

# O que é Git: tutorial para iniciantes

03/11/2023

Atualizado  
em  
04/07/2025

10min  
de  
leitura



**Douglas Vieira**



Compartilhe:



*Veja como o Git pode ajudá-lo em tarefas do dia a dia e um exemplo de como ele funciona na prática.*

Você já se perguntou como os desenvolvedores mantêm o controle das várias versões de um projeto? **O Git é a resposta para essa necessidade.**

Neste guia completo, vamos explorar o que é o Git, como funciona e oferecer um tutorial passo a passo para iniciantes.

# Introdução ao Git

O Git é um sistema de controle de versão distribuído, criado por Linus Torvalds em 2005. Ele permite que os desenvolvedores **controlem e gerenciem as mudanças em seu código-fonte ao longo do tempo.**

Utilizando o Git, os programadores podem acompanhar diferentes versões de arquivos, voltar a versões anteriores, gerenciar ramificações e facilitar a colaboração entre equipes.

**Veja também:** [O Que é JavaScript: Guia de Programação Web](#)

## Controle de versão com Git

O principal objetivo do controle de versão é **acompanhar as alterações no código-fonte.** O Git registra as mudanças feitas em um projeto, permitindo que os desenvolvedores visualizem o histórico completo das alterações, saibam quem fez as modificações e possam facilmente comparar versões.

Além disso, o Git é um sistema distribuído, o que significa que **cada desenvolvedor tem uma cópia local do repositório**. Isso possibilita trabalhar offline e mesclar suas alterações com o repositório principal posteriormente.



## Comandos Básicos do Git

**git init:** inicia um novo repositório Git;

**git add:** adiciona alterações para serem incluídas no próximo commit;

**git commit:** registra as mudanças no repositório;

**git status:** mostra o estado atual do repositório;

**git log:** exibe o histórico de commits;

**git push:** envia as mudanças para um repositório remoto;

**git pull:** obtém mudanças do repositório remoto para o repositório local;

**git branch:** cria, lista ou exclui ramificações;

**git merge:** combina alterações de diferentes ramificações.

# Tutorial do Git

## Passo 1: Instalação do Git

Para começar, é necessário instalar o Git em seu sistema. Ele pode ser baixado e instalado a partir do site oficial.

Após a instalação, é possível verificar se o Git está corretamente configurado executando o comando **git --version** no terminal.

## Passo 2: Configuração inicial

Após a instalação, é importante configurar o nome de usuário e o endereço de email no Git. Isso pode ser feito utilizando os comandos:

```
bashCopy code
```

```
git config --global user.name "Seu Nome"  
git config --global user.email "seu@email.com"
```

## Passo 3: Iniciando um repositório

Para criar um novo repositório Git, acesse a pasta do seu projeto e execute o comando **git init**. Isso criará um repositório local no diretório.

## Passo 4: Adicionando e confirmar mudanças

Após fazer alterações nos arquivos do projeto, utilize **git add** para incluir as modificações no próximo commit. Em seguida, utilize **git**

**commit** para confirmar as alterações no repositório.

## Passo 5: Ramificações e mesclagem

Ramificações são úteis para desenvolver recursos isolados do restante do código. Utilize **git branch** para criar uma nova ramificação e **git merge** para mesclar as alterações de uma ramificação para outra.

**Veja também:** [O que é GitHub e como utilizá-lo?](#)

## Gerenciamento de código com Git

Além de controle de versão, **o Git é essencial para o gerenciamento de código em equipes**. Com a funcionalidade de ramificações, diversos desenvolvedores podem trabalhar em diferentes recursos simultaneamente, mesclando suas alterações de forma organizada.

## Conclusão

O Git é uma ferramenta poderosa para o desenvolvimento de software, permitindo um controle eficaz de versões e facilitando a colaboração entre desenvolvedores.

Este tutorial básico oferece os fundamentos para começar a utilizar o Git em seus projetos. Com prática e utilização constante, **você se tornará mais confortável e eficiente ao lidar com este sistema de controle de versão.**

Experimente, explore e aprimore suas habilidades com o Git para melhorar sua produtividade no desenvolvimento de software.