



# RESIDÊNCIA EM TECNOLOGIA DA INFORMAÇÃO E COMUNICAÇÃO

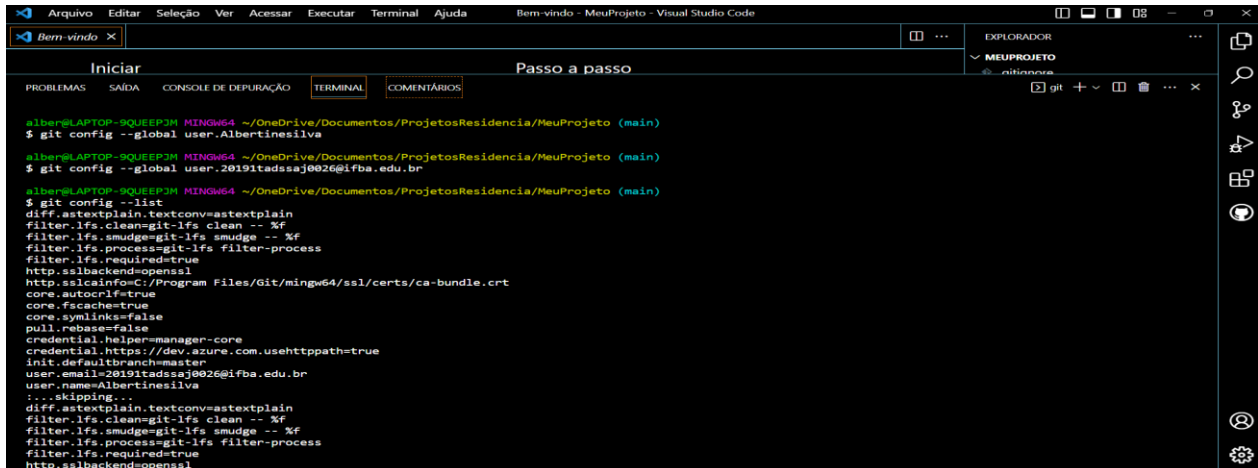
ALBERT SILVA DE JESUS

## GIT E GITHUB

Atividade PI-P002 de versionamento de código com Git e no repositório Github apresentada ao Alvaro Degas Coelho, como requisito parcial para aprovação no modulo de programação imperativa – T3 de .

## Exercício 1: Instale e configure o Git

Configure globalmente seu nome de usuário e e-mail no Git.



