

INF335 – Ambientes para Concepção de Software

Formação de Especialista em Engenharia de Software

INSTITUTO DE COMPUTAÇÃO – UNICAMP

1º SEMESTRE DE 2024

Prof. Rodrigo Bonacin rbonacin@unicamp.br

22/Junho/2024 TRABALHO 3

1 Objetivos

Este trabalho tem como objetivo desenvolver habilidades de uso de sistema de controle de versão “Git”. Este trabalho compõe a avaliação da disciplina INF335, e poderá ser realizado individualmente ou em dupla.

2 Atividade

Baixar código no Moodle na mesma pasta desta atividade.

1. Inserir código no GitHub (ou outro repositório Git remoto).
2. Clonar repositório no próprio computador (local).
3. Criar nova *branch* para correções (mesmas do Trabalho 1)
4. Alterar localmente, colocar no *stage* e fazer *commit* no repositório local.
5. Fazer *merge* das *branches* criada localmente
6. Atualizar o GitHub (*push*)

3 Entrega

Os seguintes itens devem ser entregues:

1. Link com repositório GitHub com históricos de alterações, desde a inclusão dos arquivos até a correção e merge.
2. Relatório com cada tela (*print*) da execução no repositório local com todos os comandos executados (desde a clonagem do repositório do GitHub até o “*push*” no repositório remoto).
3. Print com log de alterações/*commits* no repositório local (*git log --oneline -decorate --graph --all*).

Deverá ser entregue um arquivo por dupla, constando os respectivos nomes e números de matrículas. As entregas deverão ser realizadas pelo Moodle.

DATA FINAL DE ENTREGA: 28/06/2024

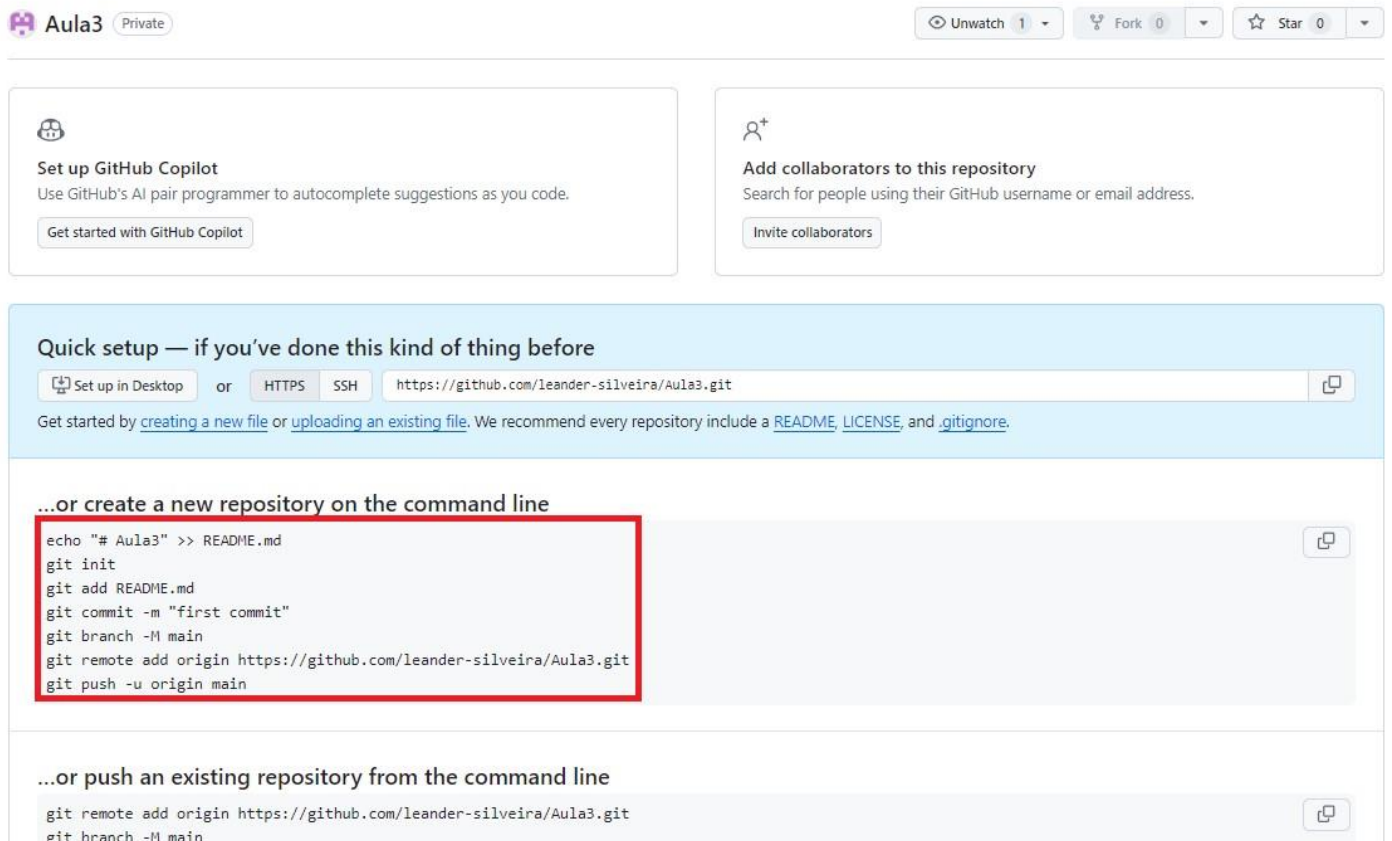
1. Link com repositório GitHub com históricos de alterações, desde a inclusão dos arquivos até a correção e merge

Link do repositório: <https://github.com/leander-silveira/Aula3>

2. Relatório com cada tela (*print*) da execução no repositório local com todos os comandos executados (desde a clonagem do repositório do GitHub até o “*push*” no repositório remoto)

