**Iniciando um repositório com Git**



**LUZDALIS LOPEZ D´ROMERO**

08/03/2023

**Confira neste artigo:**

* [Iniciando um novo repositório no Git (máquina local)](https://www.alura.com.br/artigos/iniciando-repositorio-git?_gl=1*1nli8ak*_ga*OTgyNjY4ODg3LjE3MDUyNjg1NDY.*_ga_1EPWSW3PCS*MTcwNzUzNzM5My44NC4xLjE3MDc1NDA0MjUuMC4wLjA.*_fplc*VVpDSll6YURWQlJiN1BRWnpTN3pmTEV5UFpzZWd5dWpMdkNiSjVKT0lLQ0JiTnBmNkpjS0VCU0xnTmpuRFpHd0NwWEdtdHVqUndqWHBURnlQZDdVeWxDWTJ4OHU5QTZhUk9yUENubEoyWiUyRiUyRnBjJTJCeFVKNzIzZ04xaWklMkYwcFElM0QlM0Q.#iniciando-um-novo-repositorio-no-git-(maquina-local))
* [Salvando alterações em um novo repositório local](https://www.alura.com.br/artigos/iniciando-repositorio-git?_gl=1*1nli8ak*_ga*OTgyNjY4ODg3LjE3MDUyNjg1NDY.*_ga_1EPWSW3PCS*MTcwNzUzNzM5My44NC4xLjE3MDc1NDA0MjUuMC4wLjA.*_fplc*VVpDSll6YURWQlJiN1BRWnpTN3pmTEV5UFpzZWd5dWpMdkNiSjVKT0lLQ0JiTnBmNkpjS0VCU0xnTmpuRFpHd0NwWEdtdHVqUndqWHBURnlQZDdVeWxDWTJ4OHU5QTZhUk9yUENubEoyWiUyRiUyRnBjJTJCeFVKNzIzZ04xaWklMkYwcFElM0QlM0Q.#salvando-alteracoes-em-um-novo-repositorio-local)
* [Lista de comandos básicos mais utilizados ao trabalhar com versionamento em Git](https://www.alura.com.br/artigos/iniciando-repositorio-git?_gl=1*1nli8ak*_ga*OTgyNjY4ODg3LjE3MDUyNjg1NDY.*_ga_1EPWSW3PCS*MTcwNzUzNzM5My44NC4xLjE3MDc1NDA0MjUuMC4wLjA.*_fplc*VVpDSll6YURWQlJiN1BRWnpTN3pmTEV5UFpzZWd5dWpMdkNiSjVKT0lLQ0JiTnBmNkpjS0VCU0xnTmpuRFpHd0NwWEdtdHVqUndqWHBURnlQZDdVeWxDWTJ4OHU5QTZhUk9yUENubEoyWiUyRiUyRnBjJTJCeFVKNzIzZ04xaWklMkYwcFElM0QlM0Q.#lista-de-comandos-basicos-mais-utilizados-ao-trabalhar-com-versionamento-em-git)
* [Recomendações extras:](https://www.alura.com.br/artigos/iniciando-repositorio-git?_gl=1*1nli8ak*_ga*OTgyNjY4ODg3LjE3MDUyNjg1NDY.*_ga_1EPWSW3PCS*MTcwNzUzNzM5My44NC4xLjE3MDc1NDA0MjUuMC4wLjA.*_fplc*VVpDSll6YURWQlJiN1BRWnpTN3pmTEV5UFpzZWd5dWpMdkNiSjVKT0lLQ0JiTnBmNkpjS0VCU0xnTmpuRFpHd0NwWEdtdHVqUndqWHBURnlQZDdVeWxDWTJ4OHU5QTZhUk9yUENubEoyWiUyRiUyRnBjJTJCeFVKNzIzZ04xaWklMkYwcFElM0QlM0Q.#recomendacoes-extras:)

*Antes de começar a desenvolver este conteúdo, saiba que este tema é desenvolvido em uma sequência de três artigos, sendo este o primeiro. Recomendamos a leitura do segundo artigo,*[*Criando um repositório no GitHub*](https://www.alura.com.br/artigos/criando-repositorio-remoto-github)*e do terceiro,*[*Clonando um repositório com Git e GitHub*](https://www.alura.com.br/artigos/clonando-repositorio-git-github?utm_source=gnarus&utm_medium=timeline) a fim de ter uma visão completa do assunto.

