

Lógica de Programação



Aplicar Ferramentas para gestão e controle de versão

Data: 14-08-2024

Quando trabalhamos em equipes de desenvolvimento para gerenciamento dos projetos utilizamos o controle de versão que serve para controlar as alterações realizadas e solucionar problemas de conflito em códigos do projeto. São gerados históricos por quem efetuou determinada alteração. Algumas características do controle de versão:

- Comparar um código com uma versão anterior
- Recuperar um código de versão anterior
- Histórico de atualizações
- Facilita o trabalho em equipe

Um controle de versão possui duas partes o repositório e a área de trabalho. O repositório armazena todo o histórico de evolução do projeto, onde são registradas todas as modificações para determinada versão. Os desenvolvedores trabalham com cópias do repositório no computador.

Principais ferramentas utilizadas:

CVS

Um controle de versão mais antigo, utilizado ainda em alguns projetos.

SVN

Conhecido como Subversion, desenvolvido pela Apache Foundation ele é muito utilizado em projeto de softwares corporativos. Possui um histórico de todas as alterações realizadas por um determinado usuário, permitindo que possamos recuperar qualquer versão antiga.

Utiliza os seguintes conceitos no projeto:

Branch - área do projeto que contém determinadas divisões ou etapas do desenvolvimento.

Tags - diretório que contém a versão que deverá ser liberada para o cliente após o branch está finalizado.

Trunk - diretório que contém todas as alterações realizadas diariamente.

Pode ser feito o download no link abaixo:

<https://subversion.apache.org/download.cgi>

Git

Git

Foi desenvolvido por Linux Tolvards em 2005 e tornou-se a ferramenta mais utilizada para controle de versão. No Git todo desenvolvedor tem uma cópia local do projeto. O repositório fica armazenado em um servidor Git como o Github. Quando alguém da equipe vai trabalhar no projeto uma cópia local do repositório é copiada para sua máquina.

Github

O Github é um serviço gratuito que oferece diversas funcionalidades adicionais aplicadas ao git, diversos projetos open source estão hospedados no github, onde os desenvolvedores podem contribuir com correções ou códigos adicionais.

Instalando git

O git é um programa que pode ser instalado através das opções abaixo:

Windows

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

Linux Debian ou Derivados como Mint e Ubuntu

```
sudo apt-get install git
```

Linux Red Hat ou Derivados como o Centos

```
yum install git
```

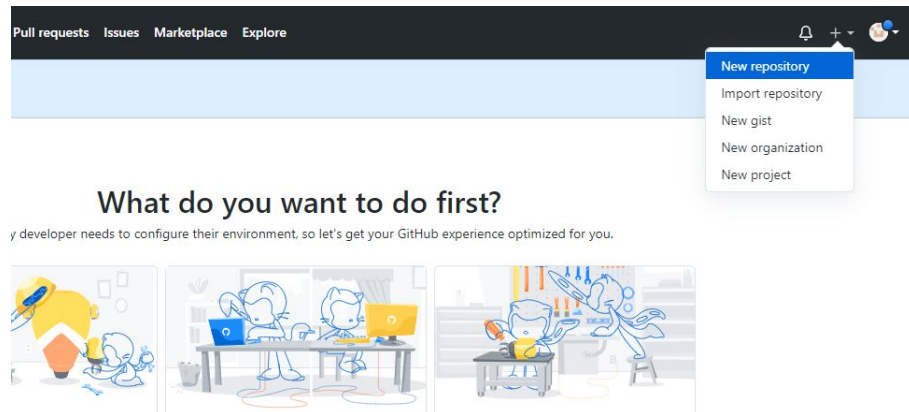
Criação da Conta no GitHub

Acesse <https://github.com/>



- ❑ crie sua conta
- ❑ cadastre seu login de acesso.

Criação do repositório

Para criação de um novo repositório clique no botão conforme imagem abaixo.




Owner * Repository name *


 roni-info / projeto-exemplo 

Great repository names are short and memorable. Need inspiration? How about [ideal-octo-couscous](#)?

Description (optional)

Projeto de testes com o git

☒  **Public**
Anyone on the internet can see this repository. You choose who can commit.

☐  **Private**
You choose who can see and commit to this repository.


Initialize this repository with:

Skip this step if you're importing an existing repository.

☒ **Add a README file**
This is where you can write a long description for your project. [Learn more.](#)

☐ **Add .gitignore**
Choose which files not to track from a list of templates. [Learn more.](#)

☐ **Choose a license**
A license tells others what they can and can't do with your code. [Learn more.](#)

This will set  **main** as the default branch. Change the default name in your [settings](#).

[Create repository](#)

Public - deixa o projeto disponível para qualquer usuário realizar alterações.

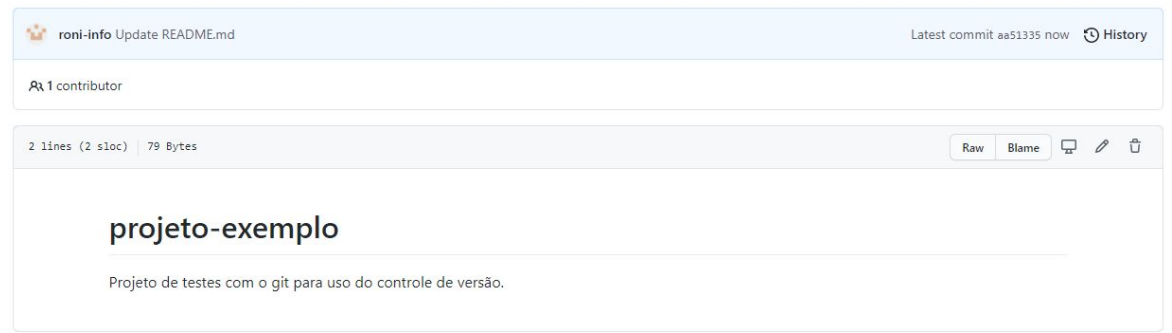
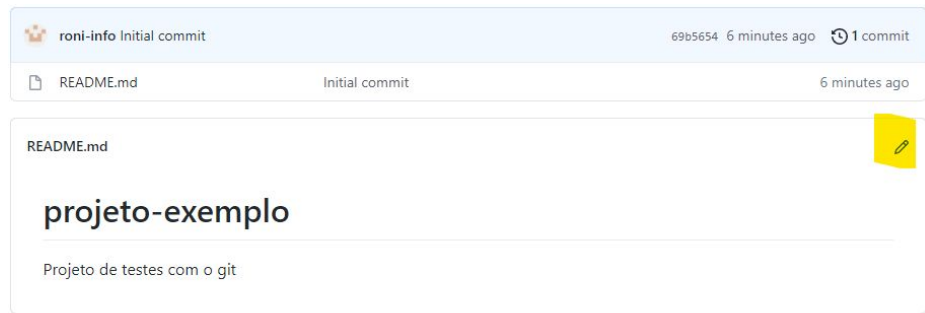
README file - permite uma descrição inicial com detalhes sobre o projeto.

.gitignore - Arquivos que não são importantes são ignorados no envio do projeto.

License - Podemos escolher um tipo de licença.

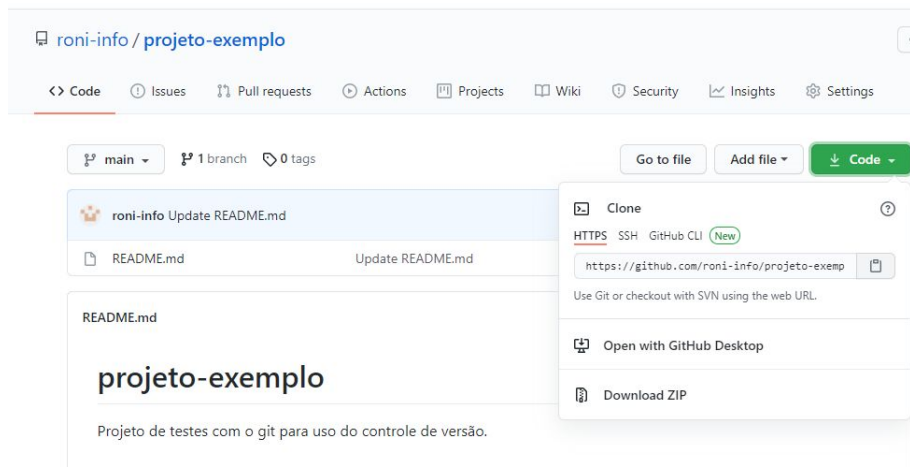
Alterando o README file

Vamos clicar no botão editar e adicionar o texto e gravar as alterações no botão **Commit**.



