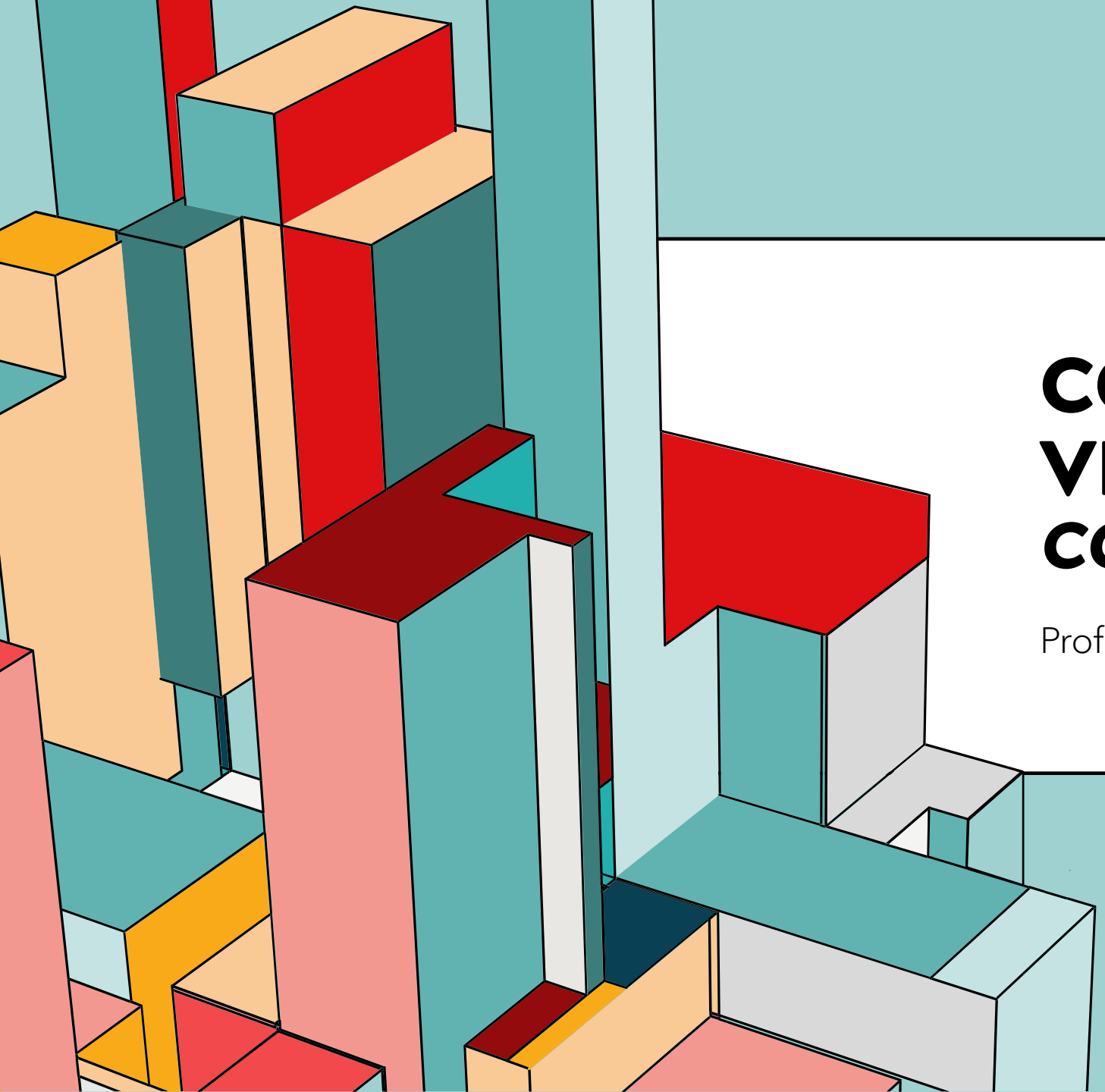


CONTROLE DE VERSÃO (GIT)
COMANDOS BÁSICOS

Prof. Dr. Dilermando Piva Jr

DIRETÓRIO .GIT

Quando você inicializa o rastreamento dos arquivos de um determinado diretório com o GIT INIT
O Git cria automaticamente um arquivo .GIT.