2.1. Inserir código no GitHub (ou outro repositório Git remoto)

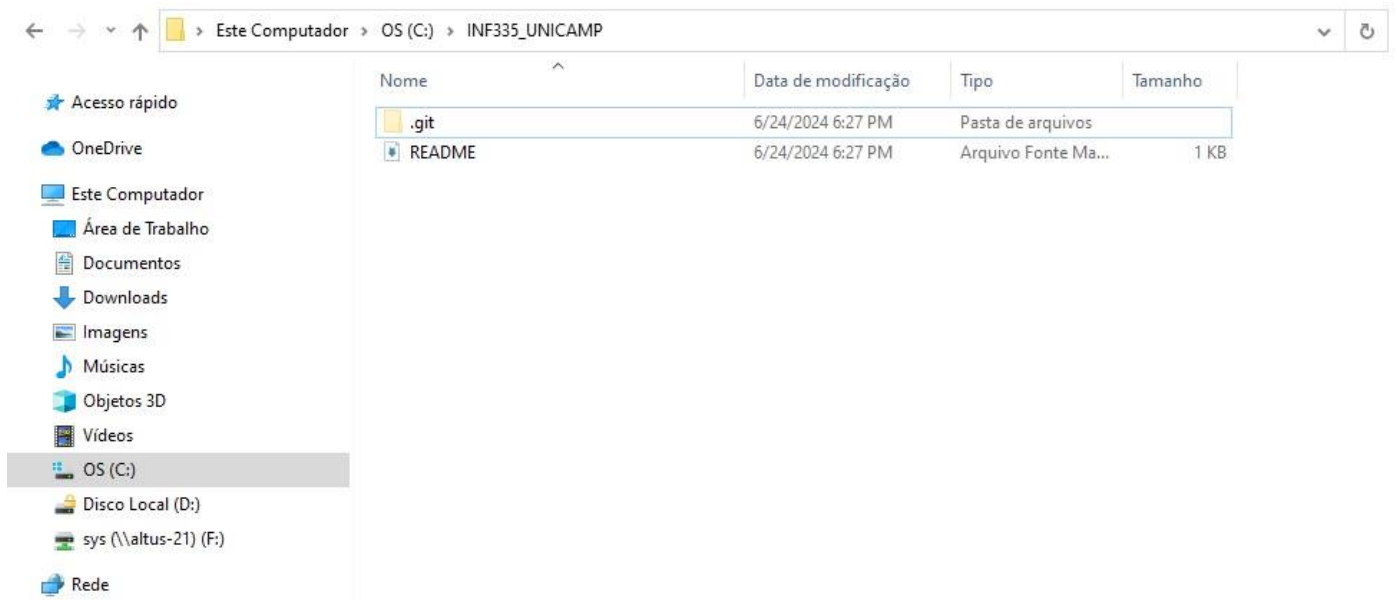
Criado repositório no GitHub.



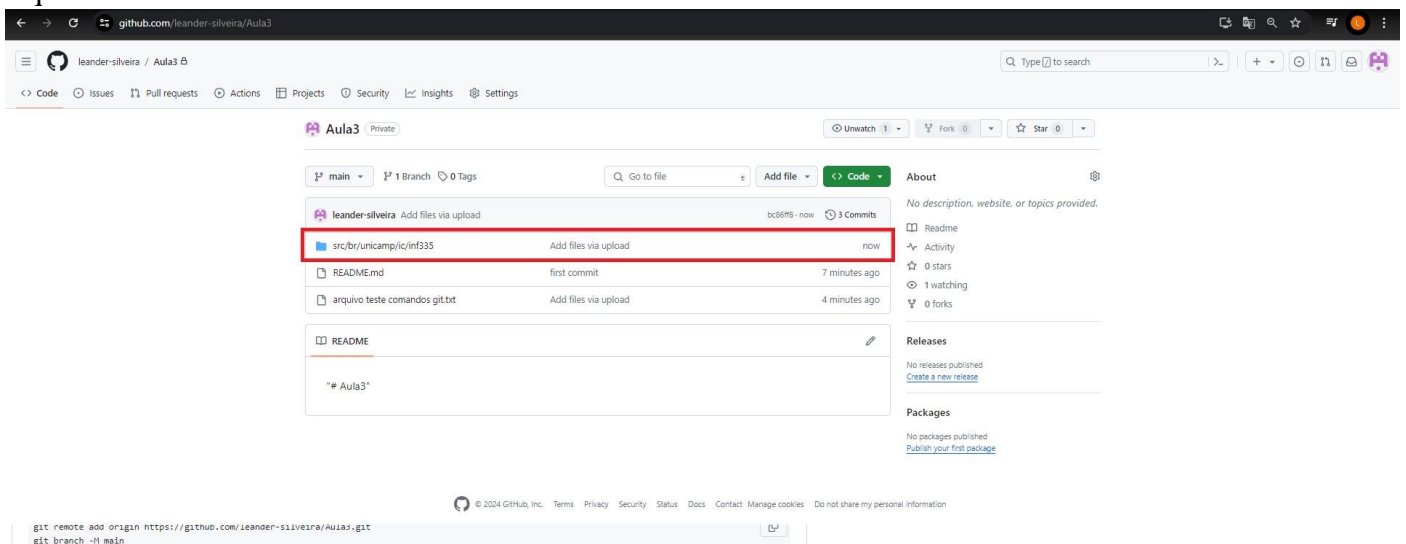
Executado os comandos a seguir no cmd do Windows.

```
echo "# Aula3" >> README.md
git init
git add README.md
git commit -m "first commit"
git branch -M main
git remote add origin https://github.com/leander-silveira/Aula3.git
git push -u origin main
```

Criada pasta local INF335_UNICAMP.