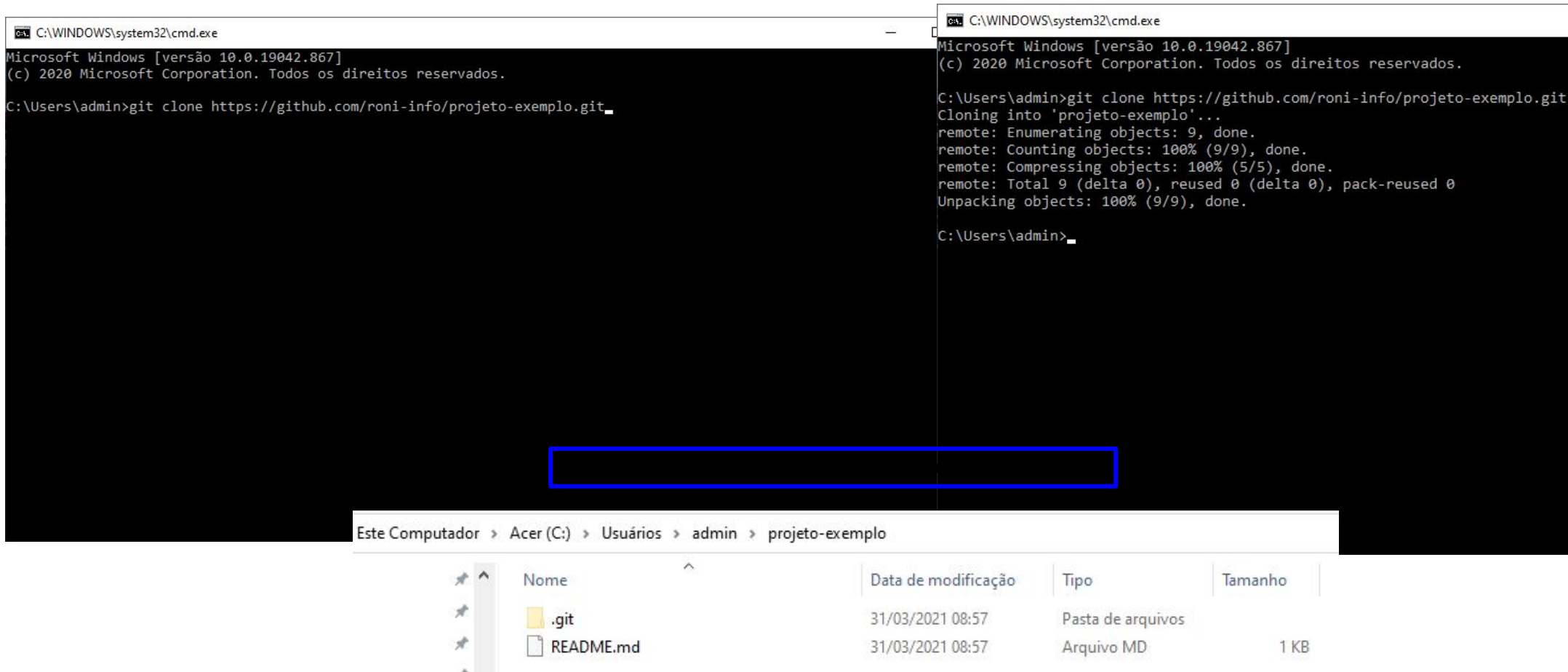
Clonar Repositório

Na página inicial do repositório temos o botão **Code** onde é possível fazer o download do arquivo zip ou copiar a url do repositório. Clique no botão para copiar a url.



Clonando Repositório

Abrir o prompt de comando do Windows digitar **git clone** e colar a url do nosso projeto. Nossa máquina terá uma cópia do repositório remoto. Todos os commits que forem feitos não afetaram o repositório remoto, mas temos comandos também que enviam as alterações para o repositório remoto quando for preciso.



```
C:\WINDOWS\system32\cmd.exe
Microsoft Windows [versão 10.0.19042.867]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.

C:\Users\admin>git clone https://github.com/roni-info/projeto-exemplo.git_

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [versão 10.0.19042.867]
(c) 2020 Microsoft Corporation. Todos os direitos reservados.

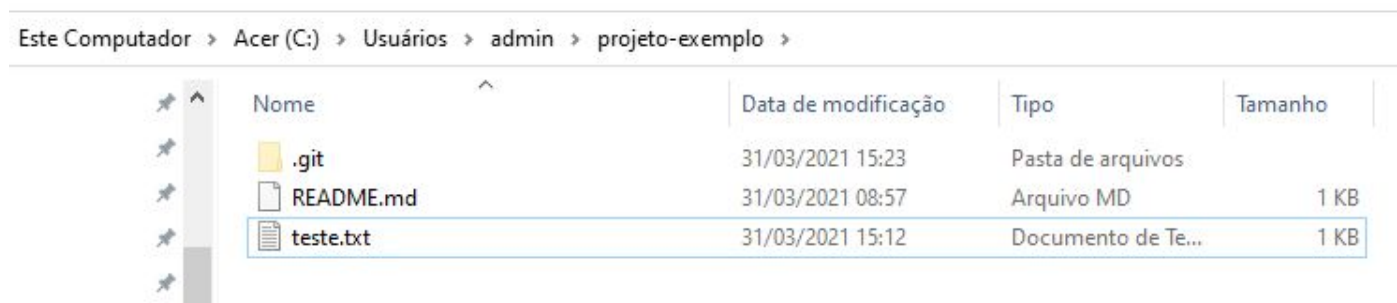
C:\Users\admin>git clone https://github.com/roni-info/projeto-exemplo.git
Cloning into 'projeto-exemplo'...
remote: Enumerating objects: 9, done.
remote: Counting objects: 100% (9/9), done.
remote: Compressing objects: 100% (5/5), done.
remote: Total 9 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (9/9), done.

C:\Users\admin>
```

Este Computador > Acer (C:) > Usuários > admin > projeto-exemplo

Nome	Data de modificação	Tipo	Tamanho
.git	31/03/2021 08:57	Pasta de arquivos	
README.md	31/03/2021 08:57	Arquivo MD	1 KB

Vamos criar um arquivo texto Teste de Commit.txt no nosso repositório local.



Git Status

Exibe o status do repositório, se existe algum arquivo novo adicionado, alterado ou removido, caso esteja rastreado com uma branch no servidor o comando verifica o estado da versão. No exemplo abaixo o comando retornou que um arquivo não está sendo rastreado e para colocar o arquivo para fazer parte do nosso repositório utilizamos o comando **git add**

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git status
On branch main
Your branch is up to date with 'origin/main'.

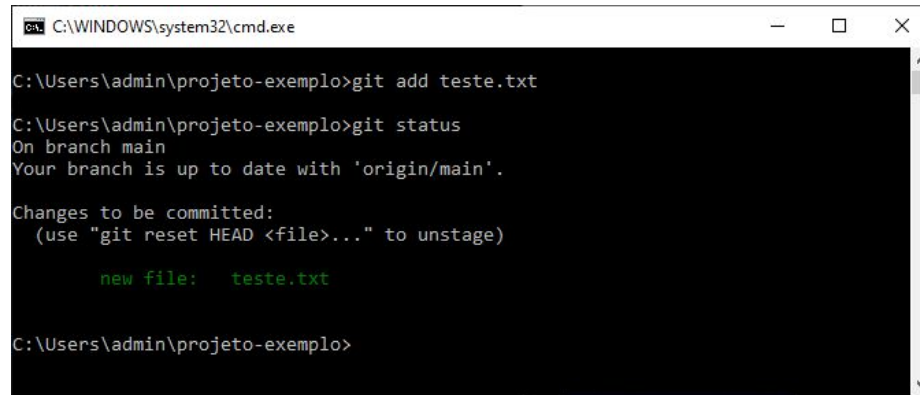
Untracked files:
  (use "git add <file>..." to include in what will be committed)

    teste.txt

nothing added to commit but untracked files present (use "git add" to track)
C:\Users\admin\projeto-exemplo>
```


git add <nome do arquivo>

Adiciona um arquivo alterado e deixa ele preparado para o commit indicando ao git que este arquivo será rastreado. No exemplo abaixo ao executar o comando **git add** e após o **git status** temos o arquivo registrado no git e pronto para o **commit**.



