

## Seção 1

O mini curso se inicia com uma breve introdução ao que motivou a criação do git, o controle de versão, pois antes era necessário armazenar várias versões de um aplicativo para caso fosse encontrado algum erro ou fosse necessário voltar, houvesse a possibilidade de retornar a uma versão funcional.

para finalizar a seção o instrutor explica o que é o github, o github é um serviço para armazenar aplicações que utilizam o versionamento por git em repositórios, armazenando seus projetos na nuvem, disponibilizando-o publicamente.

## Seção 2

A seção 2 se trata do download e configuração do git, para configurá-lo foi necessário especificar o nome de usuário que eu queria, o meu email e o editor de texto que eu utilizo.

```
C:\Users\shink>git config --global user.name "LuizNagatani"

C:\Users\shink>git config --global user.email "luiz.nagatani@gmail.com"

C:\Users\shink>git config --global core.editor "vim"

C:\Users\shink>git config user.name
LuizNagatani

C:\Users\shink>|
```

## Seção 3

É na seção 3 que se inicia a utilização do git, sendo explicado como criar um repositório e adicionar arquivos, editando-os através de um aplicativo editor de texto, para executar os testes foi utilizado o vim no cmd.

```

PS C:\Users\shink\git-course\repo_teste> vim README.md
PS C:\Users\shink\git-course\repo_teste> dir

Diretório: C:\Users\shink\git-course\repo_teste

Mode                LastWriteTime         Length Name
----                -
-a-----          13/08/2024      21:23           523 .README.md.un~
-a-----          13/08/2024      21:23           76 README.md

```

Ao criar um arquivo, deve se realizar um commit para adicioná-lo ao repositório usando o comando git add "nome do arquivo" para adicionar um arquivo a um commit

```

PS C:\Users\shink\git-course\repo_teste> git status
On branch master

No commits yet

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  .README.md.un~
  README.md

nothing added to commit but untracked files present (use "git add" to track)
PS C:\Users\shink\git-course\repo_teste> git add README.md
PS C:\Users\shink\git-course\repo_teste> git status
On branch master

No commits yet

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

Untracked files:
  (use "git add <file>..." to include in what will be committed)
  .README.md.un~

PS C:\Users\shink\git-course\repo_teste> git commit -m "adicionado o readme"
[master (root-commit) 433221b] adicionado o readme
1 file changed, 1 insertion(+)
create mode 100644 README.md

```

É possível identificar autores e horários de commits através de um git log, permitindo que se crie uma linha do tempo de updates e que o autor de um commit danoso seja responsabilizado. Também é possível filtrar logs com base em autores ou listar quantos commits cada autor fez e quais arquivos foram mudados pelos mesmos.

```

PS C:\Users\shink\git-course\repo_teste> git log
commit 433221b3cf4f1190738b96c4ed7462c29dab7ce5 (HEAD -> master)
Author: LuizNagatani <luiz.nagatani@gmail.com>
Date:   Tue Aug 13 22:00:59 2024 -0300

    adicionado o readme

```

Outra função apresentada é o git diff, que permite ver diferenças entre atualizações de arquivos seja antes ou depois de um commit do mesmo.

```
diff --git a/README.md b/README.md
index b5f73bc..5888416 100644
--- a/README.md
+++ b/README.md
@@ -1,3 @@
     esse é o teste do vim no cmd, ta mt daora, valeu LAMIA! (emoji de joinha)
+
+Estou mudando aqui para realizar o teste do comando git diff :)
PS C:\Users\shink\git-course\repo_teste>
```

Por fim, é apresentado o comando git checkout, que permite desfazer uma alteração antes do commit.

```
--- a/README.md
+++ b/README.md
@@ -1,3 +1,5 @@
     esse é o teste do vim no cmd, ta mt daora, valeu LAMIA! (emoji de joinha)

    Estou mudando aqui para realizar o teste do comando git diff :)
+
+agora é pra testar o de desfazer:(
PS C:\Users\shink\git-course\repo_teste> git checkout README.md
Updated 1 path from the index
PS C:\Users\shink\git-course\repo_teste> git diff
PS C:\Users\shink\git-course\repo_teste>
```

Para desfazer um commit é necessário o git reset, podendo ser efetuado no modo soft, mixed ou hard.