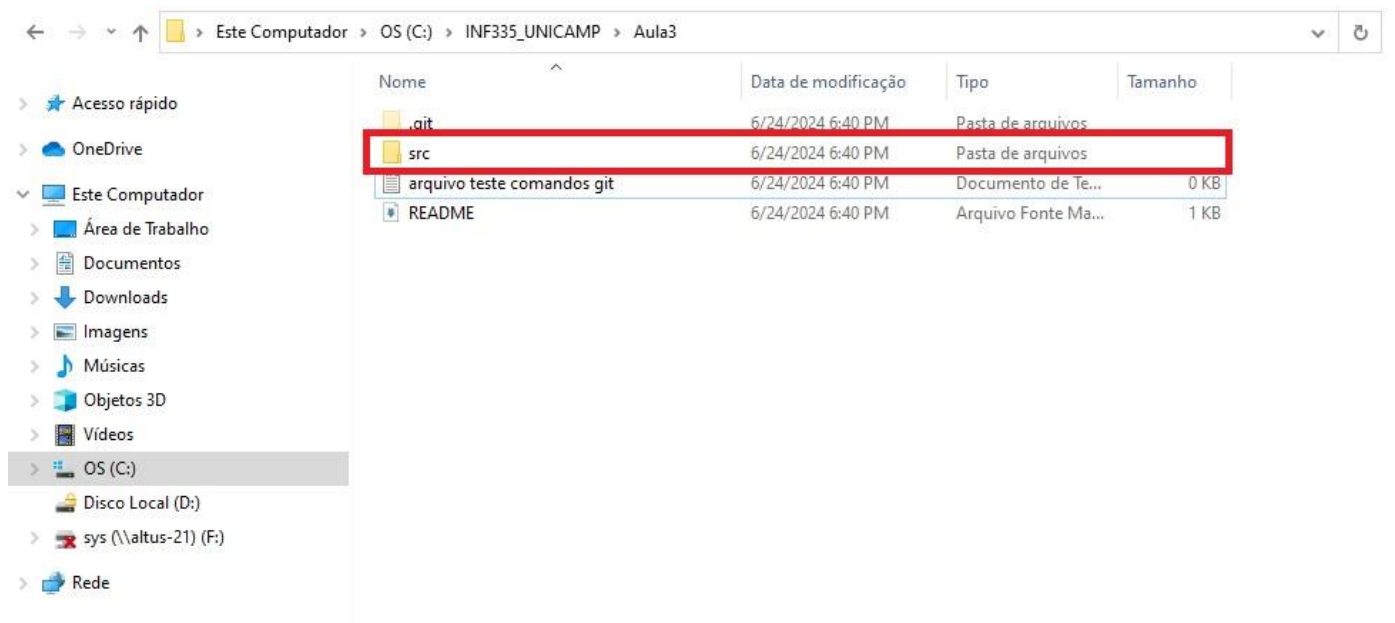
Arquivo inserido no GitHub.



2.2. Clonar repositório no próprio computador (local)

Para clonar o repositório na pasta local é utilizado o comando: git clone <https://github.com/leander-silveira/Aula3>

```
C:\INF335_UNICAMP>git clone https://github.com/leander-silveira/Aula3
Cloning into 'Aula3'...
remote: Enumerating objects: 16, done.
remote: Counting objects: 100% (16/16), done.
remote: Compressing objects: 100% (8/8), done.
remote: Total 16 (delta 1), reused 3 (delta 0), pack-reused 0
Receiving objects: 100% (16/16), done.
Resolving deltas: 100% (1/1), done.
C:\INF335_UNICAMP>
```



2.3. Criar nova *branch* para correções (mesmas do Trabalho 1)

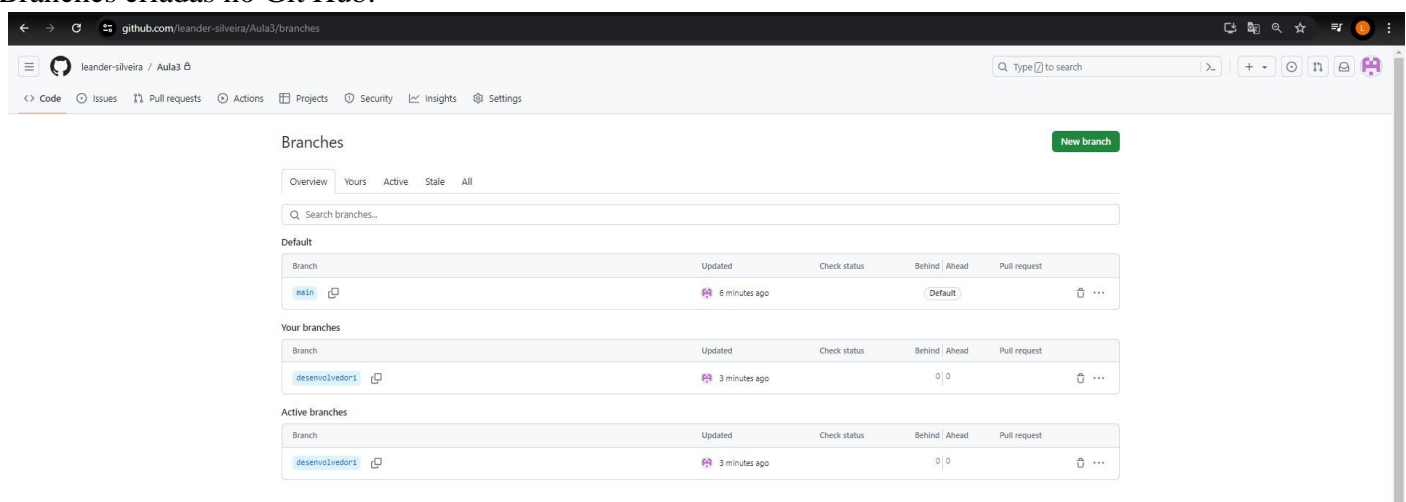
Para criar uma nova *branch* chamada *desenvolvedor1* é utilizado o comando: `git branch desenvolvedor1`.

```
C:\INF335_UNICAMP\Aula3>git branch desenvolvedor1  
  
C:\INF335_UNICAMP\Aula3>git branch  
desenvolvedor1  
* main
```