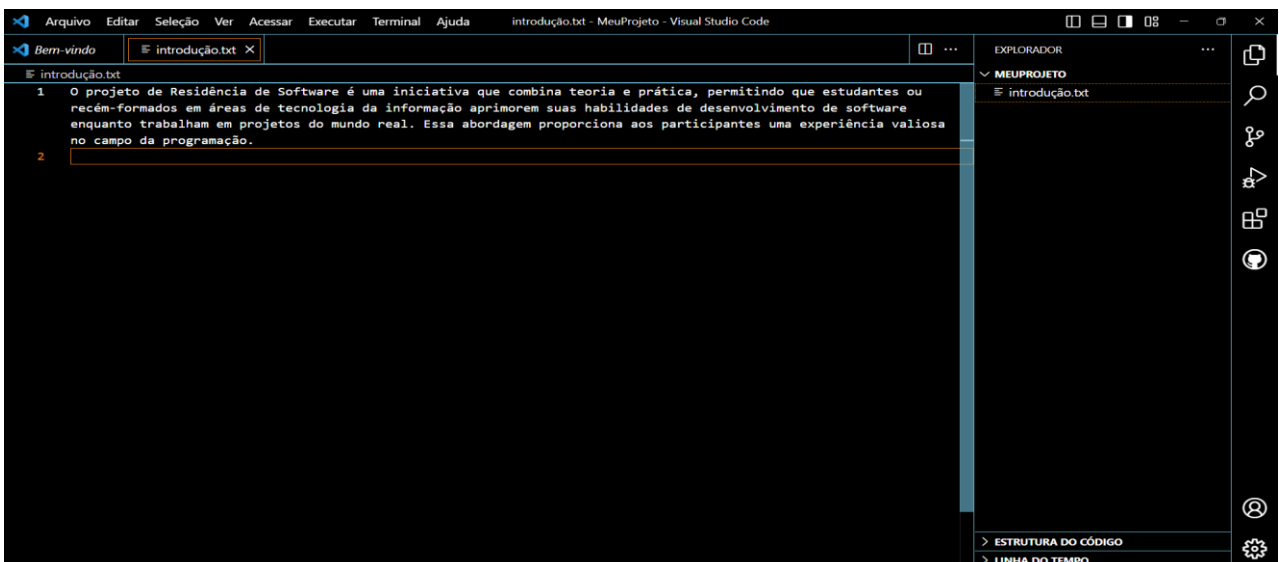
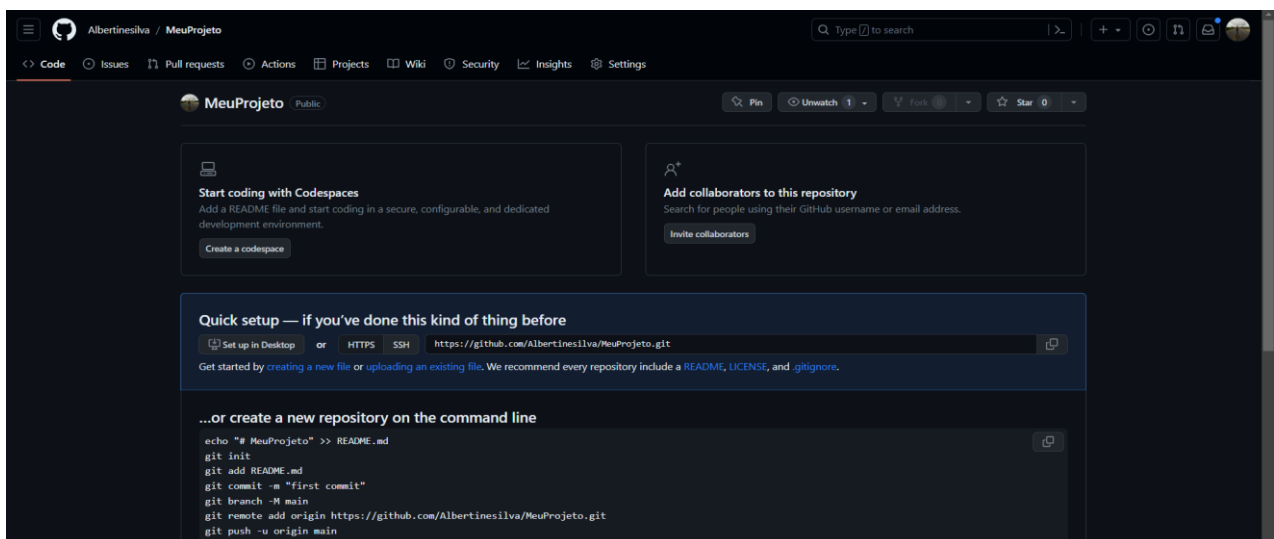
```
albert@LAPTOP-SQUEEP3M MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (main)
$ git config --global user.Albertinesilva

albert@LAPTOP-SQUEEP3M MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (main)
$ git config --global user.20191tadssaj0026@ifba.edu.br

albert@LAPTOP-SQUEEP3M MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (main)
$ git config --list
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
http.sslcainfo=C:/Program Files/Git/mingw64/ssl/certs/ca-bundle.crt
core.autocrlf=true
core.fscache=true
core.symlinks=false
pull.rebase=false
credential.helper=manager-core
credential.https://dev.azure.com.usehttppath=true
init.defaultbranch=master
user.email=20191tadssaj0026@ifba.edu.br
user.name=Albertinesilva
...skipping...
diff.astextplain.textconv=astextplain
filter.lfs.clean=git-lfs clean -- %f
filter.lfs.smudge=git-lfs smudge -- %f
filter.lfs.process=git-lfs filter-process
filter.lfs.required=true
http.sslbackend=openssl
```

## Exercício 2: Iniciando um Repositório

Crie um repositório Git em um diretório vazio chamado "MeuProjeto". Crie um arquivo chamado "introducao.txt" e adicione uma breve introdução ao projeto.



