## **O que é o Git?**

O Git é um sistema de controle de versão de arquivos. É um software livre e muito utilizado no desenvolvimento de software onde diversas pessoas estão contribuindo simultaneamente, podendo criar e editar arquivos. Sempre quando alguém disponibiliza sua parte do projeto no Git, ele gerencia as alterações feitas e guarda um histórico. Isso é importante pois se houver algum problema você pode desfazer as alterações e voltar para a versão que estava estável.

## **O que é o GitHub?**

O GitHub é uma plataforma onde você pode armazenar seus projetos. É como se fosse uma rede social, só que de códigos, onde seus desenvolvedores podem disponibilizá-los para outras pessoas verem.

**lembrem-se:**

a diferença entre git e github é que **o git é só uma ferramenta para versionar projetos**, enquanto **o github é o site no qual você colocará esses projetos versionados**. uma analogia válida seria que o git é seu pincel e tintas enquanto o github é um museu.

**Descreva a função de cada um dos comandos abaixo**

### **Criar novo repositório**

git init

##### **Adicionar um arquivo em específico**

git add meu\_arquivo.txt

##### **Adicionar um diretório em específico**

git add meu\_diretorio

##### **Adicionar todos os arquivos/diretórios**

git add .

##### **Adicionar um arquivo que está listado no .gitignore (geral ou do repositório)**

git add -f arquivo\_no\_gitignore.txt

### **Verificar estado dos arquivos/diretórios**

git status

### **Comitar arquivo/diretório**

##### **Comitar um arquivo**

git commit meu\_arquivo.txt

##### **Comitar vários arquivos**

git commit meu\_arquivo.txt meu\_outro\_arquivo.txt

##### **Comitar informando mensagem**

git commit meuarquivo.txt -m "minha mensagem de commit"

### **Visualizar histórico**

##### **Exibir histórico**

git log

##### **Exibir histórico com diff das duas últimas alterações**

git log -p -2

##### **Exibir resumo do histórico (hash completa, autor, data, comentário e qtde de alterações (+/-))**

git log --stat

##### **Exibir informações resumidas em uma linha (hash completa e comentário)**

git log --pretty=oneline

##### **Exibir histórico com formatação específica (hash abreviada, autor, data e comentário)**

git log --pretty=format:"%h - %an, %ar : %s"