Em seguida para enviar a nova *branch* para o repositório remoto no GitHub, é utilizado o comando `git push origin desenvolvedor1`.

```
C:\INF335_UNICAMP\Aula3>git push origin desenvolvedor1  
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)  
remote:  
remote: Create a pull request for 'desenvolvedor1' on GitHub by visiting:  
remote:   https://github.com/leander-silveira/Aula3/pull/new/desenvolvedor1  
remote:  
To https://github.com/leander-silveira/Aula3  
* [new branch]   desenvolvedor1 -> desenvolvedor1
```

Branches criadas no Git Hub.



2.4. Alterar localmente, colocar no *stage* e fazer *commit* no repositório local

Antes de iniciar a modificação é necessário mudar para branch `desenvolvedor1`. Para isso é utilizado o comando `git checkout desenvolvedor1`. Além disso, é necessário realizar o comando `git push -u origin desenvolvedor1` para realizar o primeiro commit. Assim, os demais poderão ser realizados por `git push`.

```
C:\INF335_UNICAMP\Aula3>git checkout desenvolvedor1
Switched to branch 'desenvolvedor1'

C:\INF335_UNICAMP\Aula3>git branch
* desenvolvedor1
  main
```

A atualização da classe `Brecho.java` é feita utilizando a branch `desenvolvedor1` e utilizando os seguintes comandos: `cd C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335` (entrar na pasta local) `git add "Brecho.java"` (arquivo adicionado ao versionamento) `git commit -a -m "correção classe Brecho"` (commit) `git push` (envio para GitHub)

```
C:\INF335_UNICAMP\Aula3>git branch
* desenvolvedor1
  main

C:\INF335_UNICAMP\Aula3>cd C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335>git add "Brecho.java"
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335>git commit -a -m "correção classe Brecho"
[desenvolvedor1 20c5bc9] correção classe Brecho
