

Instalação e primeiro uso do Git

Instalação do GIT no Windows

<https://git-for-windows.github.io> (Clique em download, baixe a versão 32 ou 64 bits conforme seu sistema operacional)

1. Execute o arquivo EXE.
2. Selecione NEXT em todas as opções, sem mudar nada, até chegar no passo "Adjusting your PATH environment". Neste passo, selecione "Use Git from the Windows Command Prompt". Clique em NEXT.
3. Na próxima opção, "Configuring the line ending conversions", selecione "Checkout as-is, commit as-is". Clique em NEXT.
4. O restante não é necessário mudar. Finalize.

Usando o prompt do Git

1. Abra o Git Bash, que é um prompt de comando especial do Git para Windows, que simula um ambiente Linux.
2. Use o comando "`cd`" (change directory) para ir até a pasta na qual você extraiu o pacote da aula.
3. Teste o comando "`diff`" para os arquivos `game_old.js` e `game_new.js`.

A sintaxe é: `diff -u velho novo`

Lembrando que no Windows também existe um comando "difference". Se quiser pode testar, no prompt CMD do Windows.

FC — File Compare — Windows

Diff — Difference — Mac / Linux

Primeiro uso do Git

1. No Git Bash, crie uma pasta de trabalho. Chame-a de "roteiro01". Para criar uma pasta no terminal baseado em Linux, o Git Bash, use o comando `mkdir roteiro01`. Esse comando criará uma pasta de nome "roteiro01".
2. Mude o diretório para a pasta de você criou: `cd roteiro01`.
3. Inicie o repositório Git com o comando `git init`. Esse comando cria a pasta ".git" onde o Git irá manter seu repositório. Não trabalhamos nessa pasta, é o Git que trabalha nela, inclusive, ela fica oculta no sistema. Lembre-se que esse repositório é local, ou seja, está apenas na sua máquina, dentro da pasta que você criou. O Git mostrará uma mensagem parecida com essa, mudando apenas o caminho indicado:

```
Initialized empty Git repository in /Users/viniciusduarte/
Documents/teste_git/.git/
```

4. Use o comando `git status`. Esse comando mostra o estado atual do seu repositório, da sua staging area e do seu working directory. Observe que ele mostrará uma mensagem assim:

```
On branch master
```

```
Initial commit
```

```
nothing to commit (create/copy files and use "git add" to track)
```

5. Crie um arquivo texto dentro da pasta "roteiro01" com a seguinte linha "Primeiro uso do Git", seguida de uma linha em branco, salve-o como "arquivo01.txt".
6. Use o comando `git status` novamente. O resultado deverá ser este:

```
On branch master
Initial commit
Untracked files:
  (use "git add <file>..." to include in what will be committed)
    arquivo01.txt
nothing added to commit but untracked files present (use "git add"
to track)
```

7. Observe que o arquivo01.txt não foi adicionado ao staging area, por isso está em vermelho. Para adicioná-lo, use o comando `git add arquivo01.txt`.
8. Use o comando `git status` para verificar a situação atual do repositório. O resultado deverá ser o seguinte, que significa que o arquivo01.txt agora está na staging area:

```
On branch master

Initial commit

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)

    new file:   arquivo01.txt
```

9. Modifique o arquivo01.txt, adicionando uma linha no fim. Essa linha deve conter seu nome e sua matrícula. Quando você salvar o arquivo com suas novas alterações, o arquivo01.txt dos seu working directory (esse que você acabou de salvar) estará diferente do arquivo que está em sua staging area (o que você adicionou com o comando `git add`).
10. Use o comando `git diff arquivo01.txt` para ver a diferença entre os dois arquivos. O resultado deverá ser semelhante a esse:

```
diff --git a/arquivo01.txt b/arquivo01.txt
index 240f7bb..6ee770c 100644
--- a/arquivo01.txt
+++ b/arquivo01.txt
@@ -1,2 @@
    Primeiro uso do Git
+meunome minhamatricula
```

11. Use o comando `git status` para ver que o arquivo01.txt que está na staging area foi modificado no working directory, o resultado deverá ser semelhante a este:

```
On branch master
```

```
Initial commit
```

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
```

```
    new file:   arquivo01.txt
```

```
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git checkout -- <file>..." to discard changes in working
directory)
```

```
    modified:   arquivo01.txt
```

12. Para atualizar o arquivo da staging area, use o comando `git add arquivo01.txt` novamente.
13. Use o comando `git status` para ver que o arquivo da staging area agora é o mesmo do working directory. O resultado deverá ser semelhante a este:

```
On branch master
```

```
Initial commit
```

```
Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
```

```
    new file:   arquivo01.txt
```

14. Use o comando `git commit -m "Meu primeiro commit"` para fazer seu primeiro commit no repositório local. Provavelmente o commit não será realizado, o Git dará a seguinte mensagem:

```
*** Please tell me who you are.
```

```
Run
```

```
git config --global user.email "you@example.com"
git config --global user.name "Your Name"
```

```
to set your account's default identity.
Omit --global to set the identity only in this repository.
```

15. Observe que o próprio Git já informa o que aconteceu. O Git não faz um commit sem saber quem está fazendo, portanto ele pede suas “credenciais”, mesmo aqui não sendo necessário nenhuma validação das informações que você forneceu. Execute os comando que o Git pediu para você executar, substituindo os exemplos por seu e-mail e nome.

16. Tente fazer o seu primeiro commit novamente. O resultado deverá ser semelhante a esse.

```
[master (root-commit) 0f614f4] primeiro commit  
1 file changed, 2 insertions(+)  
create mode 100644 arquivo01.txt
```

Exercício

Crie um novo arquivo no diretório “roteiro01” com as seguintes informações pessoais: nome, e-mail e o ano que você entrou na faculdade. Adicione esse arquivo à staging area e faça o commit, explicando que arquivo é esse.

Lembrando que os commits devem ser usados para:

- Consertar um bug
- Adicionar funcionalidade
- Melhorar documentação

Resumo: “Um commit por mudança lógica”