### Exercício 3: Trabalhando com Branches

Crie um branch chamado "feature-navbar". No arquivo "introducao.txt", adicione informações sobre um novo recurso a ser adicionado ao projeto. Faça um commit das alterações no branch "feature-navbar". Volte para o branch principal (master) e verifique que as alterações do branch "feature-navbar" não estão visíveis. Realize o merge das alterações do branch "feature-navbar" para o branch principal.

```
Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda MeuProjeto - Visual Studio Code

PROBLEMAS SAÍDA CONSOLE DE DEPURACÃO TERMINAL COMENTÁRIOS

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git branch
* master

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git branch feature-navbar

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git branch
* master
  feature-navbar

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git checkout feature-navbar
Switched to branch 'feature-navbar'

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (feature-navbar)
$
```

```
Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda introdução.txt - MeuProjeto - Visual Studio Code

introdução.txt

1 O projeto de Residência de Software é uma iniciativa que combina teoria e prática, permitindo que estudantes ou recém-formados em áreas de tecnologia
da informação aprimorem suas habilidades de desenvolvimento de software enquanto trabalham em projetos do mundo real. Essa abordagem proporciona aos
participantes uma experiência valiosa no campo da programação.
2
3 1º A etapa inicial compreende o módulo de programação imperativa - T3.
4
5 2º A etapa seguinte compreende o módulo de programação básica em java, python e .net
6

PROBLEMAS SAÍDA CONSOLE DE DEPURACÃO TERMINAL COMENTÁRIOS

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (feature-navbar)
$ git add .

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (feature-navbar)
$ git commit -m "Projeto Atualizado"
[feature-navbar 118183f] Projeto Atualizado
2 files changed, 6 insertions(+)
create mode 100644 .gitignore
create mode 100644 "introdu\303\247\303\243o.txt"

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (feature-navbar)
$
```

```
Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda secreto.txt - MeuProjeto - Visual Studio Code

introdução.txt secreto.txt U

1 Residência de Software, arquivo secreto!

PROBLEMAS SAÍDA CONSOLE DE DEPURACÃO TERMINAL COMENTÁRIOS

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (feature-navbar)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$
```

```
alber@LAPTOP-9QUEEP3M MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (feature-navbar)
$ git checkout master
Switched to branch 'master'
Your branch is up to date with 'origin/master'.

alber@LAPTOP-9QUEEP3M MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git merge feature-navbar
Updating 12f9acc..118183f
Fast-forward
 .gitignore      | 1 +
 "introdu\303\247\303\243o.txt" | 5 +++++
 2 files changed, 6 insertions(+)
 create mode 100644 .gitignore
 create mode 100644 "introdu\303\247\303\243o.txt"

alber@LAPTOP-9QUEEP3M MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$
```

Albertinesilva / MeuProjeto

Code Issues Pull requests Actions Projects Wiki Security Insights Settings

MeuProjeto Public

master 2 branches 0 tags

Your master branch isn't protected

Albertinesilva Merge pull request #1 from Albertinesilva/feature-navbar

File	Commit	Time
.gitignore	Projeto Atualizado	13 minutes ago
README.md	Primeiro Commit	30 minutes ago
introdução.txt	Projeto Atualizado	13 minutes ago

MeuProjeto

## Exercício 4: Ignorando Arquivos

Crie um arquivo chamado "secreto.txt" e adicione-o ao .gitignore para evitar que seja versionado.

Arquivo Editar Seleção Ver Acessar Executar Terminal Ajuda MeuProjeto - Visual Studio Code