NUNCA DELETE ESSE ARQUIVO. Pois é lá dentro que o GIT fará todos os controles de rastreamento e status dos arquivos.

DIRETÓRIO .GIT

Dentro desse diretório existem muitos arquivos e diretórios.
Vamos conhece-los e explorá-los na medida que formos avançando...

Arquivo: config

```
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
```

PRIMEIRA COISA QUE TEMOS QUE FAZER QUANDO INICIALIZAMOS O GIT (LOCAL)

A primeira tarefa que temos que fazer ao inicializar o rastreamento do GIT em um determinado repositório é a sua configuração:

git config user.name “Dilermando Piva Jr”
git config user.email pivajr@gmail.com

Mais informações:
https://git-scm.com/book/pt-pt/v2/Come%C3%A7ando-Configura%C3%A7%C3%A3o-Inicial-do-Git

DEPOIS DA CONFIGURAÇÃO INICIAL...

Arquivo: config

```
[core]
    repositoryformatversion = 0
    filemode = true
    bare = false
    logallrefupdates = true
[user]
    name = Dilermando Piva Jr
    email = pivajr@gmail.com
```

PRIMEIRA COISA QUE TEMOS QUE FAZER QUANDO INICIALIZAMOS O GIT (GLOBAL)

Fazendo as configurações passarem a ser globais:

git config --global user.name “Dilermando Piva Jr”
git config --global user.email pivajr@gmail.com

ATENÇÃO:
Se você está num computador público... Utilizado por múltiplas pessoas, utilize a inicialização LOCAL.
Se estiver na sua máquina.... Seu computador próprio: utilize GLOBAL.

COMANDOS BÁSICOS
GIT ADD

Para adicionar os arquivos para que o GIT inicie seu rastreamento.

git add nome.py

Para adicionar todos os arquivos:

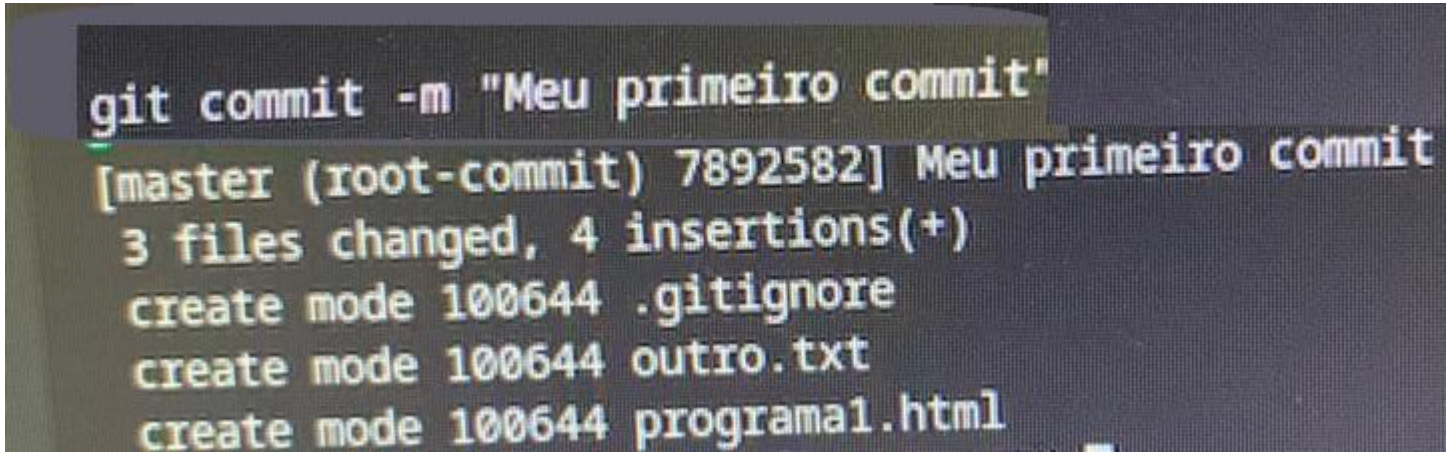
git add .