```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git add teste.txt

C:\Users\admin\projeto-exemplo>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

    new file:   teste.txt

C:\Users\admin\projeto-exemplo>
```

git add --all

Quando temos vários arquivos podemos usar o git add com a opção **--all** para registros de vários arquivos.

git commit

Salva as alterações no repositório local e adiciona uma mensagem obrigatória na realização do commit. Quando o commit é realizado as alterações são efetuadas somente no nosso repositório local e não reflete no github.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git commit -m "Commit de Teste"
[main a69dac8] Commit de Teste
 1 file changed, 1 insertion(+)
 create mode 100644 teste.txt

C:\Users\admin\projeto-exemplo>
```

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean

C:\Users\admin\projeto-exemplo>
```

git log

Exibe o histórico de alterações. Utilizando a opção **--stat** são exibidos quais arquivos foram alterados. Os linhas em vermelho abaixo são os commits no repositório remoto e o azul no repositório local.

```
C:\WINDOWS\system32\cmd.exe - git log

C:\Users\admin\projeto-exemplo>git log
commit a69dac8e7750b4bc8ce96479c2e86a7b23b2a664 (HEAD -> main)
Author: Roni Info <roni_info@yahoo.com.br>
Date:   Wed Mar 31 15:56:48 2021 -0300

    Commit de Teste

commit aa51335257fe6f6b39f778deedaeebb4b0dc5cdf (origin/main, origin/HEAD)
Author: roni-info <81621952+roni-info@users.noreply.github.com>
Date:   Wed Mar 31 08:31:38 2021 -0300

    Update README.md

commit 67d11fcd4c50a54cd86d8f85da52ffb091058a4a
Author: roni-info <81621952+roni-info@users.noreply.github.com>
Date:   Wed Mar 31 08:30:52 2021 -0300
```

```
C:\WINDOWS\system32\cmd.exe - git log --stat

C:\Users\admin\projeto-exemplo>git log --stat
commit a69dac8e7750b4bc8ce96479c2e86a7b23b2a664 (HEAD -> main)
Author: Roni Info <roni_info@yahoo.com.br>
Date:   Wed Mar 31 15:56:48 2021 -0300

    Commit de Teste

    teste.txt | 1 +
    1 file changed, 1 insertion(+)

commit aa51335257fe6f6b39f778deedaeebb4b0dc5cdf (origin/main, origin/HEAD)
Author: roni-info <81621952+roni-info@users.noreply.github.com>
Date:   Wed Mar 31 08:31:38 2021 -0300

    Update README.md

    README.md | 3 +--
    1 file changed, 1 insertion(+), 2 deletions(-)

commit 67d11fcd4c50a54cd86d8f85da52ffb091058a4a
Author: roni-info <81621952+roni-info@users.noreply.github.com>
Date:   Wed Mar 31 08:30:52 2021 -0300
```

git log --oneline

Exibe os commits em apenas uma linha

git reflog - outra forma de exibir

git log -p - mostra os arquivo e conteúdo do foi alterado, bem parecido com o **git show**

git log --graph - são exibidas linhas específicas representando o desenvolvimento.

git remote

Retorna informações sobre o nosso repositório remoto.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git remote -v
origin  https://github.com/roni-info/projeto-exemplo.git (fetch)
origin  https://github.com/roni-info/projeto-exemplo.git (push)

C:\Users\admin\projeto-exemplo>
```

git push

É utilizado para enviar todos os commits locais para o github.

```
C:\WINDOWS\system32\cmd.exe - git push origin main

C:\Users\admin\projeto-exemplo>git push origin main
```

Inserir usuário e senha do github

```
C:\WINDOWS\system32\cmd.exe

Username for 'https://github.com': roni-info
Password for 'https://roni-info@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 295 bytes | 24.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
To https://github.com/roni-info/projeto-exemplo.git
aa51335..a69dac8  main -> main

C:\Users\admin\projeto-exemplo>
```

Com os comando acima foi feito envio da branch local (main) para o servidor pela primeira vez, caso a branch que está sendo enviada não exista no servidor, ela será criada. A partir desse momento, a branch local está configurada para ser rastreada com essa origem no servidor.

Obs: Para versionar outras alterações usamos somente o comando git push.

Utilizando o comando **git log** podemos ver a alteração nos dois repositórios local e remoto que possuem as alterações do commit.

```
C:\WINDOWS\system32\cmd.exe - git log
commit a69dac8e7750b4bc8ce96479c2e86a7b23b2a664 (HEAD -> main, origin/main, origin/HEAD)
Author: Roni Info <roni_info@yahoo.com.br>
Date:   Wed Mar 31 15:56:48 2021 -0300

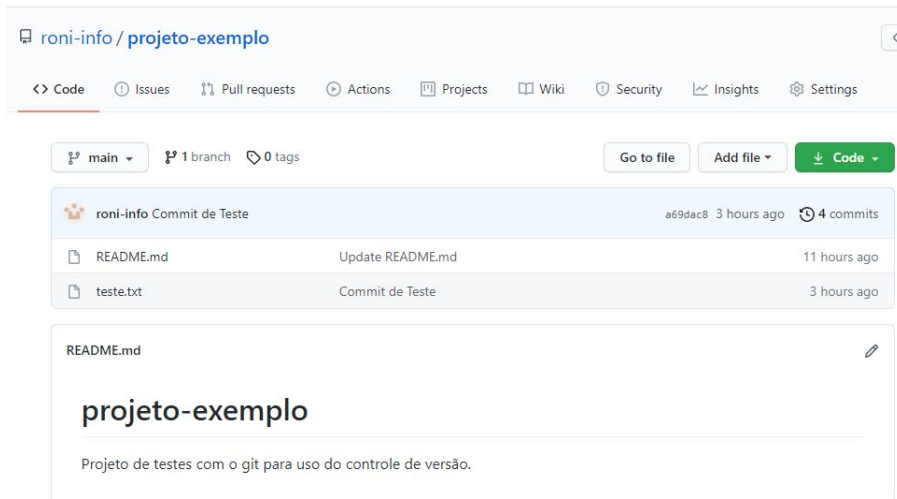
    Commit de Teste

commit aa51335257fe6f6b39f778deedaeebb4b0dc5cdf
Author: roni-info <81621952+roni-info@users.noreply.github.com>
Date:   Wed Mar 31 08:31:38 2021 -0300

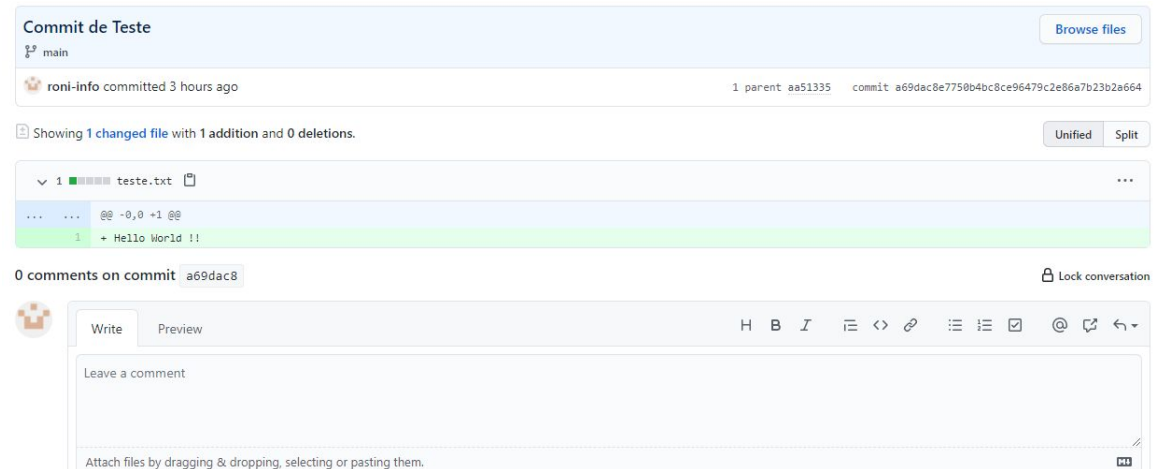
    Update README.md

commit 67d11fcd4c50a54cd86d8f85da52ffb091058a4a
```