EXPLORADOR

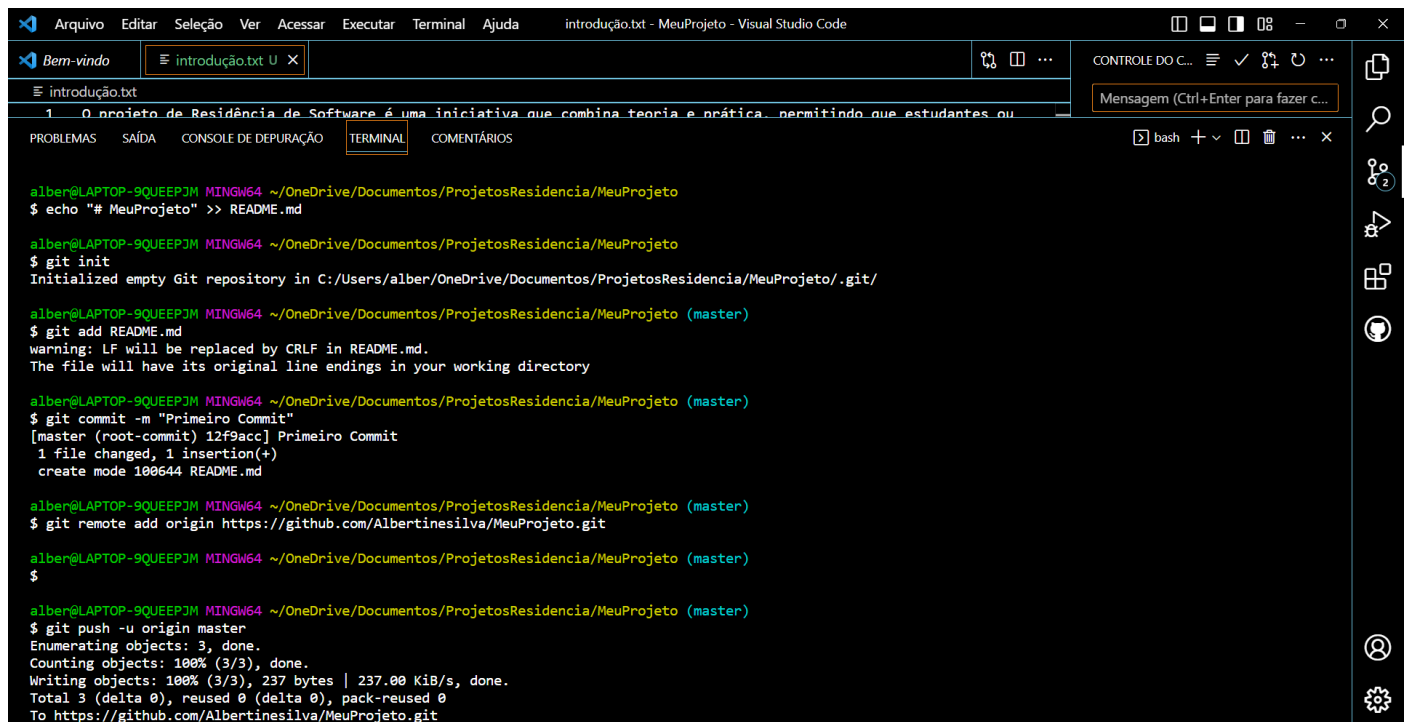
- MEUPROJETO
  - .gitignore
  - introdução.txt
  - README.md
  - secreto.txt

