

# Tutorial Git.

---

## Conceitos Fundamentais:

Tipo	Onde fica	Para que serve	Comandos mais usados
Local	No seu computador	Desenvolver e versionar o código	add, commit, status,
Remoto	Na nuvem (GitHub, GitLab...)	Compartilhar e sincronizar o código	push, pull, clone
Comando	Descrição		
commit	Registra as alterações feitas no repositório local, criando um ponto de salvamento.		
pull	Atualiza o repositório local com as alterações do repositório remoto.		
push	Envia os commits locais para o repositório remoto.		
branch	Cria uma nova ramificação (linha de desenvolvimento) independente.		
merge	Combina as alterações de uma branch com outra (por exemplo, unir tela_login à main).		

## Passo 1: Baixar e Instalar o Git

### Windows:

#### 1. Acesse o site oficial do Git:

- Vá para o [site oficial do Git](#) e clique em "**Download**" para Windows.

#### 2. Inicie o Instalador:

- Após o download, execute o instalador (.exe).

#### 3. Finalizar a Instalação:

- Após concluir as opções, clique em "Next" até o final do processo e, por fim, clique em "Install".

### Linux baseado em Debian:

#### 1. Execute na linha de comando:

```
sudo apt update
sudo apt install git
```

---

## Passo 2: Configurar o Usuário no Git

Agora que o Git está instalado, configure seu nome de usuário e e-mail.

### 1. Configurar o Nome de Usuário:

- No **git bash** ou terminal VS Code, digite:

```
git config --global user.name "Seu Nome"
```

### 2. Configurar o E-mail:

- Em seguida, configure seu e-mail associado ao Git:

```
git config --global user.email "seuemail@dominio.com"
```

---

## Passo 3: Clonar o Repositório

Clonar o repositório desejado para o seu computador.

### 1. Obtenha o URL do Repositório:

- No GitHub, GitLab ou qualquer plataforma onde o repositório está hospedado, copie a URL do repositório (geralmente no formato `https://` ou `git@`).

### 2. Clonar o Repositório:

- No terminal do VS Code, navegue até o diretório onde deseja armazenar o repositório clonado e execute:

```
git clone https://github.com/nome-usuario/nome-repositorio.git
```

### Exemplo:

```
git clone https://github.com/usuario/repo.git
```

### 3. (Opcional) Entrar no diretório do Repositório Clonado:

- Após o download do repositório, entre no diretório do repositório clonado:

```
cd nome-do-repositorio
```

---

## Comandos básicos

Verificar o status dos arquivos ou quais foram alterados:

```
git status
```

Adicionar todos os arquivos modificados ao commit.

```
git add .
```

Adicionar apenas um arquivo específico

```
git add nome_arquivo.html
```

Confirmar o commit:

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

Enviar seu commit (alterações) para o repositório remoto:

```
git push
```

## Práticas que devem ser seguidas para evitar problemas!

Atualizar o repositório local (sempre que iniciar seu trabalho no projeto, execute o comando):

```
git pull
```

Criar e mudar para uma nova branch:

■ Sempre crie uma nova branch para trabalhar, mantendo o projeto original sem alterações.

```
git checkout -b nome_branch
```

Para verificar em qual branch você está, use o comando:

```
git branch
```

**Sempre que uma tarefa está completa e você tem certeza que tudo está funcionado, faça merge da sua branch para a branch principal (normalmente chamada de main ou master) É mais simples fazer no repositório remoto, (Github por exemplo), assim que você acessa o repositório no navegador, aparece uma sugestão de pull request, ou seja, realizar merge entre sua branch de trabalho e a branch principal.**

*Uma pull request (ou merge request) é um pedido para que as alterações feitas em uma branch sejam revisadas e integradas em outra (geralmente a main).*

---

## Desfazer alterações

Desfazer todas as alterações de um arquivo:

```
git checkout nome_arquivo.html
```

Desfazer alterações em todos os arquivos:

```
git checkout .
```

---

## Atualizar e listar branches

Trocar de branch se necessário.

```
git checkout nome-branch
```

Atualizar a lista de branches:

```
git fetch
```

Listar branches existentes:

```
git branch
```