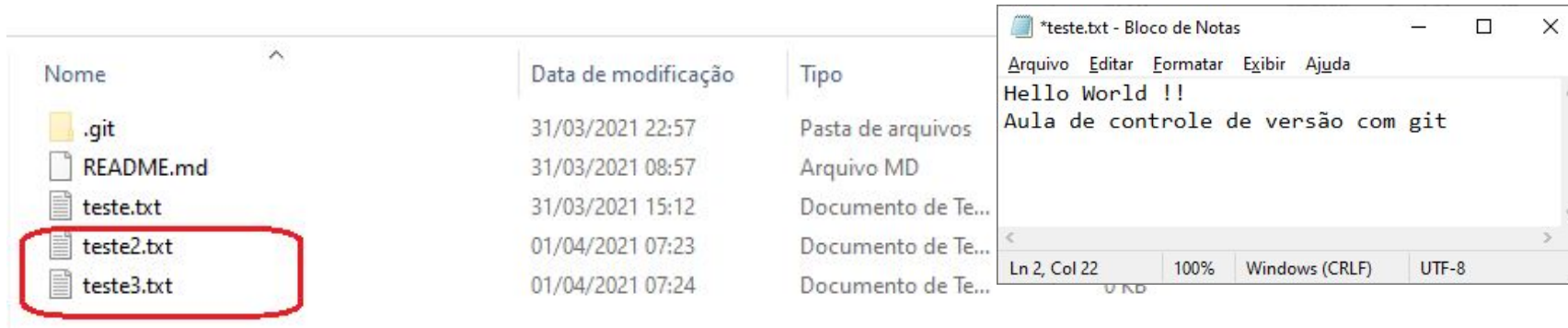
Visualização do commit no github



Informações dos arquivos estão no commit



Criar na mais dois arquivos na pasta do projeto e alterar o conteúdo do arquivo `teste.txt`



Nome	Data de modificação	Tipo
.git	31/03/2021 22:57	Pasta de arquivos
README.md	31/03/2021 08:57	Arquivo MD
teste.txt	31/03/2021 15:12	Documento de Te...
teste2.txt	01/04/2021 07:23	Documento de Te...
teste3.txt	01/04/2021 07:24	Documento de Te...

*teste.txt - Bloco de Notas

Arquivo Editar Formatar Exibir Ajuda

Hello World !!
Aula de controle de versão com git

Ln 2, Col 22 100% Windows (CRLF) UTF-8

Executando o comando `git status` verificamos que um arquivo foi alterado e dois não monitorados foram incluídos

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git status
On branch main
Your branch is up to date with 'origin/main'.

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:   teste.txt

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

        teste2.txt
        teste3.txt

no changes added to commit (use "git add" and/or "git commit -a")
```

Vamos adicionar essas alterações para registro do git com o comando **git add**

O arquivo **teste2.txt** como não foi inserido no comando add ele não vai ser monitorado pelo git, pois pode ser um arquivo temporário para testes e depois pode ser removido. Quando temos vários arquivos ou pastas para serem adicionados podem utilizar o comando **git add .**

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git add teste.txt teste3.txt

C:\Users\admin\projeto-exemplo>git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git reset HEAD <file>..." to unstage)

        modified:   teste.txt
        new file:   teste3.txt

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

        teste2.txt
```

Executando o comando **git commit -m** para salvar as alterações no repositório local

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git commit -m "Realizando commit dos arquivos teste e teste3.txt"
[main 207048d] Realizando commit dos arquivos teste e teste3.txt
 2 files changed, 2 insertions(+), 1 deletion(-)
 create mode 100644 teste3.txt

C:\Users\admin\projeto-exemplo>_
```


Vamos executar o comando **git log** para verificar como ficaram nossos commits realizados.
Verificando o comando abaixo temos um commit para ser enviado para o repositório remoto.

```
C:\WINDOWS\system32\cmd.exe - git log

C:\Users\admin\projeto-exemplo>git log
commit 207048d7579acf7b264300117e31c332d236dae8 (HEAD -> main)
Author: Roni Info <roni_info@yahoo.com.br>
Date: Thu Apr 1 07:54:27 2021 -0300

    Realizando commit dos arquivos teste e teste3.txt

commit a69dac8e7750b4bc8ce96479c2e86a7b23b2a664 (origin/main, origin/HEAD)
Author: Roni Info <roni_info@yahoo.com.br>
:
```

Vamos enviar os commits para o repositório remoto no **GitHub**

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git push origin main
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': roni-info
Password for 'https://roni-info@github.com':
Enumerating objects: 6, done.
Counting objects: 100% (6/6), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 388 bytes | 194.00 KiB/s, done.
Total 4 (delta 0), reused 0 (delta 0)
To https://github.com/roni-info/projeto-exemplo.git
    a69dac8..207048d  main -> main
```

Verificando com o **git log** se os repositórios local e remoto estão apontando para o mesmo commit.

```
C:\WINDOWS\system32\cmd.exe - git log

C:\Users\admin\projeto-exemplo>git log
commit 207048d7579acf7b264300117e31c332d236dae8 (HEAD -> main, origin/main, origin/HEAD)
Author: Roni Info <roni_info@yahoo.com.br>
Date: Thu Apr 1 07:54:27 2021 -0300


    Realizando commit dos arquivos teste e teste3.txt

commit a69dac8e7750b4bc8ce96479c2e86a7b23b2a664
Author: Roni Info <roni_info@yahoo.com.br>
Date: Wed Mar 31 15:56:48 2021 -0300




    Commit de Teste

commit aa51335257fe6f6b39f778deedaeebb4b0dc5cdf
```

Verificando no **GitHub**

 roni-info Realizando commit dos arquivos teste e teste3.txt


207048d 22 minutes ago 5 commits

 README.md	Update README.md	yesterday
 teste.txt	Realizando commit dos arquivos teste e teste3.txt	22 minutes ago
 teste3.txt	Realizando commit dos arquivos teste e teste3.txt	22 minutes ago

README.md

projeto-exemplo

Projeto de testes com o git

 [Readme](#)


Releases


No releases published
[Create a new release](#)

Packages

No packages published
[Publish your first package](#)

Realizando commit dos arquivos teste e teste3.txt



 main

 roni-info committed 25 minutes ago

1 parent a69dac8 commit 207048d7579acf7b264300117e31c332d236dae8

Showing 2 changed files with 2 additions and 1 deletion.

Unified Split



3  teste.txt 

@@ -1 +1,2 @@

1 - Hello World !!

1 + Hello World !!

2 + Aula de controle de versão com git

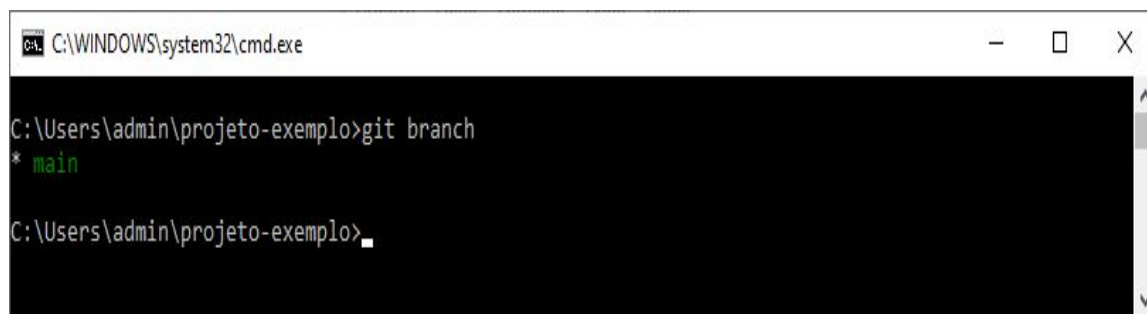
0  teste3.txt 

Empty file.

