**Comandos para subir um repositório**

* git init (inicia o git no repositório de trabalho)
* git status (verifica no repositório mudanças que ainda não foram colocadas no git)
* git add “nome\_do\_arquivo” (adiciona o arquivo para o novo controle de versão)
* git add . (adiciona todos os arquivos do repositório para o novo controle de versão)
* git commit -m “comentário\_sobre\_a\_versão” (sobe as mudanças para a nuvem git)

Caso ainda não esteja configurado o usuário o comando de commit vai apresentar meio que uma mensagem de erro, basta rodar esses comandos para vincular com a conta git na nuvem, no caso, o github.

* git config --global user.email "email\_do\_github"
* git config --global user.name "seu\_nome"

Após configurado, basta rodar o comando de commit de novo caso tenha acontecido algum “erro”. Contudo caso não tenha aparecido nenhuma mensagem de erro, pode partir para próxima parte que é criar um repositório la no github.

Texto

Descrição gerada automaticamente

Com o repositório criado, é necessário copiar o link desse repositório e colar no seguinte comando:

* git remote add origin “link copiado”
* git push --set-upstream origin branch\_selecionada (sobe o código na nuvem para o branch que está selecionada, por padrão é a master)

Conforme mudanças no código forem feitas, basta rodar os mesmos comandos a partir do git add., lembrando de fazer um comentario que faz referência as mudanças no git commit – m.

**Comandos para voltar uma versão anterior**

* git reflog (mostra todas as versões de commits, a primeira é a mais atual)
* git reset --hard id\_da\_versão (volta para versão do id utilizado no comando e que foi obtido através do git reflog)

**Utilizando branchs**

* git branch (lista as branchs disponíveis no projeto)
* git branch nome\_da\_ branch (cria uma branch nova)
* git checkout nome\_da\_branch (muda a branch que está sendo trabalhada para uma nova)
* git checkout -b nome\_branch\_vai\_criar nome\_branch\_com\_conteúdo (cria uma nova branch a partir de um contéudo de outra branch

As mudanças feitas a partir de agora não irão alterar no branch master e isso vale para mais branchs envolvidas, podendo ser feitos commits e pushs de forma mais isolada.

Para trazer as mudanças realizadas em uma branch para outra e necessário fazer um merge, nesse caso é utilizada a seguinte sequência de comando:

* git checkout nome\_branch\_principal (geralmente é a master)
* git pull (garante que a última versão da branch principal esteja no seu computador)
* git merge nome\_branch\_que\_onde\_estão\_as\_mudanças
* git push

**Utilizando o gitignore**

Caso tenha algum arquivo no repositório que não se deseja fazer um push/commit é necessário fazer o uso da ferramenta gitignore

* touch .gitinore (criar um arquivo com o nome de gitignore)

Após criar esse arquivo, basta colocar o nome dos aquivos que não se deseja no commit seguido de uma barra, conforme o exemple:

* fotos.png/

https://www.youtube.com/watch?v=kB5e-gTAl\_s