Committer: Leander P. Silveira <leander.silveira@altus.com.br>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

    git config --global --edit

After doing this, you may fix the identity used for this commit with:

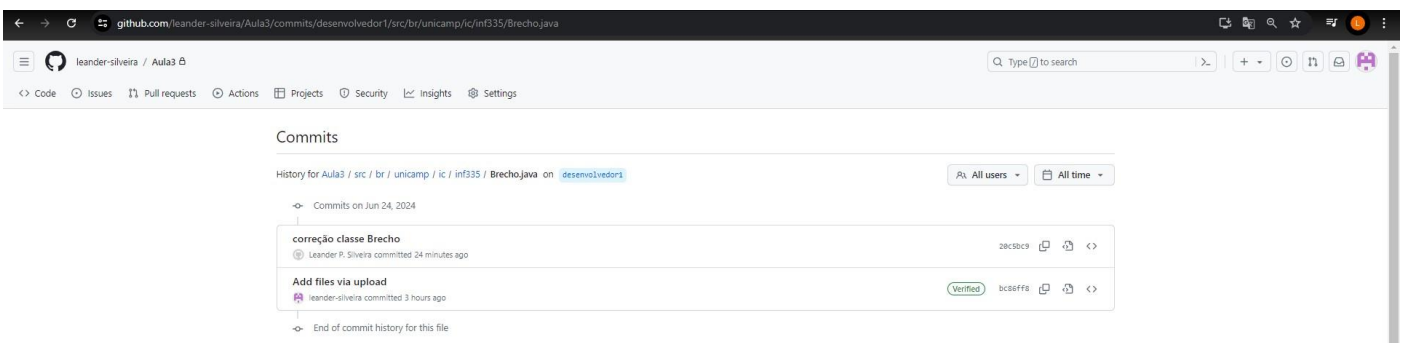
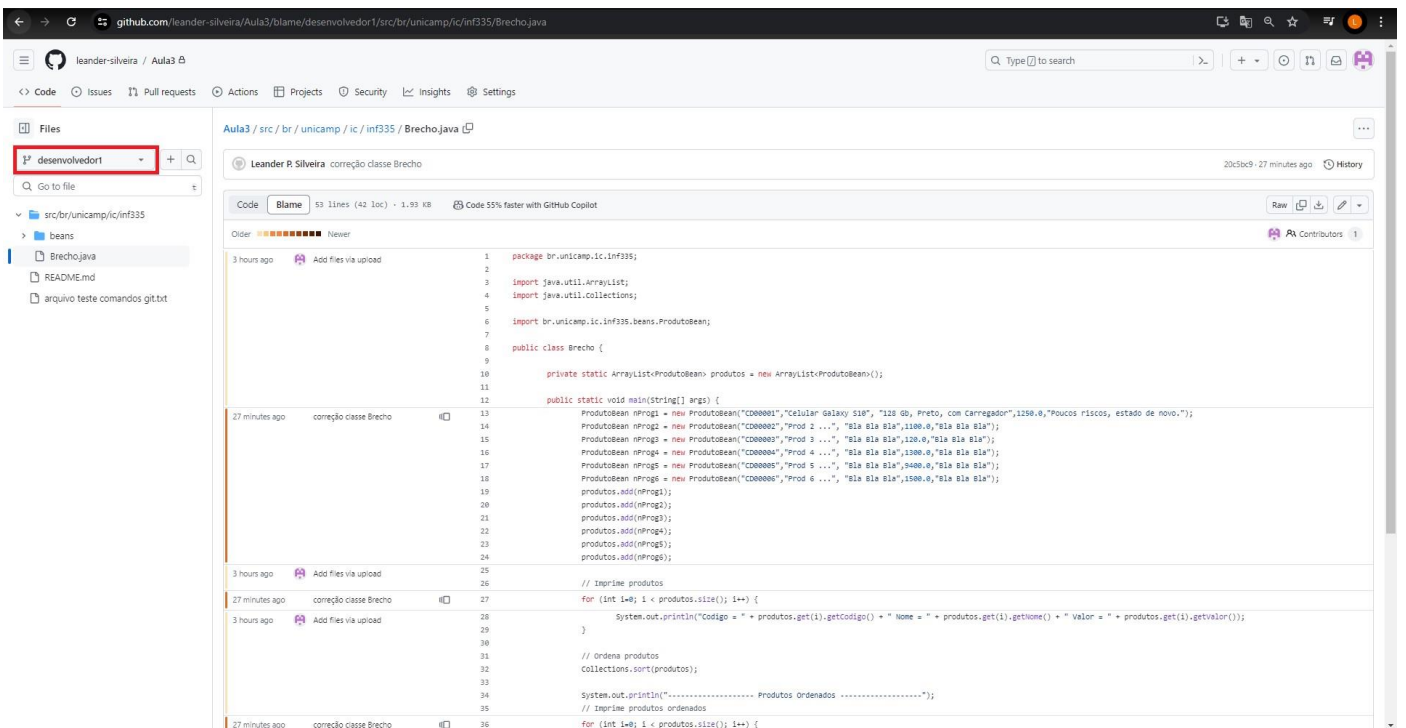
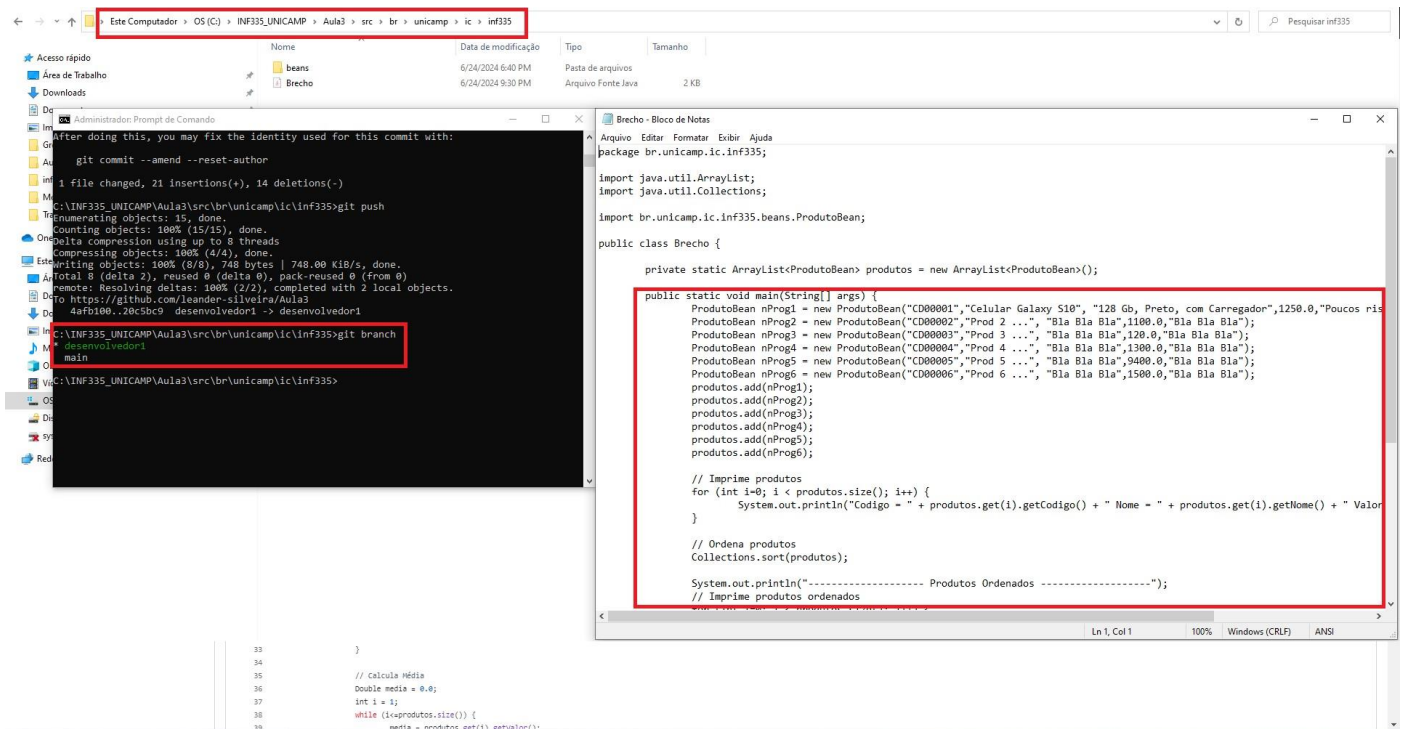
    git commit --amend --reset-author

1 file changed, 21 insertions(+), 14 deletions(-)

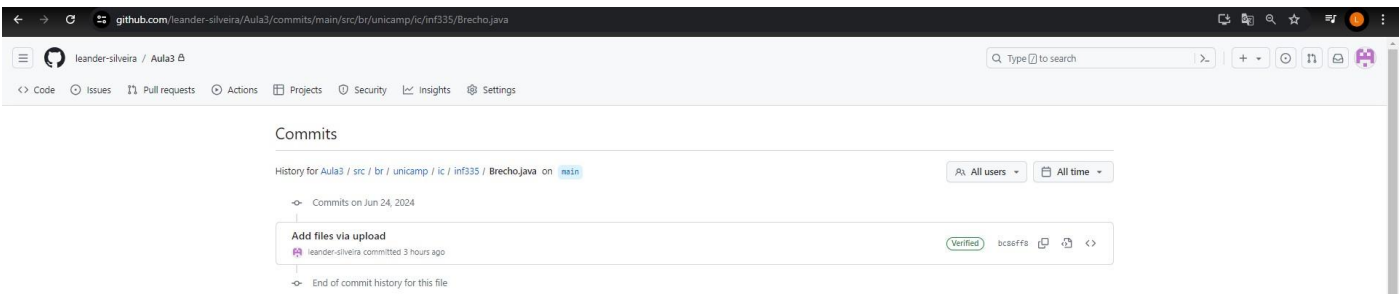
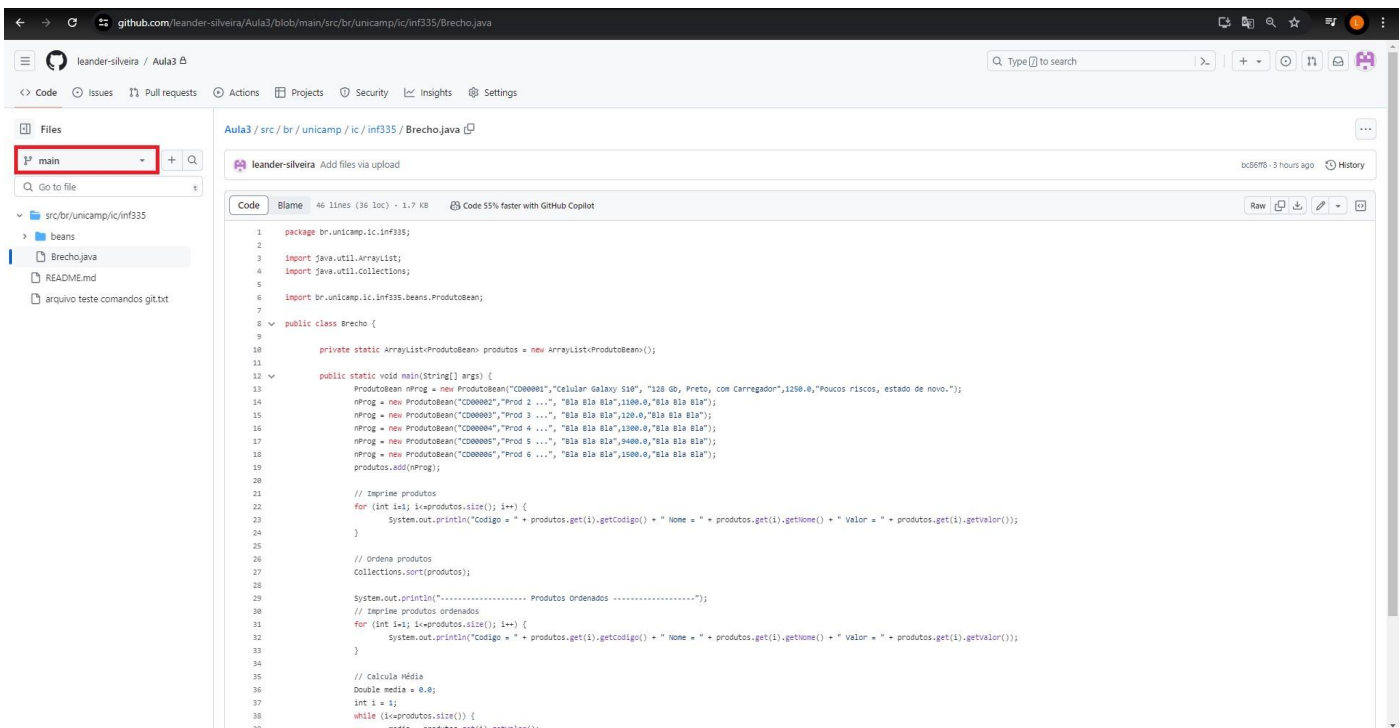
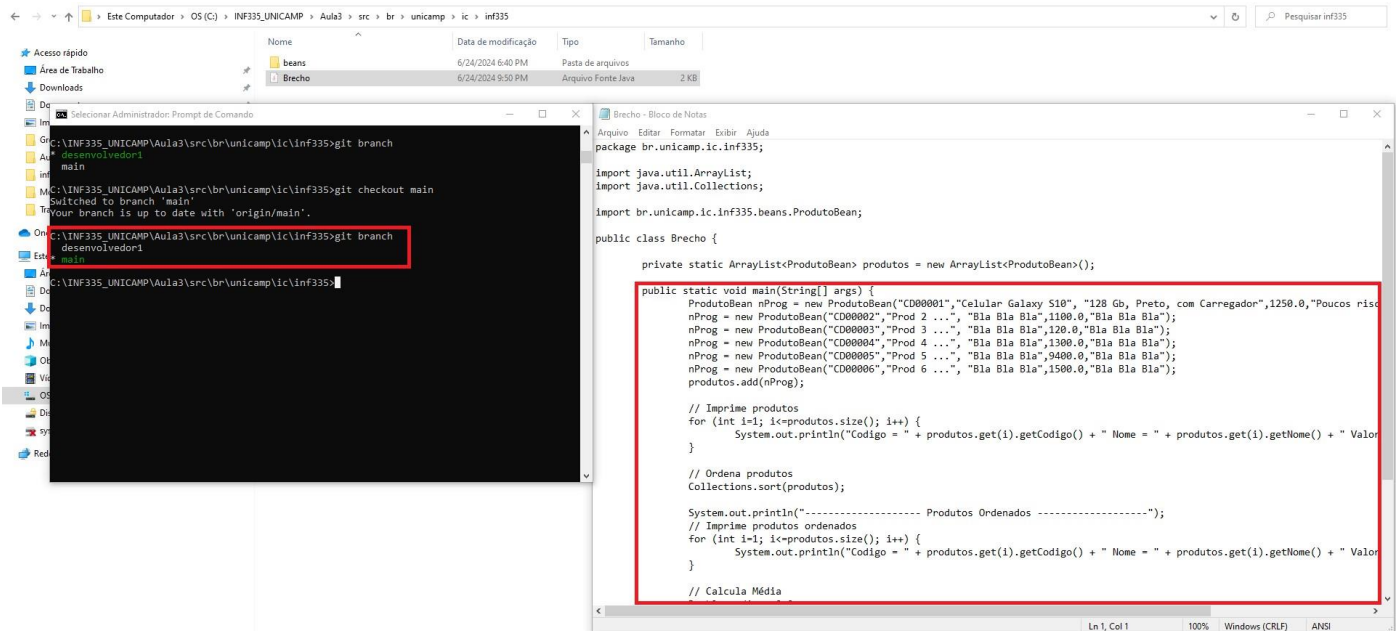
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335>git push
Enumerating objects: 15, done.
Counting objects: 100% (15/15), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (8/8), 748 bytes | 748.00 KiB/s, done.
Total 8 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/leander-silveira/Aula3
  4afb100..20c5bc9  desenvolvedor1 -> desenvolvedor1

C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335>
```