São divisões dentro do repositório. É utilizado em situações específicas como por exemplo, quando queremos adicionar um novo recurso ou corrigir um erro para garantir que o código instável não seja modificado. No nosso exemplo estamos trabalhando no repositório padrão o **main** ou **master**.

git branch

O comando **git branch** exibe as branches do repositório.

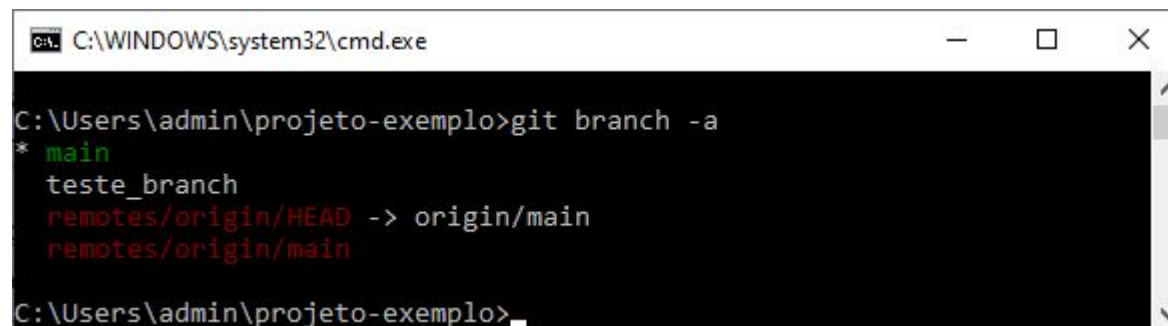


```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git branch
* main

C:\Users\admin\projeto-exemplo>_
```

git branch -a exibe as branches locais e remotas



```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git branch -a
* main
  teste_branch
  remotes/origin/HEAD -> origin/main
  remotes/origin/main

C:\Users\admin\projeto-exemplo>_
```

Vamos criar uma nova branch utilizando comando **git branch <nome da branch>**

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git branch teste_branch

C:\Users\admin\projeto-exemplo>git checkout teste_branch
Switched to branch 'teste_branch'

C:\Users\admin\projeto-exemplo>git branch
main
* teste_branch

C:\Users\admin\projeto-exemplo>
```

Para acessar a branch criada utilizamos o comando **git checkout teste_branch** a branch destacada com * é a que estamos acessando.

Quando criamos uma branch ele vai ser uma cópia do último commit da branch atual que estavamos, no nosso exemplo a branch principal. Para deletar uma branch podemos usar a opção -d ou -D. A opção -D apaga diretamente e o -d avisa que não foi feito o merge da nossa branch.

No exemplo abaixo foi criado o arquivo **teste4.txt**, adicionamos todos os arquivos com **git add .** e fizemos o commit. O arquivo **teste2.txt** que não havia sido registrado na branch principal foi registrado na **teste_branch**

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>echo "Hello World !!" > teste4.txt

C:\Users\admin\projeto-exemplo>git add .

C:\Users\admin\projeto-exemplo>git commit -m "Criação arquivo teste4.txt"
[teste_branch a026203] Criação arquivo teste4.txt
2 files changed, 1 insertion(+)
 create mode 100644 teste2.txt
 create mode 100644 teste4.txt

C:\Users\admin\projeto-exemplo>
```

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git log
commit a026203e9dfff5f5c611760700d277427e9ec2f38 (HEAD -> teste_branch)
Author: Roni Info <roni_info@yahoo.com.br>
Date:   Fri Apr 2 08:16:42 2021 -0300

    Cria<C3><A7><C3><A3>o arquivo teste4.txt

commit 207048d7579acf7b264300117e31c332d236dae8 (origin/main, origin/HEAD,
Author: Roni Info <roni_info@yahoo.com.br>
Date:   Thu Apr 1 07:54:27 2021 -0300

    Realizando commit dos arquivos teste e teste3.txt
```

Verificando as diferenças entre a branch principal (**main**) e a branch **teste_branch**

```
C:\WINDOWS\system32\cmd.exe
C:\Users\admin\projeto-exemplo>git branch
  main
* teste_branch

C:\Users\admin\projeto-exemplo>dir
O volume na unidade C é Acer
O Número de Série do Volume é 1C3E-71F2

Pasta de C:\Users\admin\projeto-exemplo

02/04/2021  08:31    <DIR>          .
02/04/2021  08:31    <DIR>          ..
31/03/2021  08:57                81 README.md
01/04/2021  07:31                51 teste.txt
02/04/2021  08:31                 0 teste2.txt
01/04/2021  07:24                 0 teste3.txt
02/04/2021  08:31                19 teste4.txt
                    5 arquivo(s)          151 bytes
                    2 pasta(s)  737.806.233.600 bytes disponíveis
```

```
C:\WINDOWS\system32\cmd.exe
C:\Users\admin\projeto-exemplo>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

C:\Users\admin\projeto-exemplo>dir
O volume na unidade C é Acer
O Número de Série do Volume é 1C3E-71F2

Pasta de C:\Users\admin\projeto-exemplo

02/04/2021  08:37    <DIR>          .
02/04/2021  08:37    <DIR>          ..
31/03/2021  08:57                81 README.md
01/04/2021  07:31                51 teste.txt
01/04/2021  07:24                 0 teste3.txt
                    3 arquivo(s)          132 bytes
                    2 pasta(s)  737.805.533.184 bytes disponíveis

C:\Users\admin\projeto-exemplo>
```

git merge <nome_da_branch>

Mescla as modificações presentes de uma branch criada para uma branch selecionada. No exemplo abaixo alternamos para a branch **main** com o comando checkout e fizemos o merge com da **teste_branch**.

```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.


C:\Users\admin\projeto-exemplo>git merge teste_branch
Updating 207048d..a026203
Fast-forward
 teste2.txt | 0
 teste4.txt | 1 +
 2 files changed, 1 insertion(+)
 create mode 100644 teste2.txt
 create mode 100644 teste4.txt
C:\Users\admin\projeto-exemplo>
```

Enviando as modificações da branch main para o GitHub