ESTRUTURA DO CÓDIGO

LINHA DO TEMPO

## Exercício 5: Trabalhando com Repositórios Remotos

Crie um repositório vazio no GitHub chamado "MeuProjeto". Conecte seu repositório local ao repositório remoto no GitHub. Envie (push) suas alterações locais para o repositório remoto.



```
alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto
$ echo "# MeuProjeto" >> README.md

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto
$ git init
Initialized empty Git repository in C:/Users/alber/OneDrive/Documentos/ProjetosResidencia/MeuProjeto/.git/

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git add README.md
warning: LF will be replaced by CRLF in README.md.
The file will have its original line endings in your working directory

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git commit -m "Primeiro Commit"
[master (root-commit) 12f9acc] Primeiro Commit
1 file changed, 1 insertion(+)
create mode 100644 README.md

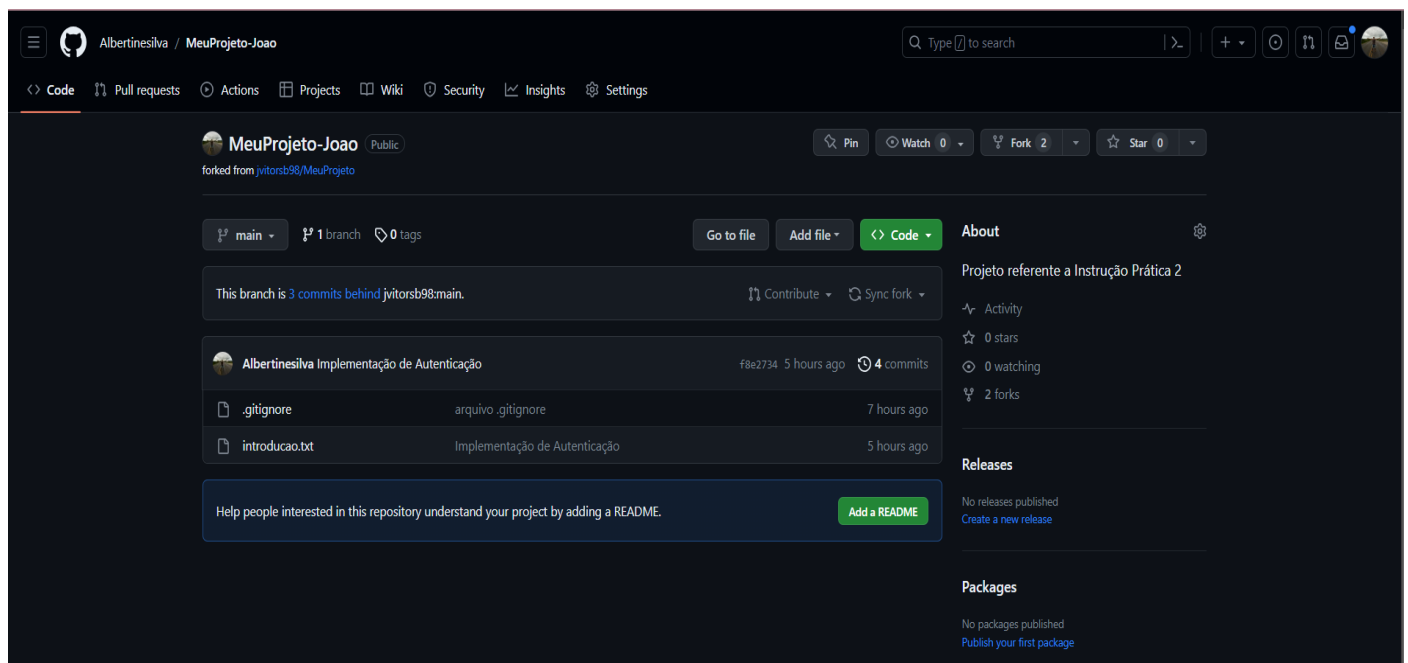
alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git remote add origin https://github.com/Albertinesilva/MeuProjeto.git

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git push -u origin master
Enumerating objects: 3, done.
Counting objects: 100% (3/3), done.
Writing objects: 100% (3/3), 237.00 KiB/s, done.
Total 3 (delta 0), reused 0 (delta 0), pack-reused 0
To https://github.com/Albertinesilva/MeuProjeto.git
```

## Exercício 6: Clonando e Forking

Clone o repositório de outro participante para o seu computador. Adicione informações adicionais no arquivo "introducao.txt" e envie um pull request para o dono do repositório original (no GitHub ou GitLab).



## Exercício 7: Resolvendo Conflitos

Trabalhe em um projeto colaborativo com pelo menos um outro colega. Faça alterações no mesmo arquivo e tente fazer um merge. Resolva o conflito manualmente.