COMANDOS BÁSICOS
GIT COMMIT

Para cada commit (envio) realizado o GIT vai realizar o empacotamento da situação dos arquivos, criando uma fotografia desse momento.

git commit -m “Descrição do que está sendo feito”

Exemplo:

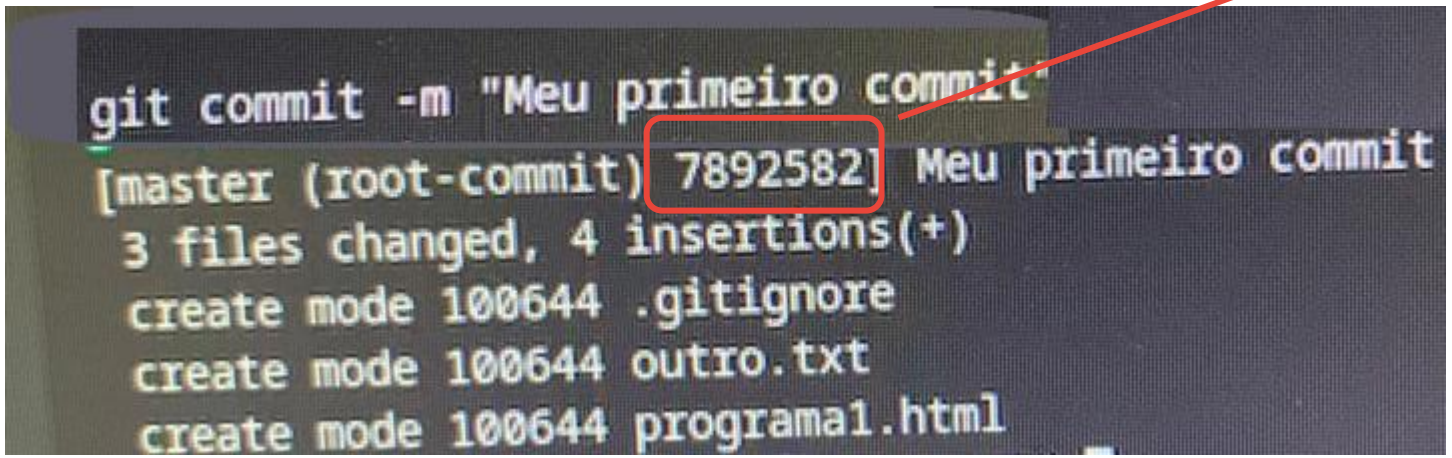


COMANDOS BÁSICOS
GIT COMMIT

Para cada commit (envio) realizado o GIT vai realizar o empacotamento da situação dos arquivos, criando uma fotografia desse momento.

git commit -m “Descrição do que está sendo feito”

Exemplo:



COMANDOS BÁSICOS
GIT LOG

Consulta todo o histórico de commits realizados ao longo do tempo naquele repositório.

git log

Exemplo:

