

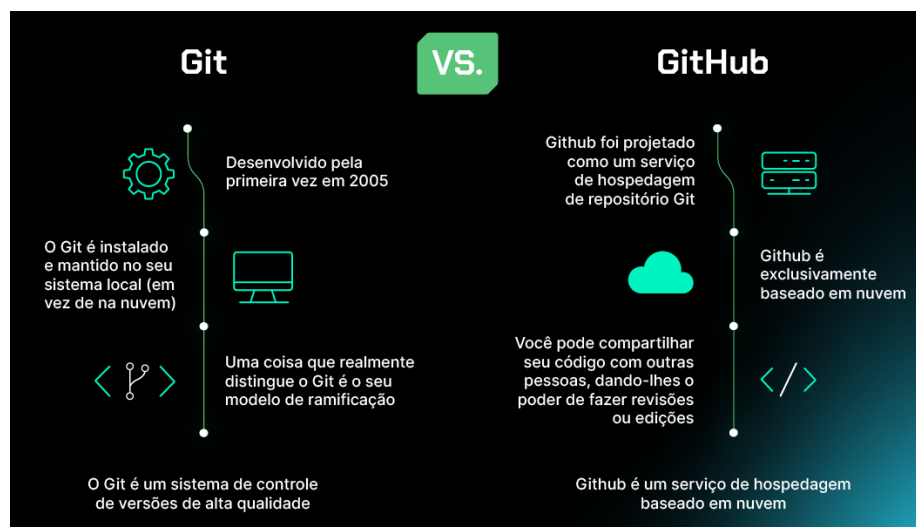
Git / Github



O **Git** é um sistema de controle de versão de arquivos; em outras palavras, é responsável por guardar o histórico de alterações sempre que alguém modificar algum arquivo que está sendo monitorado por ele.

O **GitHub** é uma “rede social dev” (repositório) em que é possível armazenar e compartilhar projetos de desenvolvimento de software.

Comparativo

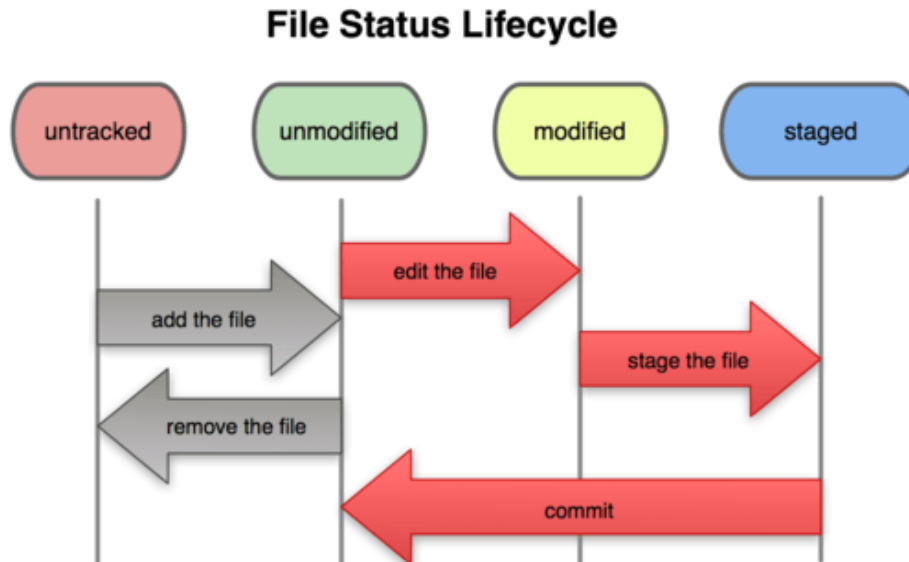


Como baixar e instalar o Git?

Windows: Acesse o site oficial do Git em “<https://git-scm.com/download/win>”

Linux: Use o comando `sudo apt-get install git`

Entendendo os estados



Usando o git

1. Configure seu nome de usuário e e-mail:

O Git registra quem fez cada alteração no código. Portanto, é importante configurar seu nome de usuário e e-mail. Use os comandos, no terminal:

2. Crie um Repositório Git:

Para começar a rastrear seu código, crie um repositório Git em seu projeto. Navegue até a pasta do seu projeto e execute:

```
git init
```

3. Adicione Arquivos ao Controle de Versão:

Use o comando git add para adicionar arquivos ao “staging area”, que é onde você prepara os arquivos para serem “commitados” ou salvos.

```
git add nome-do-arquivo
```

4. Faça um Commit:

Um commit é um snapshot de suas alterações. Use o comando git commit para criar um commit com uma mensagem descritiva do que foi alterado no projeto.

```
git commit -m "Sua mensagem de commit aqui"
```

5. Visualize o Histórico de Commits:

Use git log para ver o histórico de commits no repositório.

```
git log
```

Comandos básicos

- Git init É utilizado para inicializar um repositório Git dentro de um diretório do sistema. Após sua utilização, a ferramenta passa a monitorar o estado dos arquivos no projeto.
- Git clone É utilizado para criar uma cópia de um repositório remoto em um diretório da máquina. Este repositório poder ser criado a partir de um repositório armazenado localmente, através do caminho absoluto ou relativo, ou pode ser remoto, através do URI na rede.

A partir de um repositório clonado, é possível acompanhar o estado de um projeto e suas modificações, além de contribuir com o projeto, a partir do envio das suas modificações ao repositório central.

- Git status É utilizado para verificar o status de um repositório git, bem como o estado do repositório central. O comando mostra informações sobre se o projeto local está sincronizado com o central, quais arquivos estão sendo monitorados pelo Git e em qual branch você está no projeto.
- Git add É utilizado para adicionar arquivos ao pacote de alterações a serem feitas. É possível adicionar um único arquivo, múltiplos arquivos de uma vez, como

```
git add <-arquivo1-> <-arquivo2->
```

ou até mesmo um diretório, a partir de seu caminho. Uma vez que um arquivo é adicionado ao pacote de alterações com o comando add, ele está pronto para entrar no próximo commit.

- Git commit

```
git commit -m "mensagem do commit"
```

É utilizado para criar uma nova versão do projeto a partir de um pacote de alterações. O commit pega o pacote de modificações adicionado através do comando git add, fecha essas alterações num pacote e o identifica através de um Hashcode.

Além disso, para cada commit é necessário escrever uma mensagem para identificá-lo, com uma mensagem clara de quais alterações foram feitas neste commit.

- Git log

```
git log
```

É utilizado para ver o histórico de alterações do projeto, onde aparecerão todos os commits feitos, com suas respectivas mensagens e códigos identificadores.

O comando é muito útil quando precisamos rastrear o andamento de um projeto e verificar em qual ponto cada funcionalidade foi implementada.

Além disso, o comando conta com várias opções para mostrar o histórico de forma resumida, gráfica e até mesmo mostrando a diferença entre os commits, que podem ser vistas na documentação oficial do comando.

Como linkar os repositórios remoto e local?

Use o comando `git remote add origin` para adicionar o repositório remoto como um “remote” chamado “origin”.

Exemplo:

```
git remote add origin https://github.com/seu-usuario/seu-repositorio.git
```

Enviando para o Repositório Remoto

Use o comando `git push -u origin main` para enviar os commits para o repositório remoto.

```
git push -u origin main
```