Soft: desfaz o commit e retorna ao estado pré-commit pós-add

Mixed: desfaz o commit e o add mas mantém a alteração

Hard: desfaz todas as alterações feitas desde o commit selecionado

```
PS C:\Users\shink\git-course\repo_teste> git reset --hard
HEAD is now at aa07e44 estraguei o repo
PS C:\Users\shink\git-course\repo_teste> git reset --hard e8beab46634b12405caacac86762c4b84bb17b36
HEAD is now at e8beab4 Please enter the commit message for your changes. Lines starting
PS C:\Users\shink\git-course\repo_teste> git log
commit e8beab46634b12405caacac86762c4b84bb17b36 (HEAD -> master)
Author: LuizNagatani <luiz.nagatani@gmail.com>
Date: Tue Aug 13 22:08:44 2024 -0300

    adicionado o readme

Please enter the commit message for your changes. Lines starting
commit 433221b3cf4f1190738b96c4ed7462c29dab7ce5
Author: LuizNagatani <luiz.nagatani@gmail.com>
Date: Tue Aug 13 22:00:59 2024 -0300

    adicionado o readme
PS C:\Users\shink\git-course\repo_teste> |
```

## Seção 4

A seção 4 aborda dispositivos remotos, mais especificamente o GitHub, ferramenta já utilizada atualmente para disponibilizar os relatórios deste bootcamp.



Um conceito novo apresentado é o das chaves SSH, chaves utilizadas para autenticar computadores ao github

Em seguida mostra operações com repositórios remotos, como efetuar pushes, clonar repositórios remotos para uma máquina e finalmente como criar um fork de um repositório, clonando-o para sua máquina, porém um commit do mesmo não pode ser enviado para o repositório base, pois apenas o dono e pessoas permitidas podem realizar mudanças em um repositório.

## SSH keys

This is a list of SSH keys associated with your account.

### Authentication keys

**PC Toledo**

SHA256: g7H9yJ0vAp/LhXPJHjTx7IwC7ZZp2ajs

Added on Aug 13, 2024

Never used — Read/write

## Seção 5

A seção 5 se trata de branches, explicando como criar ramificações de um repositório, que pode ser alterado sem alterar a ramificação inicial, permitindo que várias pessoas trabalhem simultaneamente em um repositório sem conflitos de arquivos.

```
PS C:\Users\shink\git-course\repo_teste> git checkout -b teste
Switched to a new branch 'teste'
PS C:\Users\shink\git-course\repo_teste> git branch
  master
* teste
PS C:\Users\shink\git-course\repo_teste> |
```

Alguns comandos relacionados com branches são apresentados, por exemplo o checkout que move o “cursor” para a branch selecionada ou a cria caso inexistente, e o argumento -d para o git branch, que deleta o branch especificado.

```

PS C:\Users\shink\git-course\repo_teste> git checkout -b teste
Switched to a new branch 'teste'
PS C:\Users\shink\git-course\repo_teste> git branch
  master
* teste
PS C:\Users\shink\git-course\repo_teste> git checkout master
Switched to branch 'master'
PS C:\Users\shink\git-course\repo_teste> git branch -D teste
Deleted branch teste (was e8beab4).
PS C:\Users\shink\git-course\repo_teste> |

```

Por último são mostradas duas formas de unir a branch master com uma branch criada, o merge e o rebase, no master é criado um commit com base na branch master e na branch desejada, que une as duas, porém polui os logs de commits.

Já o rebase cria um commit e o posiciona após o último commit do branch master, evitando um log poluído e extenso, porém perdendo a ordem cronológica do mesmo

```

PS C:\Users\shink\git-course\repo_teste> git checkout -b teste
Switched to a new branch 'teste'
PS C:\Users\shink\git-course\repo_teste> vim commitai
PS C:\Users\shink\git-course\repo_teste> git add commitai
PS C:\Users\shink\git-course\repo_teste> git commit
[teste 85db074] add commitai
 1 file changed, 1 insertion(+)
 create mode 100644 commitai
PS C:\Users\shink\git-course\repo_teste> git checkout master
Switched to branch 'master'
PS C:\Users\shink\git-course\repo_teste> git merge teste
Updating 27180f2..85db074
Fast-forward
 commitai | 1 +
 1 file changed, 1 insertion(+)
 create mode 100644 commitai
PS C:\Users\shink\git-course\repo_teste> |

```

Merge

```

PS C:\Users\shink\git-course\repo_teste> git checkout teste
Switched to branch 'teste'
PS C:\Users\shink\git-course\repo_teste> vim votestarebase
PS C:\Users\shink\git-course\repo_teste> git add votestarebase
PS C:\Users\shink\git-course\repo_teste> git commit
[teste a5d011c] ó o rebase
1 file changed, 1 insertion(+)
create mode 100644 votestarebase
PS C:\Users\shink\git-course\repo_teste> git checkout master
Switched to branch 'master'
PS C:\Users\shink\git-course\repo_teste> git rebase teste
Successfully rebased and updated refs/heads/master.
PS C:\Users\shink\git-course\repo_teste> |

```

## Rebase

## Vídeo youtube

Foi disponibilizado como conteúdo opcional um mini curso sobre github desktop, exibindo como realizar todos os comandos aprendidos no curso da udemy neste aplicativo, que efetua as mesmas operações porém com uma interface em vez do terminal, facilitando muitas operações, principalmente a conexão com dispositivos remotos.