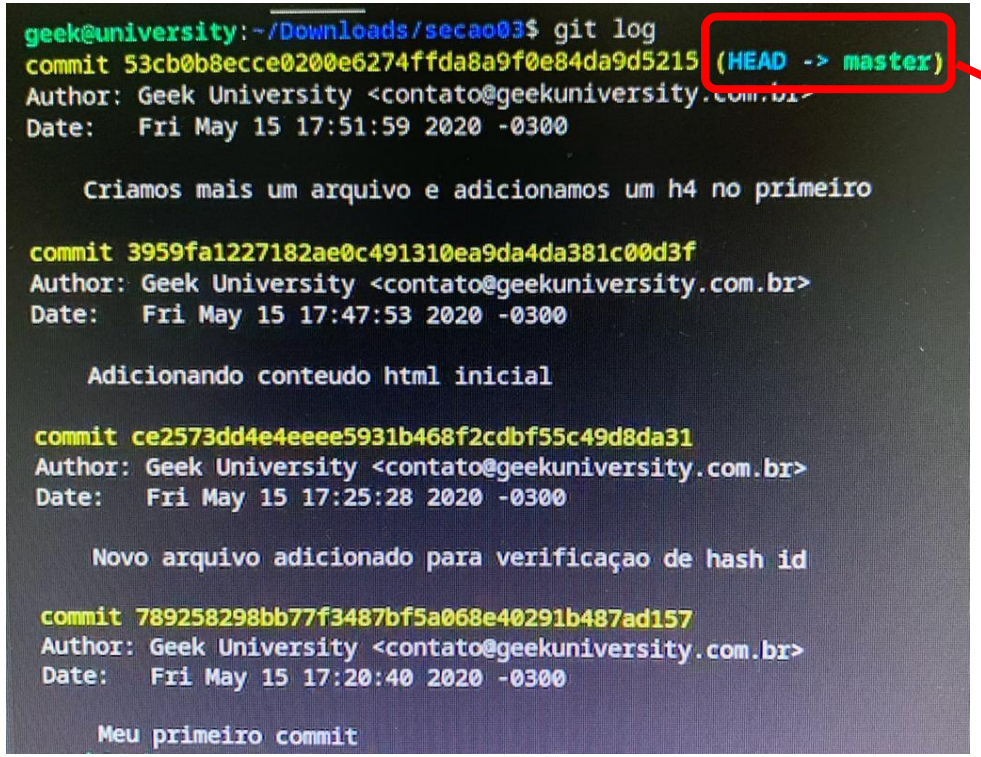
```
geek@university:~/Downloads/secao03$ git log
commit 53cb0b8ecce0200e6274ffda8a9f0e84da9d5215 (HEAD -> master)
Author: Geek University <contato@geekuniversity.com.br>
Date:   Fri May 15 17:51:59 2020 -0300

    Criamos mais um arquivo e adicionamos um h4 no primeiro

commit 3959fa1227182ae0c491310ea9da4da381c00d3f
Author: Geek University <contato@geekuniversity.com.br>
Date:   Fri May 15 17:47:53 2020 -0300

    Adicionando conteudo html inicial

commit ce2573dd4e4eeee5931b468f2cdbf55c49d8da31
Author: Geek University <contato@geekuniversity.com.br>
Date:   Fri May 15 17:25:28 2020 -0300

    Novo arquivo adicionado para verificaçao de hash id

commit 789258298bb77f3487bf5a068e40291b487ad157
Author: Geek University <contato@geekuniversity.com.br>
Date:   Fri May 15 17:20:40 2020 -0300

    Meu primeiro commit
```

COMANDOS BÁSICOS
GIT LOG

Consulta todo o histórico de commits realizados ao longo do tempo naquele repositório.

git log

Exemplo:



Identifica onde estamos.
O último commit...

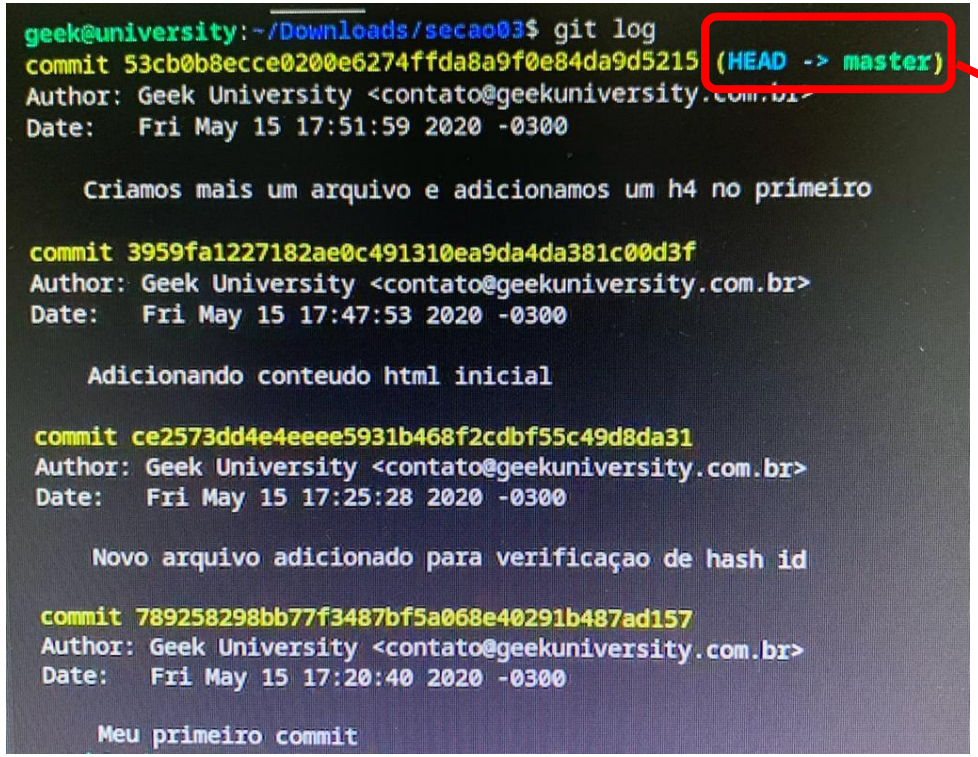
Portanto, a ordem é de baixo para cima
Sendo o último (mais abaixo) o primeiro commit realizado.