Toda pessoa iniciante em programação deve levar em consideração a necessidade de saber o que é Git e Github, a final são ferramentas que serão muito úteis ao longo de sua carreira.

Imagine que você já criou seu primeiro projeto, porém ainda não o versionou? Se quiser que o git comece a rastrear seu código para guardar um registro e controlar as mudanças realizadas nele ao mesmo tempo, mas ainda não tem certeza de como fazê-lo, este é o artigo ideal para você, uma vez que te guiará sobre como lidar com este versionamento.

**Iniciando um novo repositório no Git (máquina local)**

Primeiramente, se ainda não tem o git instalado, recomendo que faça o [download e o instale no seu computador](https://git-scm.com/downloads). Normalmente, a instalação é feita por padrão no disco rígido C, é aí que esta prática será baseada.

Com o git instalado, abra o terminal Git Bash, que se iniciará de forma predeterminada no diretório principal (~) ou /c/Users// do seu usuário, como exemplificado na seguinte imagem:

Diagrama

Descrição gerada automaticamente com confiança média

Se deseja determinar o diretório atual, digite pwd na linha de comando $, como demonstrado abaixo:

Interface gráfica do usuário

Descrição gerada automaticamente com confiança média

**Informações importantes:**

* Se for usar o git pela primeira vez, primeiro deverá configura-lo, estabelecendo seu nome de usuário e endereço de email, já que esta informação será usada de maneira inalterável nos “commits” que enviar.
  + Abaixo você pode encontrar os comandos para a configuração destas variáveis globais:
  + $ git config --global user.name "Nome do usuário"
  + $ git config --global user.email "seu@email.com"
* Antes de iniciar deve saber em qual rota se encontra a pasta do projeto, evitando maior trabalho ao usar o terminal.

1. O comando ls listará os arquivos presentes no diretório principal, depois, você deverá acessar a pasta Desktop; nela se encontra o projeto no qual se deseja iniciar o repositório:

Tela de computador

Descrição gerada automaticamente

1. Em seguida, listando os arquivos com o comando ls, serão apresentados os arquivos da pasta Desktop e devemos acessar a pasta portfolio onde se encontram os arquivos do projeto:

Texto

Descrição gerada automaticamente

1. Finalmente, executaremos o comando git init para começar a rastrear o repositório indicado:

Texto

Descrição gerada automaticamente

Uma vez executado o comando, aparecerá a seguinte mensagem no terminal, indicando que o repositório foi iniciado:

Initialized empty Git repository in C:/Users/Luzdalis/Desktop/portfolio/.git/

O git mostrará o nome da branch principal que está em uso que, normalmente, se chama **master**.

**Nota**: Para garantir a compatibilidade no futuro é recomendado atualizar o nome da branch principal, de **master** para **main**, com o seguinte comando:

$ git branch -M main

Tela de computador

Descrição gerada automaticamente

**Salvando alterações em um novo repositório local**

É importante que considere que cada arquivo do seu repositório pode ter um, dos dois estados: rastreado ou não-rastreado.

Os arquivos rastreados (***tracked files*** em inglês) são os arquivos que o git está acompanhando e podem ter três status: não-modificados (***unmodified***), modificados (***modified***) ou prepadados (***staged***).

Por outro lado, os arquivos não-rastreados (***untracked files***) são todos aqueles os quais o git não acompanha e não se encontram na área de preparação.

Para ilustrar, os estados citados acima estão dispostos na imagem abaixo, extraída da documentação oficial do Git:

Linha do tempo

Descrição gerada automaticamente

Se tem interesse em se aprofundar um pouco mais em como salvar alterações no repositório, pode acessar a documentação oficial do git [clicando aqui](https://git-scm.com/book/es/v2/Fundamentos-de-Git-Saving-cambios-en-el-Repositorio).

Nesta ordem de ideias se iniciará o controle e registro de versões do projeto.

**Primeiro passo: verificar o estado do repositório**

Para conhecer o estado em que se encontram os arquivos do repositório, utilize o seguinte comando:

$ git status

É importante destacar que ao iniciar um repositório, a branch principal (**main**) estará vazia, ou seja, não possuí arquivos, portanto, o Git não está rastreando nenhuma alteração.

