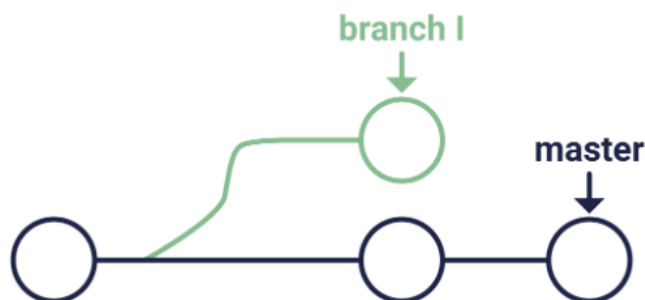


GIT

Um repositório Git é um espaço de **armazenamento virtual para projetos**, utilizado para gravar e deixar sempre acessíveis as versões do código. O repositório local é a pasta/diretório de sua máquina, no qual está seu projeto.

No Git, o código-fonte do projeto é o tronco principal, chamado *master*. A partir do *master*, são criadas ramificações, chamadas *branches*. Cada tarefa do projeto é feita em uma *branch* e, depois de concluída, unificada no *master*. Assim, podemos ter várias tarefas de vários programadores sendo feitas ao mesmo tempo independentemente e, mesmo assim, manter a integridade do *master*.



Criando um repositório remoto no GitHub

Para deixar o projeto acessível, é necessário criar um repositório remoto, ou seja, publicá-lo online usando a plataforma gratuita chamada GitHub.

Acesse o site do GitHub
(<https://github.com>).

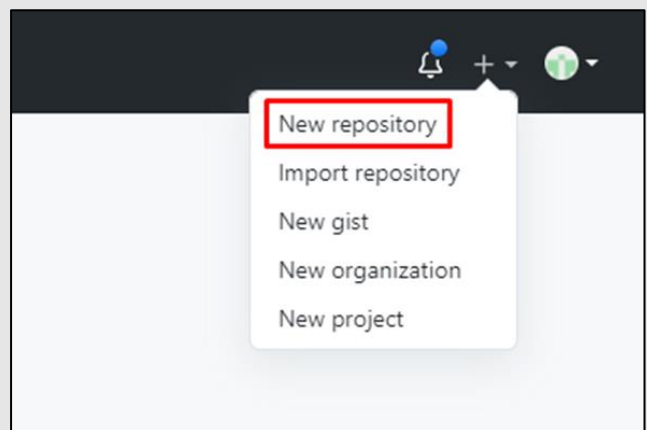
Na tela da *home*,
clique em “Sign up”
para criar uma conta.



Git – Fonte: <https://github.com/>.

Feito o cadastro no site,
clique em “Sign in”
para acessar seu perfil.

Depois, **clique em “New repository”**, para criar o repositório.



Git – Fonte: <https://github.com/>.

Agora, siga o passo a passo:

The screenshot shows the GitHub 'Create a new repository' page. It includes a title 'Create a new repository', a subtitle 'A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)', and a progress bar at the top with six numbered steps. The form fields are: 1. Owner (dropdown menu), 2. Repository name (text input), 3. Description (optional) (text area), 4. Visibility (radio buttons for Public and Private), and 5. Initialization options (checkboxes for 'Add a README file', 'Add .gitignore', and 'Choose a license'). A green 'Create repository' button is at the bottom, labeled with step 6.

Git – Fonte: <https://github.com/new>.

- 1) Preencha com o tipo de usuário.
- 2) Atribua um nome ao repositório.
- 3) Opcionalmente, faça uma breve descrição do repositório.
- 4) Escolha o tipo de visualização do repositório:
 - **Public:** qualquer pessoa pode ver o projeto.
 - **Private:** apenas o proprietário pode ver o projeto.

- 5) Escolha quais arquivos podem ser criados com o repositório:
 - **README:** arquivo de informações sobre o projeto, que funciona como cartão de visitas e histórico. É opcional, mas altamente recomendado.
 - **.gitignore:** arquivo especial do Git. Os arquivos listados dentro dele são ignorados pelo Git.
 - **License:** licença do seu projeto.
- 6) Para finalizar, clique em “*Create repository*” para criar o repositório.