The screenshot shows the Visual Studio Code interface with a file named 'introducao.txt' open. The editor displays a merge conflict between the 'HEAD (Alteração Atual)' and the 'alber' branch. The conflict is located in the middle of the file, where the 'alber' branch has a change that conflicts with the current HEAD. The editor shows the conflicting lines with a yellow background and a red 'X' icon. A button labeled 'Resolver no Editor de Mesclagem' is visible. The Explorer sidebar on the right shows the project structure with 'MEUPROJETO-JOAO' containing '.gitignore' and 'introducao.txt'. The Terminal at the bottom shows the command 'git merge alber' and the resulting error message: 'CONFLICT (content): Merge conflict in introducao.txt. Automatic merge failed; fix conflicts and then commit the result.'

```
1 # Desenvolvimento de novo projeto
2
3 ## Este projeto consiste no desenvolvimento de uma interface para o site da LevelUp, onde o usuário irá navegar
4 até o game de sua preferência.
5
6 ### Será inicialmente implementado uma tela responsiva para a MainWindow
7
8 ### Será implementado verificação de autenticação utilizando spring security.
9
10 <<<<<< HEAD (Alteração Atual)
11 ### A linguagem escolhida para back-end foi: java com sprig boot.
12 #####
13 ### A aplicação sera em Java com spring boot
14 >>>>>> alber (Alteração da Entrada)
15
```

Resolver no Editor de Mesclagem

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto-Joao/MeuProjeto-Joao (main)  
\$  
alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto-Joao/MeuProjeto-Joao (main)  
\$ git merge alber  
Auto-merging introducao.txt  
CONFLICT (content): Merge conflict in introducao.txt  
Automatic merge failed; fix conflicts and then commit the result.

## Exercício 8: Histórico de Commits

Utilize o comando git log para visualizar o histórico de commits em seu repositório. Examine as informações disponíveis em cada commit.

The screenshot shows the Visual Studio Code interface with a file named 'introdução.txt' open. The Explorer sidebar on the right shows the project structure with 'MEUPROJETO' containing '.gitignore', 'introdução.txt', and 'README.md'. The Terminal at the bottom shows the command 'git log' and the resulting output, which displays the commit history. The output includes the commit hash, author, date, and a description of the commit. The first commit is 'commit 1f09f7cbfa06ca98bf36f79268a0bf963025d423 (HEAD -> master, origin/master)' by 'Merge: 12f9acc 118183f'. The second commit is 'commit 118183ff2fd6f6fc0f56826916e8a86f8c50cd71 (origin/feature-navbar, feature-navbar)' by 'Author: Albertinesilva <20191tadssaj0026@ifba.edu.br>' with the date 'Wed Aug 30 19:07:36 2023 -0300'. The third commit is 'commit 12f9accb970cf214729825dc3f7af3c08db8c551' by 'Author: Albertinesilva <20191tadssaj0026@ifba.edu.br>' with the date 'Wed Aug 30 18:55:08 2023 -0300'. The fourth commit is 'commit 12f9accb970cf214729825dc3f7af3c08db8c551' by 'Author: Albertinesilva <20191tadssaj0026@ifba.edu.br>' with the date 'Wed Aug 30 18:37:52 2023 -0300'.

```
alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git log
commit 1f09f7cbfa06ca98bf36f79268a0bf963025d423 (HEAD -> master, origin/master)
Merge: 12f9acc 118183f
Author: Albert <102269168+Albertinesilva@users.noreply.github.com>
Date: Wed Aug 30 19:07:36 2023 -0300

    Merge pull request #1 from Albertinesilva/feature-navbar

    Projeto Atualizado

commit 118183ff2fd6f6fc0f56826916e8a86f8c50cd71 (origin/feature-navbar, feature-navbar)
Author: Albertinesilva <20191tadssaj0026@ifba.edu.br>
Date: Wed Aug 30 18:55:08 2023 -0300

    Projeto Atualizado

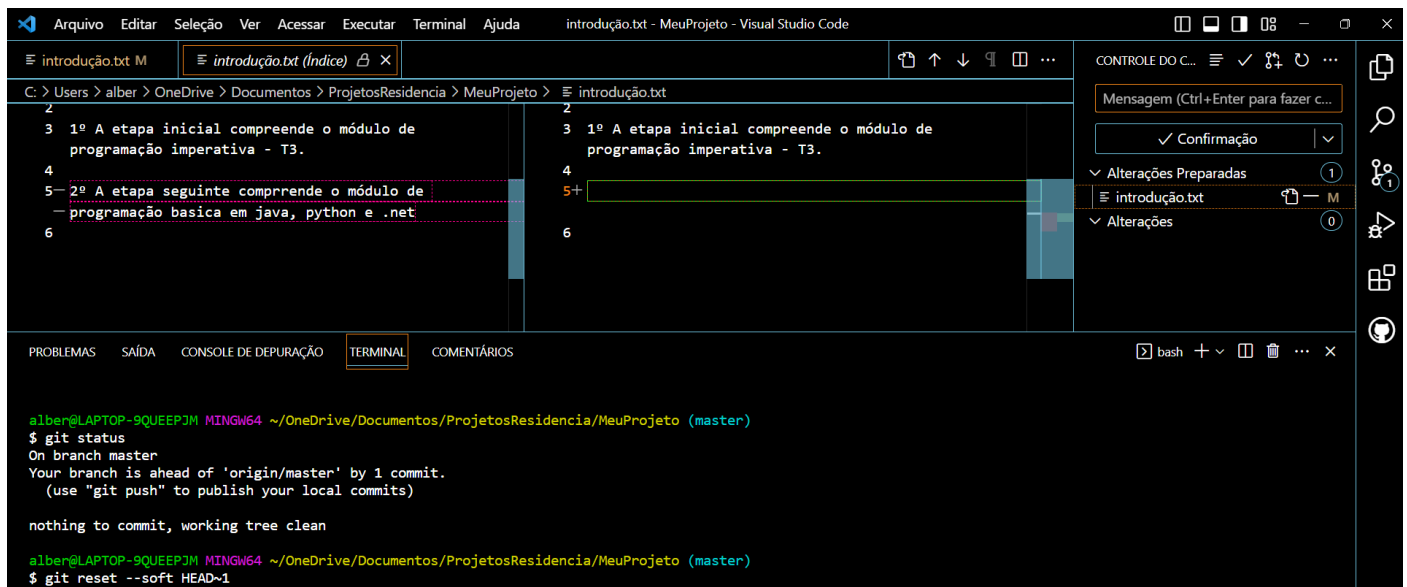
commit 12f9accb970cf214729825dc3f7af3c08db8c551
Author: Albertinesilva <20191tadssaj0026@ifba.edu.br>
Date: Wed Aug 30 18:37:52 2023 -0300

    Primeiro Commit

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$
```