Texto

Descrição gerada automaticamente

*Arquivos não-rastreados (****untracked files****)*

**Segundo passo: adicionar arquivos para rastrear**

Para começar a rastrear os arquivos, executamos o comando git add, que adiciona o conteúdo ao rastreio do Git. Podemos faze-lo das seguintes formas:

* $ git add nome\_do\_arquivo /\*\*\*Adiciona apenas o arquivo mencionado\*\*\*/ ou
* $ git add . /\*\*\*Adiciona todos os arquivos do projeto de uma só vez\*\*\*/

Texto

Descrição gerada automaticamente com confiança média

*Mudanças a serem feitas (****changes to be committed****)*

**Terceiro passo: confirmar os arquivos**

Para confirmar os arquivos nomeamos as alterações com um nome que faça sentido, e descreva as mudanças que estamos realizando em uma frase simples. Utilizamos, então, o seguinte comando:

$ git commit -m "Mensagem descritiva do commit"

Texto

Descrição gerada automaticamente

**Lista de comandos básicos mais utilizados ao trabalhar com versionamento em Git**

* **git config:** Este é o primeiro comando a ser executado após instalar o Git. Ele possibilita configurar seu nome e endereço de email que ficará vinculado às alterações.
  + $ git config --global user.name "Nome do usuário"
  + $ git config --global user.email "seu@email.com"
* **git init:** Inicia ou cria um repositório.
  + $ git init
* **git status:** Permite visualizar o estado do repositório.
  + $ git status
* **git add:** Prepara o conteúdo para o próximo commit.
  + $ git add nome\_do\_arquivo
  + $ git add .
* **git commit:** Salva o conteúdo atual junto com uma mensagem de registro do usuário que descreve as alterações.
  + $ git commit -m "Mensagem descritiva do commit"
* **git clone:** Clona um repositório existente.
  + $ git clone [url]
* **git branch:** Uma branch nada mais é do que uma ramificação dentro do repositório. Este comando pode ser utilizado de diversas maneiras.
  + $ git branch // Mostra as branches existentes em um repositório
  + $ git branch nome\_da\_branch // Cria uma nova branch
  + $ git branch -M nome\_da\_branch // Renomeia a branch atual
* **git log:** Exibe um histórico de commits. Este comando pode ser utilizado de diversas formas.
  + $ git log
  + $ git log --oneline
* **git remote:** Exibe o repositório remoto.
  + $ git remote
  + $ git remote -v
* **git pull:** Baixa o conteúdo do repositório remoto, atualizando automaticamente o repositório local.
  + $ git pull
* **git push:** Envia o conteúdo do repositório local, atualizando automaticamente o repositório remoto.
  + $ git push

**Recomendações extras:**

* Para a criação de pastas ou diretórios, evite escrever os nomes com acentos e espaços, isto evitará rupturas ou erros no projeto ao executar os comandos no terminal.
* Se você é um novo usuário de Git e GitHub é muito importante que realize a etapa de configuração das variáveis globais e da chave SSH. Desta forma, evitará problemas e poderá visualizar o versionamento corretamente. Um dos erros mais comuns vistos como resultado de falhas nesta etapa é o seguinte:

Texto

Descrição gerada automaticamente

*Permissão negada (****permission to… denied to…****)*

Este erro aparece quando clonamos um repositório utilizando o caminho da URL visto anteriormente e tentamos enviar atualizações ao repositório remoto e o GitHub não reconhece quem está realizando estes envios. Para solucionar este problema, devemos criar uma chave SSH que autentique e possa fazer alterações neste repositório. Esta ação precisará ser feita apenas uma vez. A partir de então, seu usuário estará autenticado para realizar alterações em qualquer repositório existente dentro do seu GitHub sem nenhum problema.

Para finalizar, abaixo, temos os links dos artigos que compõem esta série, os mesmos podem te auxiliar no início de sua trajetória com versionamento com Git e GitHub. Estes artigos fortalecem sua aprendizagem para sua carreira profissional:

* O segundo artigo [Criando um repositório no GitHub](https://www.alura.com.br/artigos/criando-repositorio-remoto-github)
* O terceiro artigo [Clonando um repositório com Git e GitHub](https://www.alura.com.br/artigos/clonando-repositorio-git-github?utm_source=gnarus&utm_medium=timeline)