```
C:\WINDOWS\system32\cmd.exe

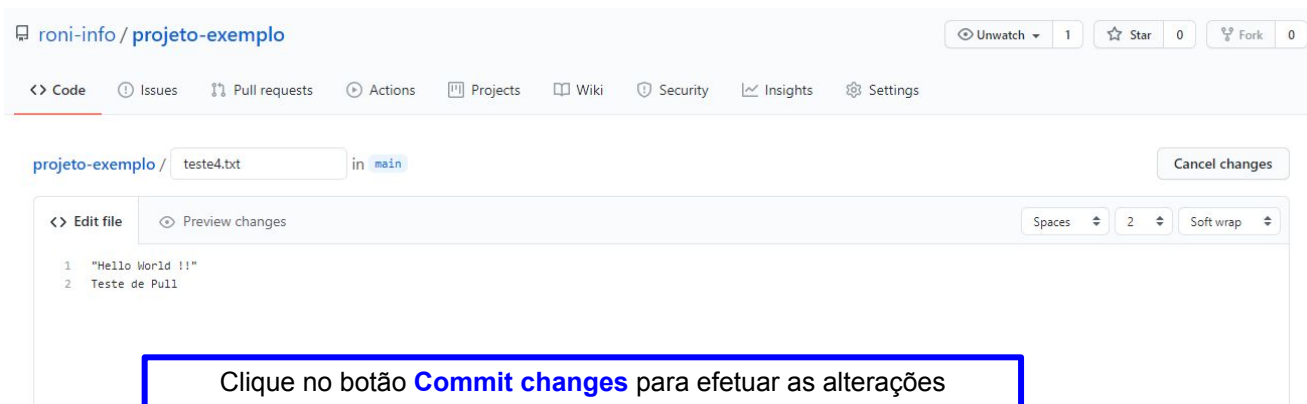
C:\Users\admin\projeto-exemplo>git push
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': roni-info
Password for 'https://roni-info@github.com':
Enumerating objects: 4, done.
Counting objects: 100% (4/4), done.
Delta compression using up to 4 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 377 bytes | 377.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0)
```

Quando não temos conflito de código entre as branches a propriedade Fast-Forward realiza o mesmo commit feito pela branch_teste para fazer o merge com a branch principal.

roni-info Criação arquivo teste4.txt		a026203 1 hour ago 6 commits
README.md	Update README.md	2 days ago
teste.txt	Realizando commit dos arquivos teste e teste3.txt	yesterday
teste2.txt	Criação arquivo teste4.txt	1 hour ago
teste3.txt	Realizando commit dos arquivos teste e teste3.txt	yesterday
teste4.txt	Criação arquivo teste4.txt	1 hour ago
README.md		

Quando trabalhamos em equipe precisamos sincronizar o que está no repositório remoto com o nosso repositório local para fazer isto utilizamos o comando **git pull**. O pull é uma junção dos comandos fetch(buscar) e merge (fundir).

Edite o arquivo **teste4.txt** no GitHub e acrescente um a linha com texto qualquer



roni-info / projeto-exemplo

<> Code Issues Pull requests Actions Projects Wiki Security Insights Settings

projeto-exemplo / teste4.txt in main

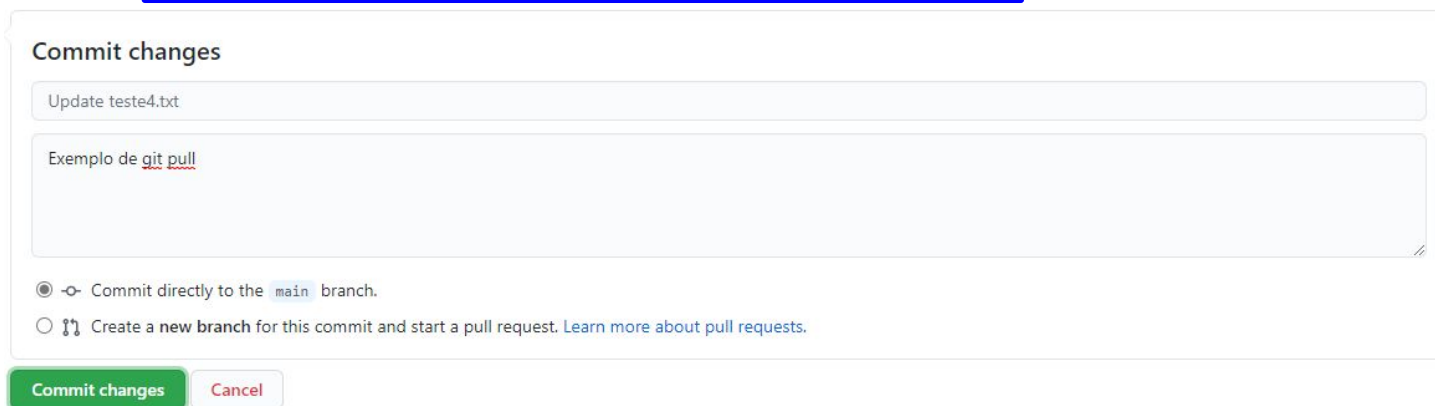
Cancel changes

<> Edit file Preview changes

Spaces 2 Soft wrap

```
1 "Hello World !!"  
2 Teste de Pull
```

Clique no botão **Commit changes** para efetuar as alterações



Commit changes

Update teste4.txt

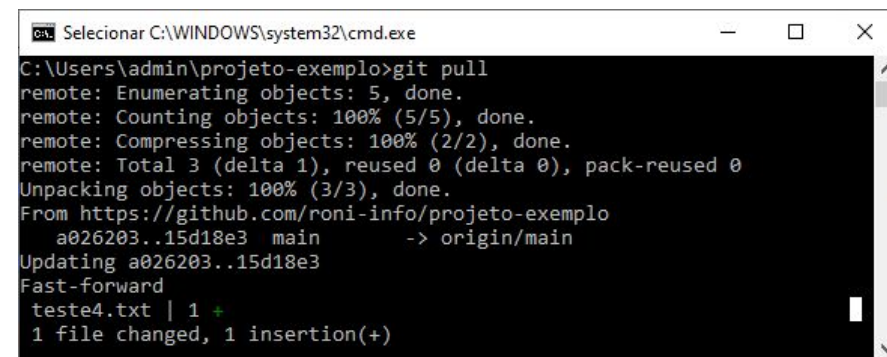
Exemplo de [git pull](#)

☒ Commit directly to the **main** branch.

☐ Create a new branch for this commit and start a pull request. [Learn more about pull requests.](#)

Commit changes Cancel

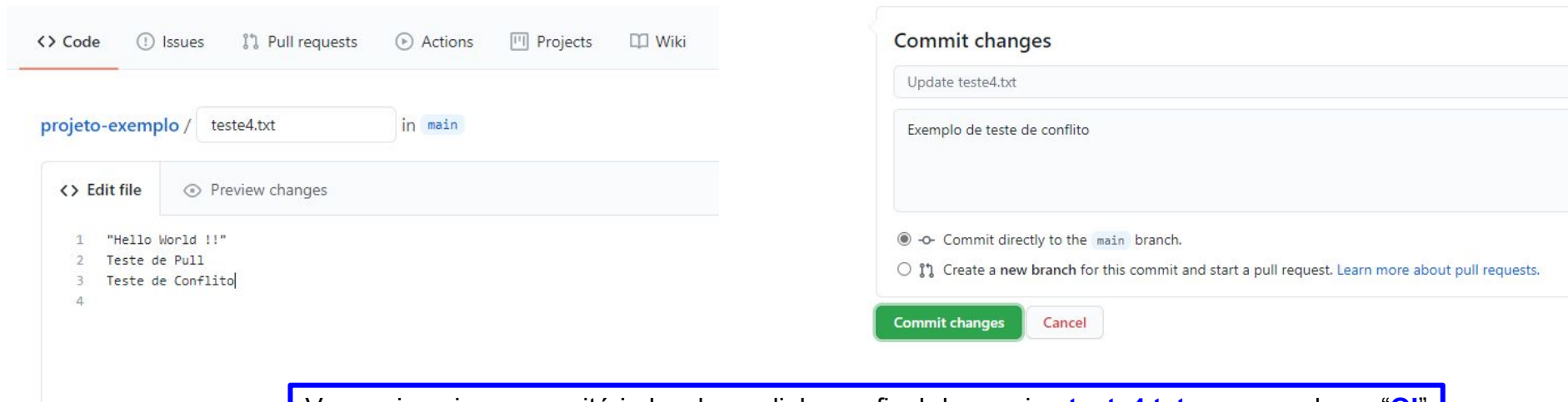
Execute o git pull para efetuar a alteração para o repositório local



```
C:\Users\admin\projeto-exemplo>git pull  
remote: Enumerating objects: 5, done.  
remote: Counting objects: 100% (5/5), done.  
remote: Compressing objects: 100% (2/2), done.  
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0  
Unpacking objects: 100% (3/3), done.  
From https://github.com/roni-info/projeto-exemplo  
a026203..15d18e3 main -> origin/main  
Updating a026203..15d18e3  
Fast-forward  
 teste4.txt | 1 +  
 1 file changed, 1 insertion(+)
```


Resolvendo conflitos

Vamos inserir uma linha com o texto abaixo no arquivo `teste4.txt` inserir o comentário e clicar em **Commit changes**

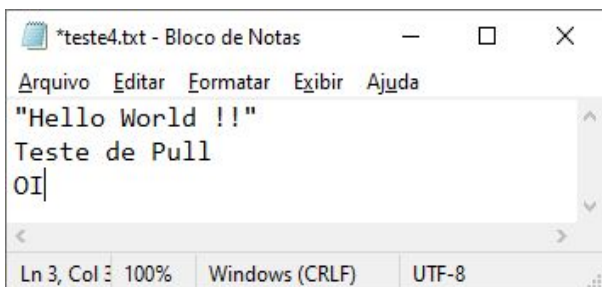


The screenshot shows the GitHub web interface. On the left, the file editor for `teste4.txt` in the `main` branch is open. The file content is:

