch:

$ git checkout -b <nome_do_branch_novo>

9 - git push

Esse comando serve para subir suas modificações para um repositório remoto conectado anteriormente com git remote.

Exemplo:
$ git push

É importante especificar a origem e o upstream antes de usar o git push. Caso tenha criado a branch localmente com o comando citado no item 8, no primeiro comando de push é necessário setar para o repositório remoto que esta branch existe

$ git push --set-upstream <repository> <nome_do_branch>

10 - git pull

O comando git pull é utilizado para buscar e baixar conteúdo de repositórios remotos e atualizar imediatamente o repositório local para que ambos fiquem em sincronia

Exemplo:

$ git pull

11 - git merge

O comando git merge fará a junção das alterações feitas à base de código em uma branch separada à sua branch atual como um novo commit.

Exemplo:

$ git merge <nome_da_branch>

Por exemplo, se você está trabalhando atualmente em uma branch chamada dev e gostaria de fazer o merge de alterações feitas em uma branch chamada new-features, o comando teria esta sintaxe:

$ git merge new-features *(você deve de estar na branch dev para rodar este comando, uma vez que é lá que você quer juntar as modificações feitas em new-featrures)*

12 - git stash

sse comando Git armazena temporariamente seus arquivos modificados em uma área chamada stash (“esconderijo”), sem interagir com os outros arquivos até ser necessário.

Exemplo:
$ git stash

Para listar todos os seus “esconderijos”, usamos:
$ git stash list

Quando for o momento de aplicar o conteúdo do stash a um branch, usamos o parâmetro “apply”:

$ git stash apply

vamos em frente

FERNANDO GABRIEL (DEV SAMURAI)