# Apresentação Github 17/05

Funcionamento

Ela basicamente funciona para produzir “Backups” que podem ser editados por várias pessoas simultaneamente, permite fazer operações de inclusão, validação e exclusão de arquivos na internet, como uma ‘nuvem’, permitindo que as pessoas com acesso possam verificar o dia e o que cada pessoa alterou/adicionou

Quando usar

Recomendado para ser usado em projetos em equipe, por ser uma ferramenta exclusiva de código, deve ser usada em projetos de desenvolvimento

Por que é útil?

Porque ela oferece a possibilidade de criar “Branches”, uma ramificação do projeto original, isso é muito interessante quando se tem a necessidade de fazer alguns testes no projeto sem que eles possam ser danosos ao arquivo original (aquele que está em funcionamento), fornecendo a possibilidade de fazer “upload” apenas da versão que trouxeram resultados positivos dos testes, sem que esse processo afete o original

Comandos

Configuração inicial:

Git config(usuário, e-mail,editor padrão):

## git config --global user.name “Seu Nome”

## git config --global user.email “ [seu.email@exemplo.com](mailto:seu.email@exemplo.com)”

## git config --global core.editor”nano”

Para sobrescrever usar o comando de:

## git config --global --replace-all user.name "Seu Nome"

Verificar se houve ou não alteração

## git status

Adicionar arquivos

## git add <arquivo> \\(Adiciona arquivos específicos)

## git add . //(Adiciona TODOS os arquivos)

Criar o commit

## git commit -m

Exibir o histórico de commits

## git log --graph --oneline --decorate --all (Verifica o histórico de commits)

## git log --author="Nome do Autor" (Verifica o histórico por autor)

Alternar entre as branches

## git checkout <nome da branch> (entra na branch específica)

## git checkout -b <nome da Branch

Iniciando a criação do arquivo

## git init

## git remote add <nome do arquivo> <url>

Abrindo uma branch

## git push -u origin main

## git branch –master

Comando para dar o commit

## git add .

## git commit -m “descrição do commit”

Criando o README

## echo "# workshop" >> README.md

Conectando com o diretório

## git remote add origin

Subir o diretório

## git push -u origin main

Para baixar alterações que estão feitas na nuvem para a máquina

## git pull origin main

Processo para adicionar o arquivo para a nuvem

## git checkout -b <nome para criar branch>

## git checkout <nome da branch>

## git add . ou git add <nome do arquivo >

## git commit -m “descrição”

## git push

Convenções

Antes de subir uma alteração deve ter uma Branch com a versão completa a ser usada na main, para que possam ser usada para testar possíveis problemas