```
1 "Hello World !!"
2 Teste de Pull
3 Teste de Conflito
4
```

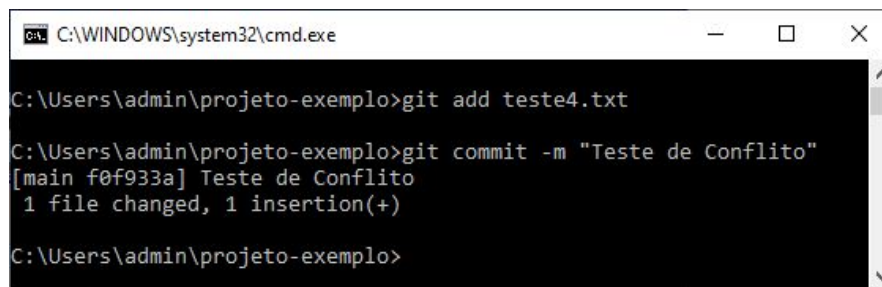
On the right, the "Commit changes" dialog is open. The commit message is "Exemplo de teste de conflito". The "Commit directly to the main branch" option is selected. The "Commit changes" button is highlighted in green.

Vamos inserir no repositório local uma linha no final do arquivo `teste4.txt` com o conteúdo "OI" fazer o commit e depois ao executar o `git push` para enviar para como resposta um erro por causa do conflito gerado.



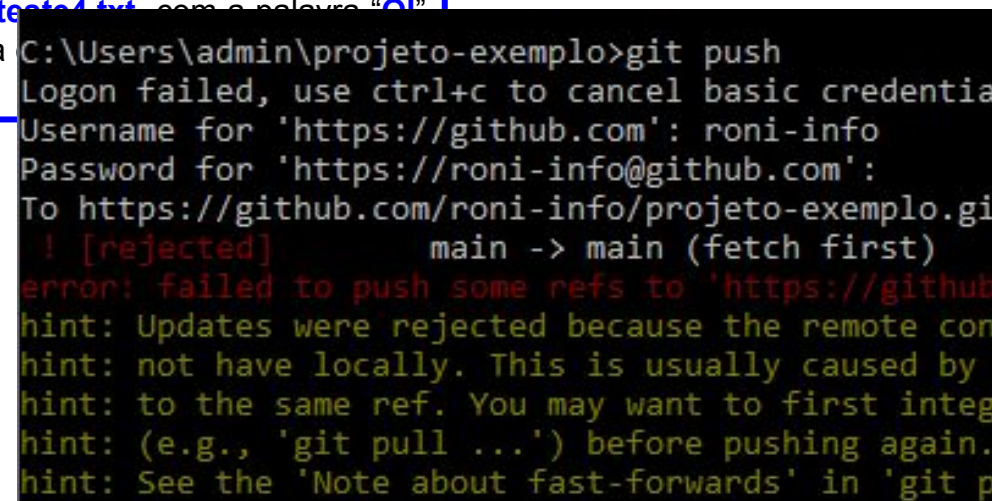
The screenshot shows a Notepad window titled `*teste4.txt - Bloco de Notas`. The content of the file is:

```
"Hello World !!"
Teste de Pull
OI
```



```
C:\WINDOWS\system32\cmd.exe

C:\Users\admin\projeto-exemplo>git add teste4.txt
C:\Users\admin\projeto-exemplo>git commit -m "Teste de Conflito"
[main f0f933a] Teste de Conflito
1 file changed, 1 insertion(+)
C:\Users\admin\projeto-exemplo>
```



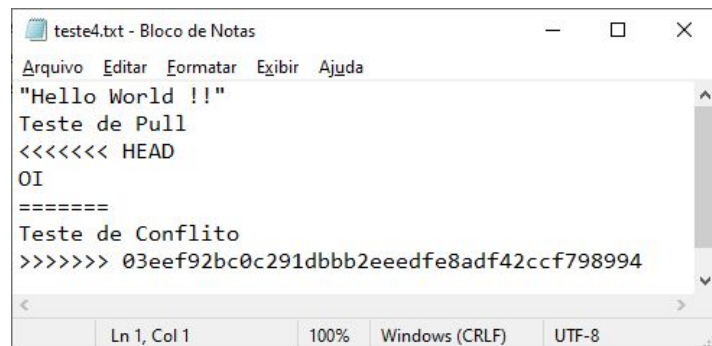
```
C:\Users\admin\projeto-exemplo>git push
Logon failed, use ctrl+c to cancel basic credential
Username for 'https://github.com': roni-info
Password for 'https://roni-info@github.com':
To https://github.com/roni-info/projeto-exemplo.git
 ! [rejected]        main -> main (fetch first)
error: failed to push some refs to 'https://github.com/roni-info/projeto-exemplo.git'
hint: Updates were rejected because the remote contains changes that
hint: you do not have locally. This is usually caused by another commit
hint: to the same ref. You may want to first integrate the remote changes
hint: (e.g., 'git pull ...') before pushing again.
hint: See the 'Note about fast-forwards' in 'git push' man page.
```

O procedimento correto é trazer o que tem no repositório primeiro com o git pull e depois fazer o push.

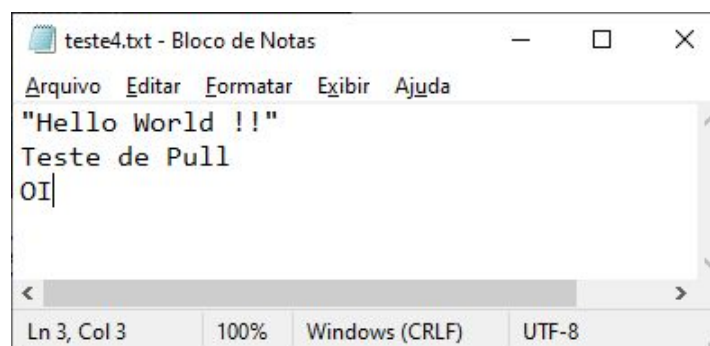
```
C:\Users\admin\projeto-exemplo>git pull
remote: Enumerating objects: 5, done.
remote: Counting objects: 100% (5/5), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 1), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), done.
From https://github.com/roni-info/projeto-exemplo
   068b994..03eef92  main       -> origin/main
Auto-merging teste4.txt
CONFLICT (content): Merge conflict in teste4.txt
Automatic merge failed; fix conflicts and then commit the result.
```

Vamos verificar e resolver os conflitos, fazer o commit e enviar para o repositório remoto

Foi deixada apenas a palavra OI



```
teste4.txt - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
"Hello World !!"
Teste de Pull
<<<<<< HEAD
OI
=====
Teste de Conflito
>>>>>> 03eef92bc0c291dbbb2eedfe8adf42ccf798994
```

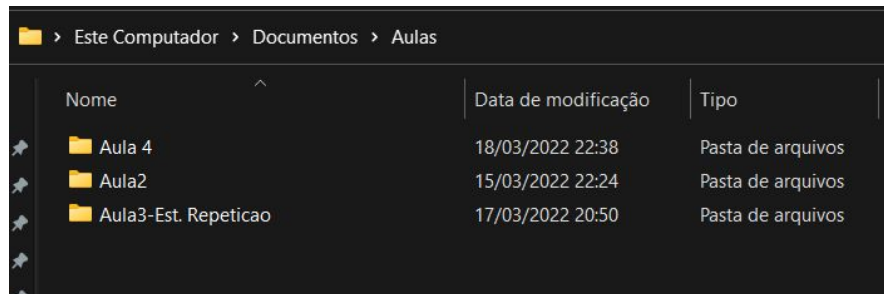