## Exercício 9: Revertendo Alterações

Faça uma alteração indesejada no arquivo "introducao.txt" e realize um commit. Utilize o comando git revert para desfazer essa alteração sem excluir a história do commit.



The screenshot shows the Visual Studio Code interface. The top editor pane displays a diff for 'introdução.txt', showing a change from line 5 to line 5+ where a new line was added. The bottom editor pane shows the terminal output of the following commands:

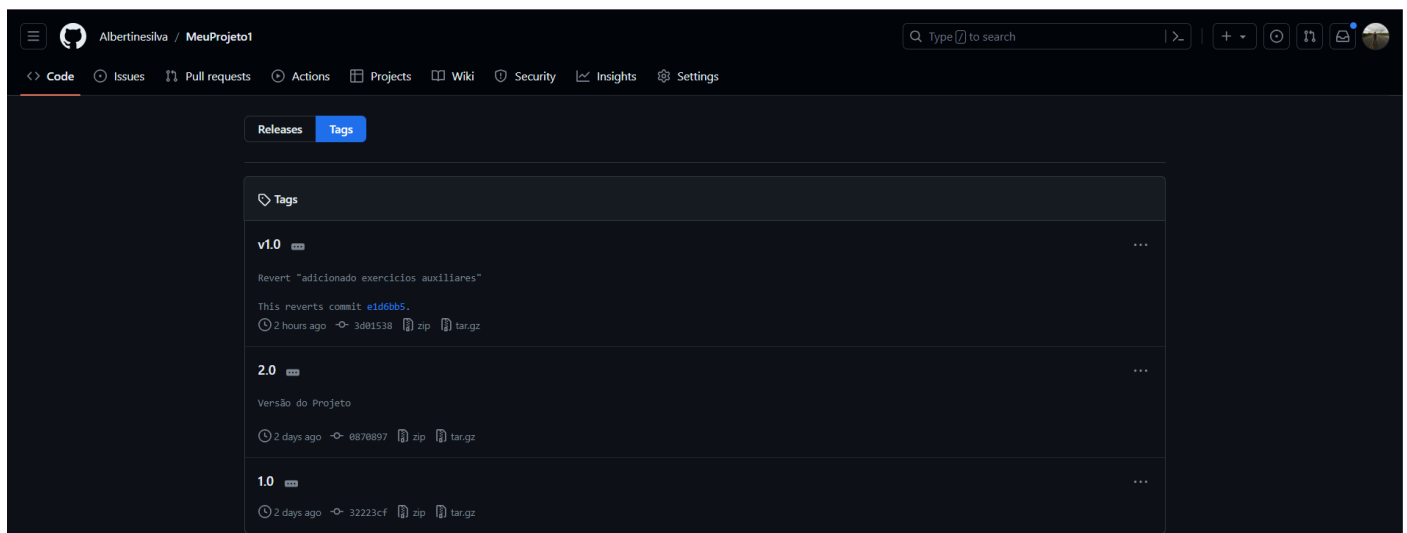
```
alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git status
On branch master
Your branch is ahead of 'origin/master' by 1 commit.
(use "git push" to publish your local commits)

nothing to commit, working tree clean

alber@LAPTOP-9QUEEPJM MINGW64 ~/OneDrive/Documentos/ProjetosResidencia/MeuProjeto (master)
$ git reset --soft HEAD~1
```

