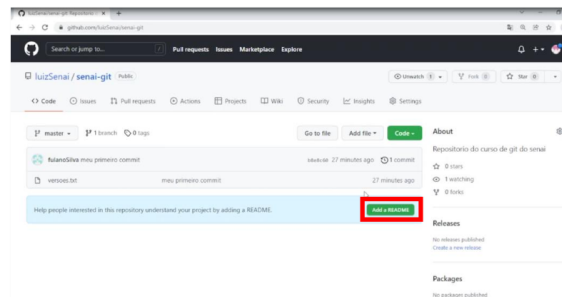


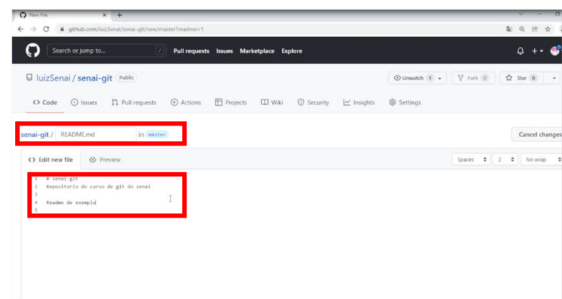
VERSIONAMENTO DE SOFTWARE COM GIT

Controle de alterações

Na pasta do seu projeto no GitHub, clique em Add README. Com isso, você irá fazer uma alteração diretamente no seu repositório local.

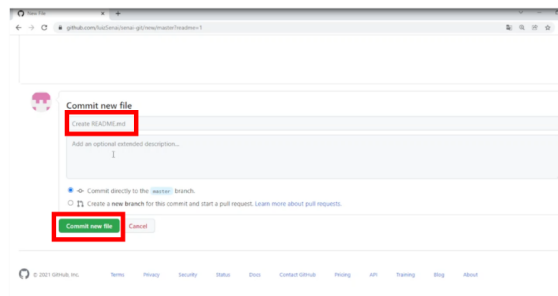


Ao abrir a página, você perceberá que foi criado um arquivo chamado README.md.



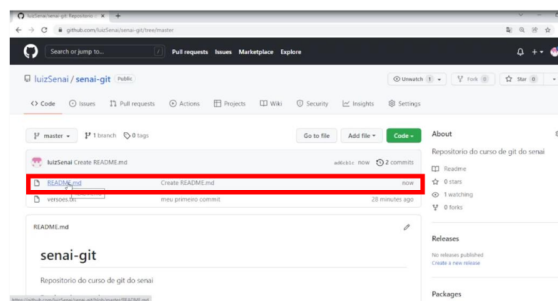
Para testar este arquivo, digitamos a frase "Readme de exemplo".

Indo até o final desta janela, você encontrará a opção para criar um novo *commit*. Para isso, clique em Commit new file. Neste caso, mantivemos o nome original do arquivo, o README.md.



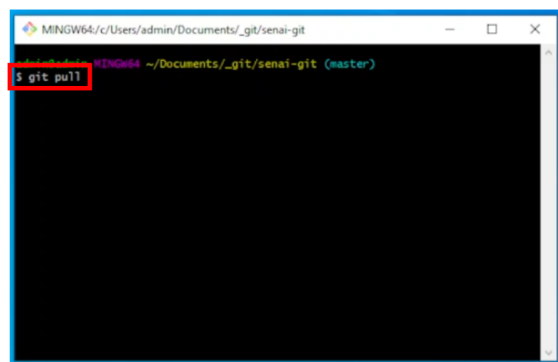
Após criar o novo *commit*, ele passará a aparecer no repositório.

Esta demonstração serve para que você perceba como um arquivo pode ser criado ou mesmo editado no repositório remoto.



Agora, vamos baixar este arquivo para o servidor local.

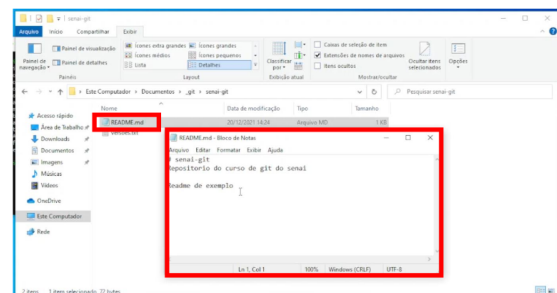
Para tanto, abra o terminal, digite o comando `git pull` e pressione a tecla *enter*.



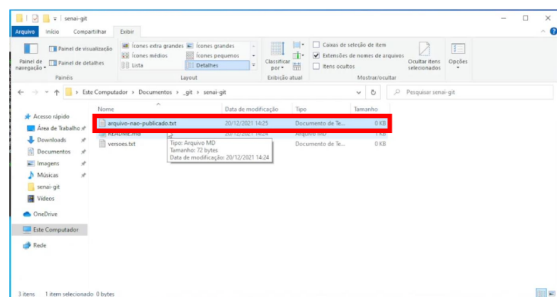
Note que terminal já reconhece a presença do novo arquivo, o README.md.

```
MINGW64/c/Users/admin/Documents/_git/senai-git
admin@admin: ~/Documents/_git/senai-git (master)
$ git pull
remote: Enumerating objects: 4, done.
remote: Counting objects: 100% (4/4), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
Unpacking objects: 100% (3/3), 723 bytes | 89.00 KiB/s, done.
From https://github.com:luizsenai/senai-git
   b8e8c60..ad6cb1c master -> origin/master
Updating b8e8c60..ad6cb1c
create mode 100644 README.md
1 file changed, 4 insertions(+)
```

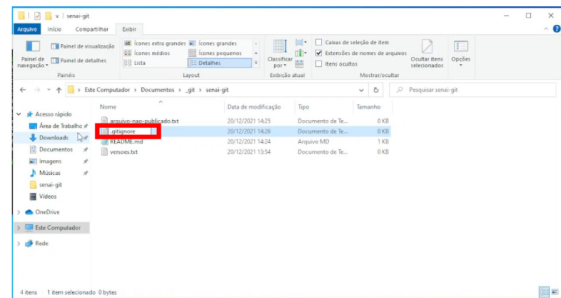
Você também irá encontrar o arquivo no diretório local do projeto. Abra o arquivo com o bloco de notas para verificar o conteúdo, que deve ser idêntico à edição feita no servidor remoto.



