**Git**

- É um gerenciados de versões de códigos, onde é possível controlar o fluxo de versões do mesmo código.

**GitHub**

- É para onde as versões dos códigos vão, é um site onde se pode gerenciar de forma gráfica as versões do código

**Após instalar**

- Abrir o CMD e digitar **git –help**

**Preparando os arquivos**

1- Selecionar um diretório e com o botão direito apertar em “Git bash here”

2- Agora, digitar o seguinte código - “git init”

3- No diretório, ativar a visualização de itens ocultos

**Preparando o ambiente pra iniciar o versionamente**

4- Verificar o status do diretório usando o comando “git status”

5- Para adicionar um arquivo para o controle de versão basta usar o código “git add ‘nome do arquivo’”

6- Para adicionar mais de um arquivo ao mesmo tempo podemos usar o comando “git add .”

7- Se for a primeira vez usando o commit, deveremos configurar o acesso, em seguida rodar o código “git config --global user.email [you@example.com](mailto:you@example.com)”” e depois “git config –global user.name “nome do usuario””

**Criando a primeira versão do código**

8- Criando a primeira versão do código com o comando “git commit -m “commit incial””

9- Para enviar os arquivos para o github, primeiro precisamos criar um repositório, depois copiar o link desse repositório, por fim usar o comando “git remote add origin link.do.diretorio”

10- Agora devemos criar uma branch git push --set-upstream origin master “token de acesso ghp\_JwcrLwNQ6bpvp2NcjaBLEGhzLsOc6619v06A”

11- Para enviar, basta usar o comando git push

12- Adicionando uma nova versão como comando: git commit m- “Nova versão – nova linha”

depois rodar o comando git push

**Verificar o histórico de atualizações**

13- Como verificar o histórico de atualizações? Usando o comando git reflog

14- Como voltar uma versão anterior do código: Pegar o id da versão com o git reflog, depois usar o comando git reset --hard id\_da\_versão

**Trabalhando com Branchs (servem para criar uma outra linha de fluxo de código)**

15- Verificando quais branch estão diponiveis: git branch

16- Criando uma nova branch: git branch nome da branch

17- Para mudar de branch: git chekout nome da branch

**Mudando de uma branch para a outra (unir códigos de uma branch com outra)**

18- Antes de unir, precisamos verificar se estamos usando as versões mais recentes de uma branch. Para isso devemos usar o comando: git pull

19- Para unir, usar o código: git marge nome da branch (antes de fazer isso devemos voltar para a branch princial)

20- Criando uma nova branch com base em outra: git checkout -b nome-da-nova-branch nome da branch de base

**Pull request – um merge demográfico**

21- Serve para efetuar as aprovações de alteração de códigos onde quando se trabalha em uma equipe, pode se usar para aprovar ou não merges entre branch segundarias e branch principal

22- Após alterar o código basta efetuar os commit, entretanto no lugar de dar um merge, usamos um push diretamente, logo após o commit, usando o comando: git push --set-upstream origin sistema-de-login

23- Devemos ir agora no site do github e apertar em copare & pull request

24- Definir como será o merge

25- Adicionar os comentário e titulo

26- Create pull request

27- Aguardar marge (caso trabalhe só, basta efetuar o marge pull request)