É possível observar, na branch `desenvolvedor1`, que tanto a pasta local quanto no GitHub a atualização da classe `Brecho.java` foi realizada.



Já na branch main, a atualização tanto na pasta local quanto no GitHub ainda não está aplicada, pois falta fundir a branch `desenvolvedor1` com a `main` por meio do comando `merge`.



Para modificação da classe `ProdutoBean.java` foi criado uma nova branch chamada `desenvolvedor2` e realizado os mesmos passos descritos no tópico 2.4. Os prints são apresentados a seguir.

Criação da branch `desenvolvedor2`.

```
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git branch desenvolvedor2
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git branch
desenvolvedor1
desenvolvedor2
* main
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git push origin desenvolvedor2
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'desenvolvedor2' on GitHub by visiting:
remote:   https://github.com/leander-silveira/Aula3/pull/new/desenvolvedor2
remote:
To https://github.com/leander-silveira/Aula3
* [new branch]   desenvolvedor2 -> desenvolvedor2
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>
```

Adição da classe ProdutoBean.java atualizada no GitHub na branch desenvolvedor2.

```
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git branch
desenvolvedor1
* desenvolvedor2
main
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git add "ProdutoBean.java"
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git commit -a -m "correção classe ProdutoBean"
[desenvolvedor2 131864e] correção classe ProdutoBean
Committer: Leander P. Silveira <leander.silveira@altus.com.br>
Your name and email address were configured automatically based
on your username and hostname. Please check that they are accurate.
You can suppress this message by setting them explicitly. Run the
following command and follow the instructions in your editor to edit
your configuration file:

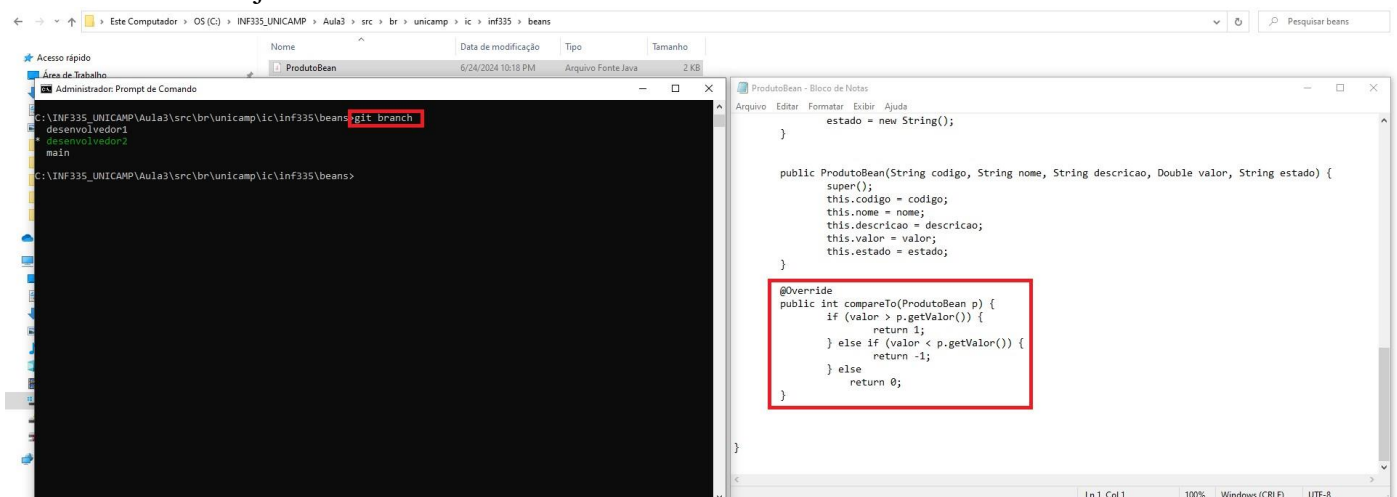
    git config --global --edit

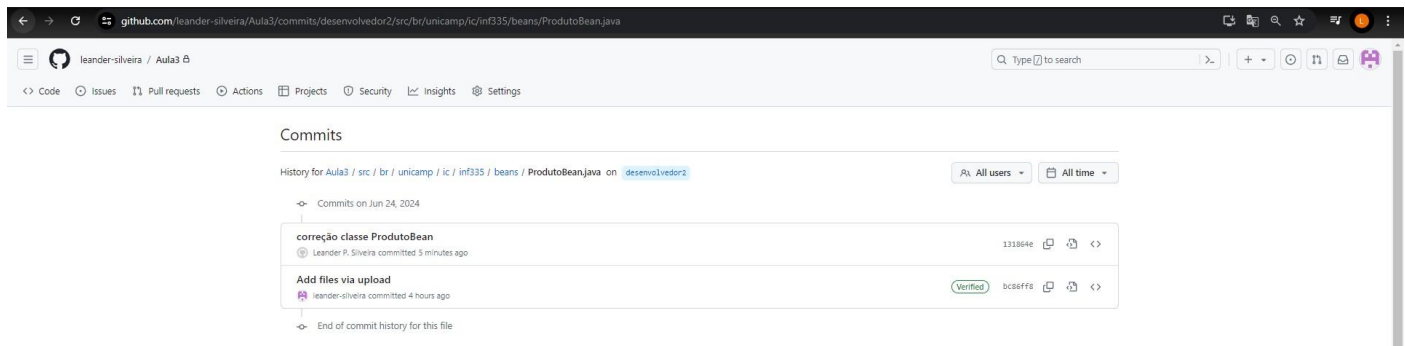
After doing this, you may fix the identity used for this commit with:

    git commit --amend --reset-author

1 file changed, 1 insertion(+), 1 deletion(-)
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git push -u origin desenvolvedor2
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 8 threads
Compressing objects: 100% (4/4), done.
Writing objects: 100% (9/9), 611 bytes | 611.00 KiB/s, done.
Total 9 (delta 2), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (2/2), completed with 2 local objects.
To https://github.com/leander-silveira/Aula3
20c5bc9..131864e desenvolvedor2 -> desenvolvedor2
branch 'desenvolvedor2' set up to track 'origin/desenvolvedor2'.
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>
```

