

Tempo estimado de leitura:
5 min

Módulo #3

Guias de estudo

Introdução

Nesse módulo, estudamos sobre **versionamento de código**. Antigamente era muito comum disponibilizar versões dos nossos programas por meio de pastas que levavam a identificação de cada versão (ex.: v1, v2, v3, etc...). Graças ao **Git** e outras ferramentas de versionamento de código, isso passou a ser obsoleto.

Vamos para o nosso resumo do módulo?

O Git

O **Git** pode ser baixado nesse endereço:

<https://git-scm.com/>.

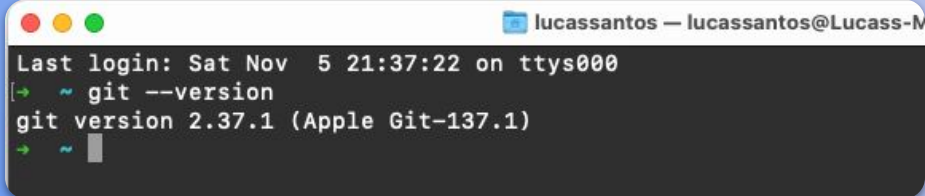
Para o **Windows**, após instalar o **Git** você terá acesso ao programa **Git Bash**. O **Git Bash** possibilita executar as principais operações do **Git**. No caso do **Linux**, consulte a melhor forma de instalar o **Git** para sua distribuição Linux. No caso do **Ubuntu**, basta executar o comando "**sudo apt-get install git-all**" no **Terminal**. Para o **macOS**, o **Git** já virá instalado por padrão.

Testando e instalando o Git

Após instalado, abra o **Git Bash** no **Windows** ou o **Terminal** no **Linux** ou **macOS**. Então escreva o seguinte comando:

```
"git --version"
```

Deverá ser exibida na tela a versão do Git instalada.



```
lucassantos — lucassantos@Lucass-M
Last login: Sat Nov  5 21:37:22 on ttys000
➜ ~ git --version
git version 2.37.1 (Apple Git-137.1)
➜ ~
```

Principais comandos do Git

git init: é o comando a ser dado para inicializar o versionamento de um projeto

git status: exibe os arquivos e diretórios que foram modificados ou estão prontos para serem comitados a partir da área de staging

git diff: mostra as alterações em cada um dos arquivos que foram modificados

git add: Adiciona todos os arquivos modificados à área de staging

git commit -m "Mensagem de commit": cria um novo commit com os arquivos que foram adicionados na área de staging

git log: mostra todos os commits realizados

git push origin main: envia os commits feitos na sua máquina para um repositório remoto

No link de documentação oficial abaixo, você pode aprender muito mais sobre os comandos mencionados acima e outros existentes:

<https://git-scm.com/docs>

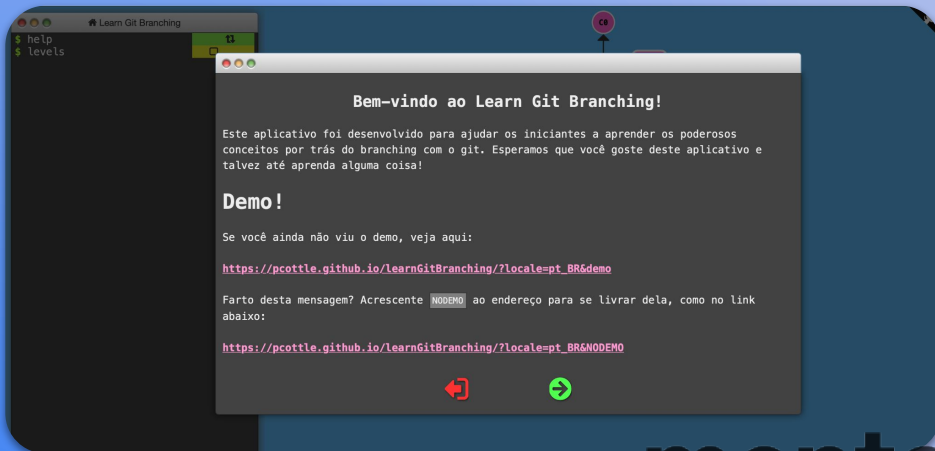
Recomendamos que você invista um tempo para estudar mais detalhes sobre os comandos do Git.

Branches

Branches são espaços separados em um repositório feitos para receber alterações sem impactar o ramo principal do projeto (normalmente chamado de branch "**main**" ou "**master**").

No link a seguir, é possível aprender de forma interativa como criar e manipular branches:

https://learngitbranching.js.org/?locale=pt_BR



5

GitHub e GitLab

O **GitHub** e **GitLab** são serviços online e gratuitos que permitem que você armazene seus projetos e, caso queira, compartilhe eles com o mundo todo. Eles podem ser utilizados como seu portfólio.

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

GitLab: <https://about.gitlab.com/>

Esses serviços também são como uma rede social para desenvolvedores e várias pessoas também podem trabalhar de forma colaborativa em um mesmo projeto.

6