Criando um repositório local no Git

Depois de criado o repositório remoto, é necessário cloná-lo no repositório local. Desse modo, é possível publicar arquivos locais ou importar arquivos remotos.

O primeiro passo é acessar o site <https://git-scm.com/downloads>, baixar e instalar o Git.



GitHub – Fonte: <https://github.com/>.

Agora, abra um *prompt* de comando em seu computador para fazer as configurações iniciais do Git e digite os seguintes comandos:

- ***git config --global user.name “Seu nome”***

Importante: O seu nome deve corresponder ao seu usuário do GitHub. Por exemplo, caso seu usuário criado seja joaodasilva-123456789, **faça *git config --global user.name “joaodasilva-123456789”***, aperte “*enter*” e digite:

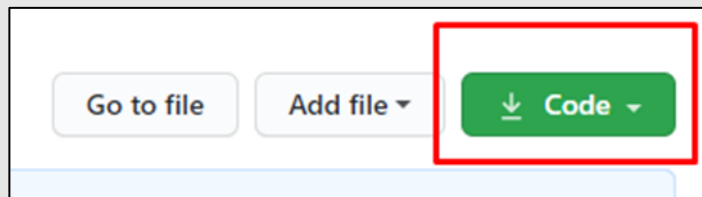
- ***git config --global user.email exemplo@seuemail.com.br***

Importante: O e-mail que você irá configurar deve ser o mesmo utilizado para criar a conta no GitHub.

Começando a usar o Git

Agora é o momento de clonar o repositório remoto em sua máquina.

Acesse o site do GitHub, entre no repositório remoto do projeto e clique em “**Code**”, depois, em “**HTTPS**” e copie o link.



GitHub – Repository – Fonte: <https://github.com/>.

Em sua máquina, dentro da pasta do projeto, clique com o botão direito do mouse, selecione a opção “**git bash here**” para abrir o terminal do Git e digite os comandos:

- **git init:** para iniciar o Git.
- **git remote add origin:** para colar a URL copiada do GitHub.

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

Dessa forma, a sua máquina local será sincronizada com o repositório do Git.

Além desses dois comandos iniciais, há outros comandos que são importantes:

- ***git status***: utilizado para verificar os arquivos alterados;
- ***git clone***: utilizado para copiar um repositório local;
- ***git add nome-do-arquivo***: adiciona o arquivo alterado ao *staging*;
- ***git add .***: adiciona todos os arquivos alterados;
- ***git commit -m "comentário"***: confirma e salva as alterações no histórico;
- ***git push origin master***: publica os arquivos locais no repositório remoto;
- ***git pull***: importa as atualizações do repositório remoto para a sua máquina;
- ***git log***: utilizado para ver o histórico do projeto, ou seja, ver os *commits* já feitos em uma *branch*;
- ***git branch***: cria uma nova *branch*;
- ***git checkout***: utilizado para navegar entre as *branches*;
- ***git merge***: utilizado para fazer a junção de *branches*;
- ***git pull origin master***: utilizado para atualizar o repositório local.

Dica!

Antes de iniciar os trabalhos, é uma boa prática fazer o comando ***git pull origin nome-branch***. Assim, evita-se o trabalho em um código desatualizado, minimizando conflitos.



Saiba mais

Para mais informações, clique e acesse o site do Git <https://git-scm.com/docs> e leia a documentação disponível.



Dica!

Lembre-se de fazer em seu repositório:

- ***git pull origin master***: caso tenha feito algum *push*. Dessa forma, você atualizará sua *branch*.
- ***git status***: para verificar todos os arquivos que você alterou.
- ***git add .*** : para adicionar todos os arquivos alterados de uma só vez.
- ***git commit -m "seu-comentario"***: para adicionar um comentário ao *push* que você fará. lembre-se de ser curto e direto.
- ***git push origin master***: para enviar seu projeto.

