# Git

O Git permite acompanhar alterações feitas em um projeto. O Git registra as alterações feitas em um projeto.

## Git Workflow

Podemos separar o Git Workflow em três partes:

1. **Diretório de Trabalho:** Onde é feito toda a organização do trabalho, como criar, editar ou excluir arquivos.
2. **Área de Preparação:** Onde serão listadas todas as alterações feitas no **Diretório de Trabalho**.
3. **Repositório:** Onde o Git armazena permanentemente todas as alterações e versões do projeto.

## Inicializando o Git

O comando **‘git init’** inicializa o Git, habilitando todas as ferramentas necessárias.

## ‘git status’

Mostra os status. Arquivos verdes são arquivos novos. Os arquivos vermelhos são arquivos não “comitados”.

## ‘git add’

Adiciona o arquivo à área de preparação. A sintaxe é a seguinte:

**‘git add <nome\_arquivo>’**

## ‘git diff’

Compara um arquivo em sua versão na área de trabalho com o mesmo arquivo na área de preparação, mostrando suas diferenças.

A sintaxe é a seguinte:

**‘git diff <nome\_arquivo>’**

## ‘git commit’

Armazena permanentemente as alterações da área de preparação para o repositório.

A sintaxe é a seguinte:

**‘git commit -m “Mensagem sobre as alterações”’**

A mensagem deve respeitar as seguintes regras:

* Deve estar entre aspas.
* Resume o que foi feito naquela versão.
* Deve possuir 50 caracteres ou menos.
* Deve vir depois do **‘-m’**.

## ‘git log’

Permite consultar as versões anteriores do projeto. Todos os commits são armazenados em ordem cronológica no repositório, e o comando **‘git log’** permite visualizá-los.

Sua sintaxe é:

**‘git log’**

Esse comando retornará algumas informações:

* Um texto em laranja que possui 40 caracteres chamado *SHA*. Esse texto especifica o commit.
* O autor do commit.
* A data e o horário do commit.
* A mensagem do commit.