```
teste4.txt - Bloco de Notas
Arquivo Editar Formatar Exibir Ajuda
"Hello World !!"
Teste de Pull
OI
```

Após o push podemos verificar no [GitHub](https://github.com)

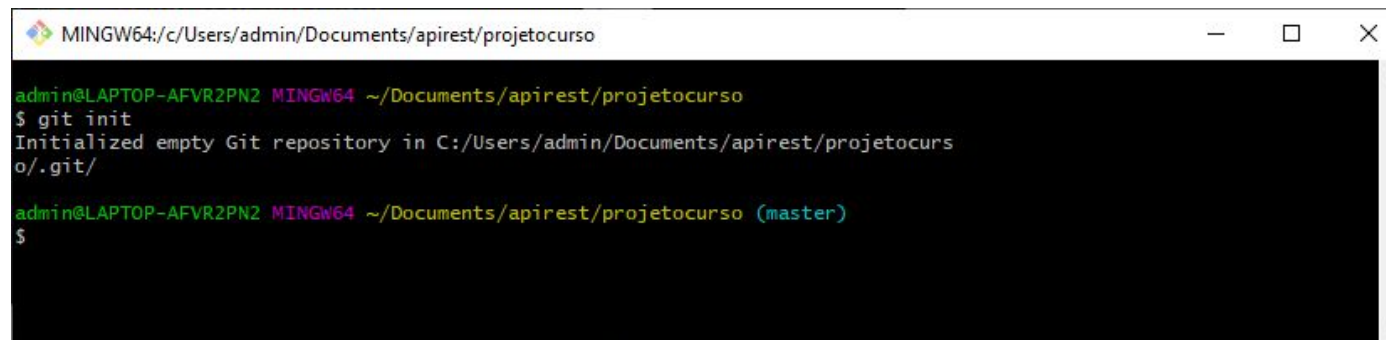
```
C:\Users\admin\projeto-exemplo>git add teste4.txt
C:\Users\admin\projeto-exemplo>git commit -m "Eliminando Conflitos"
[main f421c62] Eliminando Conflitos
C:\Users\admin\projeto-exemplo>git push
Logon failed, use ctrl+c to cancel basic credential prompt.
Username for 'https://github.com': roni-info
Password for 'https://roni-info@github.com':
Enumerating objects: 8, done.
Counting objects: 100% (8/8), done.
Delta compression using up to 4 threads
Compressing objects: 100% (3/3), done.
Writing objects: 100% (4/4), 428 bytes | 428.00 KiB/s, done.
Total 4 (delta 2), reused 0 (delta 0)
remote: Resolving deltas: 100% (2/2), completed with 1 local object.
To https://github.com/roni-info/projeto-exemplo.git
   03eef92..f421c62  main -> main
```

Abrir a pasta onde se encontra o projeto e clicar em **Git Bash Here**



Este Computador > Documentos > Aulas		
Nome	Data de modificação	Tipo
Aula 4	18/03/2022 22:38	Pasta de arquivos
Aula2	15/03/2022 22:24	Pasta de arquivos
Aula3-Est. Repeticao	17/03/2022 20:50	Pasta de arquivos

Criando um repositório de versionamento na pasta



```
MINGW64:/c:/Users/admin/Documents/apirest/projetocurso
admin@LAPTOP-AFVR2PN2 MINGW64 ~/Documents/apirest/projetocurso
$ git init
Initialized empty Git repository in C:/Users/admin/Documents/apirest/projetocurso/.git/
admin@LAPTOP-AFVR2PN2 MINGW64 ~/Documents/apirest/projetocurso (master)
$
```

Será criada uma pasta chamada oculta com o nome **.git** que será usada para o controle do versionamento.

Configurando o usuário e email do GitHub

```
admin@LAPTOP-AFVR2PN2 MINGW64 ~/Documents/apirest/projetocurso (master)
$ git config --global user.name "roni-info"

admin@LAPTOP-AFVR2PN2 MINGW64 ~/Documents/apirest/projetocurso (master)
$ git config --global user.email "roni_info@yahoo.com.br"
```

Listar configurações

```
git config --list
git config user.email
git config user.name
```

Adicionando todos os arquivos para staging área com o comando `git add .`

```
admin@LAPTOP-AFVR2PN2 MINGW64 ~/Documents/apirest/projetocurso (master)
$ git add .
warning: LF will be replaced by CRLF in .gitignore.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .mvn/wrapper/MavenWrapperDownloader.java.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in .mvn/wrapper/maven-wrapper.properties.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in mvnw.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in mvnw.cmd.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in pom.xml.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/main/java/br/com/roni/ProjetocursoApplication.java.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/main/resources/application.properties.
The file will have its original line endings in your working directory
warning: LF will be replaced by CRLF in src/test/java/br/com/roni/ProjetocursoApplicationTests.java.
The file will have its original line endings in your working directory
```

Fazendo o commit

```
admin@LAPTOP-AFVR2PN2 MINGW64 ~/Documents/apirest/projetocurso (master)
$ git status
On branch master

No commits yet

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

        new file:   .gitignore
        new file:   .mvn/wrapper/MavenWrapperDownloader.java
        new file:   .mvn/wrapper/maven-wrapper.jar
        new file:   .mvn/wrapper/maven-wrapper.properties
        new file:   mvnw
        new file:   mvnw.cmd
        new file:   pom.xml
        new file:   src/main/java/br/com/roni/ProjetocursoApplication.java
        new file:   src/main/resources/application.properties
        new file:   src/test/java/br/com/roni/ProjetocursoApplicationTests.java

admin@LAPTOP-AFVR2PN2 MINGW64 ~/Documents/apirest/projetocurso (master)
$ git commit -m "Projeto inicial"
[master (root-commit) 6b3d5c2] Projeto inicial
10 files changed, 712 insertions(+)
create mode 100644 .gitignore
create mode 100644 .mvn/wrapper/MavenWrapperDownloader.java
create mode 100644 .mvn/wrapper/maven-wrapper.jar
create mode 100644 .mvn/wrapper/maven-wrapper.properties
create mode 100644 mvnw
create mode 100644 mvnw.cmd
create mode 100644 pom.xml
create mode 100644 src/main/java/br/com/roni/ProjetocursoApplication.java
create mode 100644 src/main/resources/application.properties
create mode 100644 src/test/java/br/com/roni/ProjetocursoApplicationTests.java

admin@LAPTOP-AFVR2PN2 MINGW64 ~/Documents/apirest/projetocurso (master)
$ git log
commit 6b3d5c202f5de0821da99b3753a73a682dc9f039 (HEAD -> master)
Author: roni-info <roni_info@yahoo.com.br>
Date:   Wed Jun 30 14:24:05 2021 -0300

    Projeto inicial
```

Create a new repository

A repository contains all project files, including the revision history. Already have a project repository elsewhere? [Import a repository.](#)

Owner *



roni-info ▾



Repository name *

projetocurso



Great repository names are short and memorable. Need inspiration? How about [studious-enigma](#)?

Description (optional)



Public

Anyone on the internet can see this repository. You choose who can commit.



Private

You choose who can see and commit to this repository.

Initialize this repository with:

Skip this step if you're importing an existing repository.

☐ Add a README file

This is where you can write a long description for your project. [Learn more.](#)

☐ Add .gitignore

Choose which files not to track from a list of templates. [Learn more.](#)

☐ Choose a license

A license tells others what they can and can't do with your code. [Learn more.](#)

Create repository

Associando projeto da nossa máquina com o repositório criado no github, copiando o comando do github

```
git remote add origin https://github.com/roni-info/projetocurso.git
```

```
admin@LAPTOP-AFVR2PN2 MINGW64 ~/Documents/apirest/projetocurso (main)
$ git remote add origin https://github.com/roni-info/projetocurso.git
```

Fazendo o push

```
admin@LAPTOP-AFVR2PN2 MINGW64 ~/Documents/apirest/projetocurso (main)
$ git push -u origin main
Enumerating objects: 26, done.
Counting objects: 100% (26/26), done.
Delta compression using up to 4 threads
Compressing objects: 100% (17/17), done.
Writing objects: 100% (26/26), 52.36 KiB | 4.76 MiB/s, done.
Total 26 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/roni-info/projetocurso.git
 * [new branch]      main -> main
Branch 'main' set up to track remote branch 'main' from 'origin'.
```

Nos próximo envios usar apenas git push

<https://learngitbranching.js.org/>

<https://onlywei.github.io/explain-git-with-d3/>