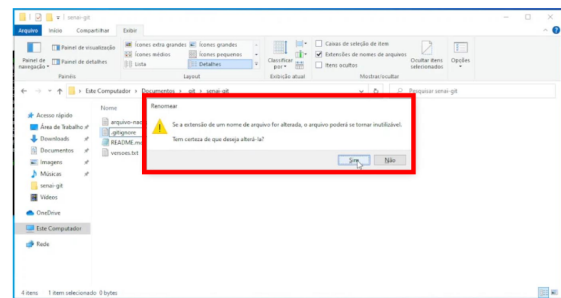
Agora, vamos criar um novo arquivo chamado arquivo-não-publicado.txt. Este novo arquivo será configurado para que não seja monitorado.



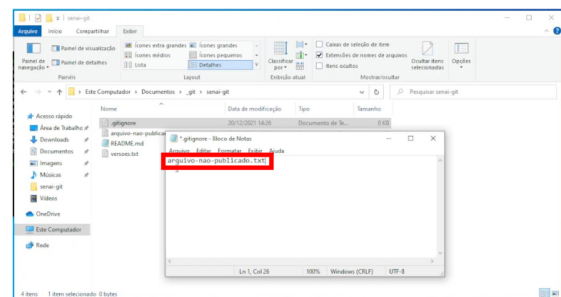
Para impedir este monitoramento de um arquivo, vamos criar um novo arquivo de texto e, em vez de usar a extensão .txt, nomearemos o arquivo como .gitignore.



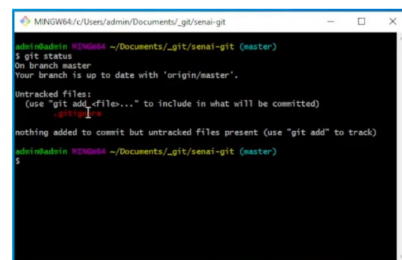
O Windows deve solicitar que você confirme a mudança da extensão do arquivo. Clique em sim.



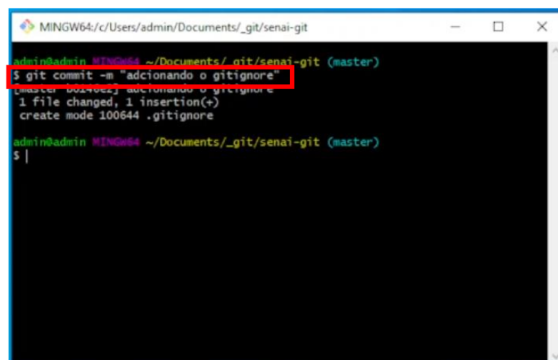
Após colar o nome do arquivo e salvar o .gitignore, os arquivos que estiverem dentro deste arquivo serão ignorados.



Ao conferir os arquivos no terminal, perceba que o arquivo .gitignore já pode ser encontrado enquanto o arquivo-não-publicado não aparecer.

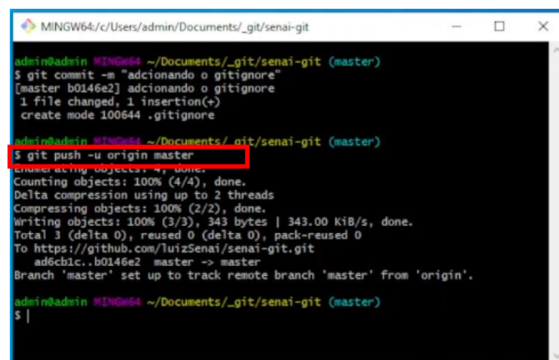


Para adicionar este arquivo .gitignore ao repositório remoto, comece digitando o comando `git add` no terminal para criar o *commit*.



```
admin@admin: ~/Documents/_git/senai-git (master)
$ git commit -m "adicionando o gitignore"
[master b0146e2] adicionando o gitignore
1 file changed, 1 insertion(+)
create mode 100644 .gitignore
admin@admin: ~/Documents/_git/senai-git (master)
$ |
```

Em seguida, use o comando `git push -u origin master` e tecle *enter*. Esta ação deve salvar as alterações no servidor.



```
admin@admin: ~/Documents/_git/senai-git (master)
$ git push -u origin master
Counting objects: 100% (4/4), done.
Delta compression using up to 2 threads
Compressing objects: 100% (2/2), done.
Writing objects: 100% (3/3), 343 bytes | 343.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/luis2senai/senai-git.git
   ad6c1c..b0146e2 master -> master
Branch 'master' set up to track remote branch 'master' from 'origin'.
admin@admin: ~/Documents/_git/senai-git (master)
$ |
```

Volte ao navegador e atualize o servidor remoto para verificar as alterações. O arquivo .gitignore deve estar entre os arquivos do projeto enquanto o arquivo-não-publicado não aparecer, pois todos os arquivos que estiverem indicados dentro do arquivo .gitignore não devem ser monitorados.