COMANDOS BÁSICOS
GIT LOG

Consulta todo o histórico de commits realizados ao longo do tempo naquele repositório.

git log

Exemplo:



Identifica onde estamos.
O último commit...

Portanto, a ordem é de baixo para cima
Sendo o último (mais abaixo) o primeiro commit realizado.

COMANDOS BÁSICOS
GIT LOG

Opções:

git log -2 → apresenta apenas os dois últimos commits

git log --online → apresenta os commits de forma resumida

git log --before="2023-01-15" → todos os commits anteriores a essa data

git log --after="2023-01-15" → todos os commits depois/após dessa data

git log --since="3 days ago" → todos os commits que ocorreram até 3 dias atras.

git log –author="Piva"

COMANDOS BÁSICOS
GIT LOG

Para saber mais...

git help log

COMANDOS BÁSICOS
GIT CHECKOUT

Para deslocar a nossa cabeça (HEAD) para qualquer um dos commits (fotografias temporais), basta passar o número do hash id.

git checkout 7892582

Note que ele muda completamente o conteúdo do projeto (e de seus arquivos) e nos informa que o *HEAD detached* (desencaixada).

Para voltarmos ao HEAD novamente (para o commit principal), basta:

git checkout master

COMANDOS BÁSICOS
GIT MV ARQ.X ARQ2.Y

Podemos renomear um arquivo utilizando o próprio git, para não gerar a necessidade de adição desse novo arquivo ao stage. Isso permite que o tracking desse arquivo seja mais leve.

git mv arquivo.txt teste.txt

Note que se você fizer essa operação pelo terminal, o que ocorrerá é que o arquivo.txt aparecerá como DELETADO e o texte.txt como um NOVO ARQUIVO.

Idem para remoção de arquivos:
git rm arquivo.txt

COMANDOS BÁSICOS
GIT DIFF

Verifica quais as diferenças que existem entre os arquivos atuais e o último commitado.

git diff --staged

Eu também posso verificar a diferença entre os arquivos atuais e um commit específico, passando o hash id dele.

git diff --878a675

Tambem posso consultar a diferença existente entre os arquivos entre um commit e outro:

git diff 878a675..6f5678f (mais velho .. mais novo)

COMANDOS BÁSICOS
GIT COMMIT --AMEND –M “MENSAGEM CORRETA”

Quando você faz um commit e verifica que, por exemplo, errou na escrita da mensagem, você pode corrigir isso, utilizando o atributo --amend

git commit --amend –m “essa mensagem agora está correta”

COMANDOS BÁSICOS
GIT COMMIT --AMEND –M “NOVA MENSAGEM”

O atributo --amend serve para também adicionar um outro arquivo que se esqueceu no último commit ou alterar algo que era pra ter sido feito no último commit.

git commit --amend –m “essa mensagem agora está correta”

COMANDOS BÁSICOS
GIT RESTORE --STAGED <FILE>

Utilizamos esse comando para tirarmos um arquivo de staged (prontos para commit).

git restore --staged arquivo.txt

COMANDOS BÁSICOS
GIT CHECKOUT <FILE>

Ele pega a última versão do <file> commitado e volta a ela (no repositório do projeto).

git checkout arquivo.txt

COMANDOS BÁSICOS
GIT RESET HEAD --HARD

Se você quiser voltar todos os arquivos de seu repositório a última versão commitada, aí você pode fazer o RESET para HEAD, de maneira HARD.

git reset head --hard

COMANDOS BÁSICOS
GIT RESET HEAD^ --HARD

Para descartar o último commit e todas as mudanças que foram feitas no arquivo, voltando todo o repositório no estado que estava no commit anterior ao último....

git reset head^ --hard **(acrescenta-se o circunflexo)**

TAREFAS...

- ...

OBRIGADO

Prof. Dr. Dilermando Piva Jr.

https://piva.pro.br

piva.jr@fatec.sp.gov.br

https://pypro.com.br