## Exercício 10: Trabalhando com Tags

Crie uma tag no seu repositório para marcar a versão 1.0 do projeto. Adicione informações adicionais no arquivo "introducao.txt" após a criação da tag e crie uma nova tag para marcar a versão 2.0.



## Desafios na Utilização de Git e Resolução de Conflitos

Ao trabalhar com o controle de versão Git, uma das etapas mais desafiadoras é lidar com a resolução de conflitos. Conflitos ocorrem quando diferentes colaboradores realizam modificações em um mesmo arquivo ou linha de código, resultando em uma situação em que o Git não pode determinar automaticamente como combinar as alterações. Esses conflitos podem ocorrer durante

um merge, pull ou rebase, e a resolução adequada requer compreensão, comunicação e habilidades técnicas.

Uma das dificuldades iniciais é identificar a origem dos conflitos. Quando o Git sinaliza um conflito, muitas vezes é necessário revisar e comparar as alterações feitas por diferentes colaboradores para determinar quais partes do código foram modificadas e onde os conflitos surgiram. Isso exige uma compreensão detalhada das mudanças recentes e da estrutura do código.

Além disso, a resolução de conflitos pode ser uma tarefa complexa, especialmente em projetos grandes ou com muitos colaboradores. O desenvolvedor precisa entender o contexto das alterações, o impacto delas no código e o motivo pelo qual foram feitas. Isso pode ser desafiador quando se trabalha com partes do código que não estão familiares.

O processo de resolução de conflitos também exige habilidades de comunicação e colaboração. Em projetos de equipe, é importante manter uma comunicação aberta para garantir que todos os envolvidos estejam cientes dos conflitos e possam discutir as melhores abordagens para a resolução. Decisões devem ser tomadas em conjunto para evitar perda de funcionalidades ou introdução de erros.

Uma vez que os conflitos são identificados, o desenvolvedor precisa decidir qual versão das alterações deve ser mantida e como integrá-las harmoniosamente. Isso pode envolver editar manualmente o código, mesclar as modificações de forma coesa e garantir que a lógica do programa seja preservada. É uma tarefa que exige atenção aos detalhes para evitar a introdução de bugs sutis.

Outra dificuldade é assegurar que a resolução de conflitos seja rastreável e bem documentada. Os commits relacionados à resolução de conflitos devem conter informações claras sobre quais decisões foram tomadas e por quê. Isso é essencial para rastrear a evolução do código e entender como os problemas foram abordados.

Embora o Git seja uma ferramenta poderosa para controle de versão e colaboração, a resolução de conflitos é uma parte crucial do processo que exige compreensão técnica, habilidades de comunicação e atenção aos detalhes. Lidar com conflitos de forma eficaz pode garantir a qualidade do código, a coesão da equipe e a continuidade do desenvolvimento sem interrupções.