É possível observar, na branch desenvolvedor2, que tanto na pasta local quanto no GitHub a atualização da classe ProdutoBean.java foi realizada.





The screenshot displays a Windows desktop with two open applications. On the left is a Windows Terminal window titled 'Administrador Prompt de Comando'. It shows a series of git commands and their outputs in a black background with green text. The commands are: `git branch`, `git checkout desenvolvedor1`, and `git branch`. The outputs show the current branch and the result of the checkout. The right application is a Notepad window titled 'ProdutoBean - Bloco de Notas'. It contains Java code for a `ProdutoBean` class. The code includes a constructor, a `super()` call, and a `compareTo` method. The `compareTo` method is highlighted with a red rectangle. The status bar at the bottom of the Notepad window shows 'Ln 1, Col 1', '100%', 'Windows (CRLF)', and 'UTF-8'.

```
Nome           Data de modificação   Tipo           Tamanho
-----
ProdutoBean    6/24/2024 10:43 PM   Arquivo Fonte Java   2 KB
```

```
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git branch
desenvolvedor1
desenvolvedor2
main

C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git checkout desenvolvedor1
Switched to branch "desenvolvedor1"
Your branch is up to date with 'origin/desenvolvedor1'.

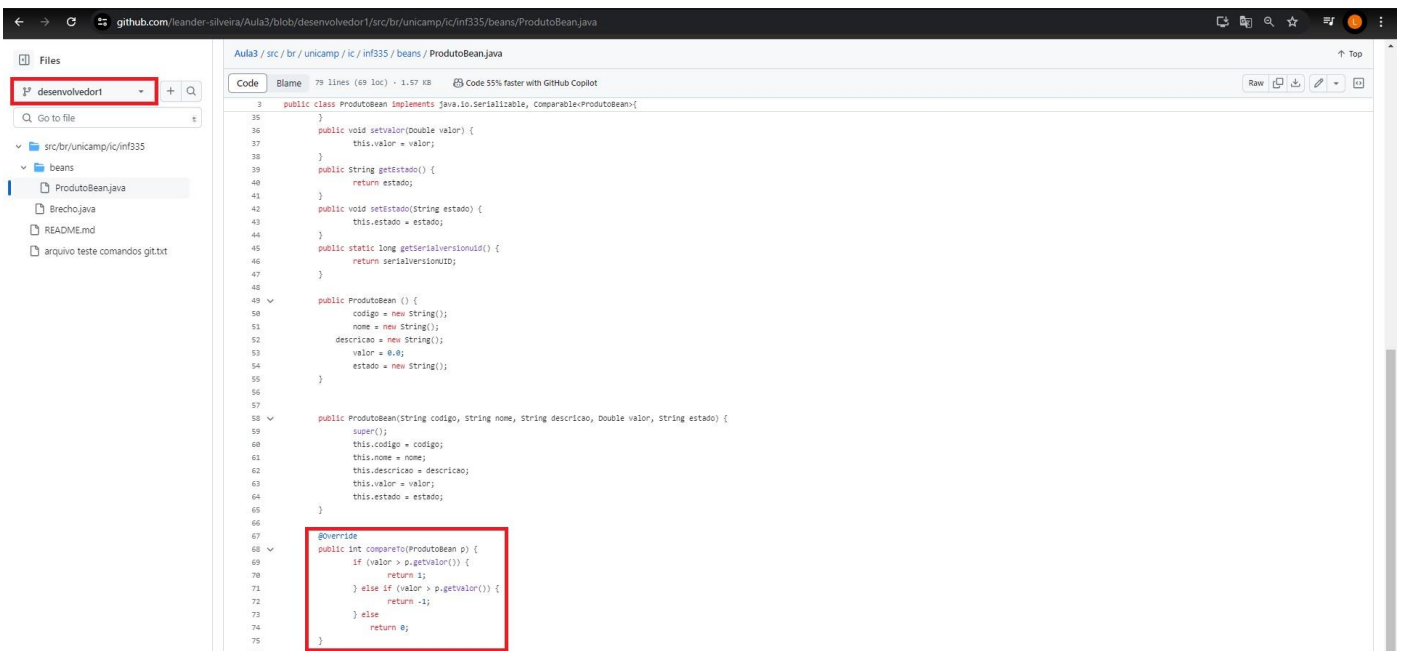
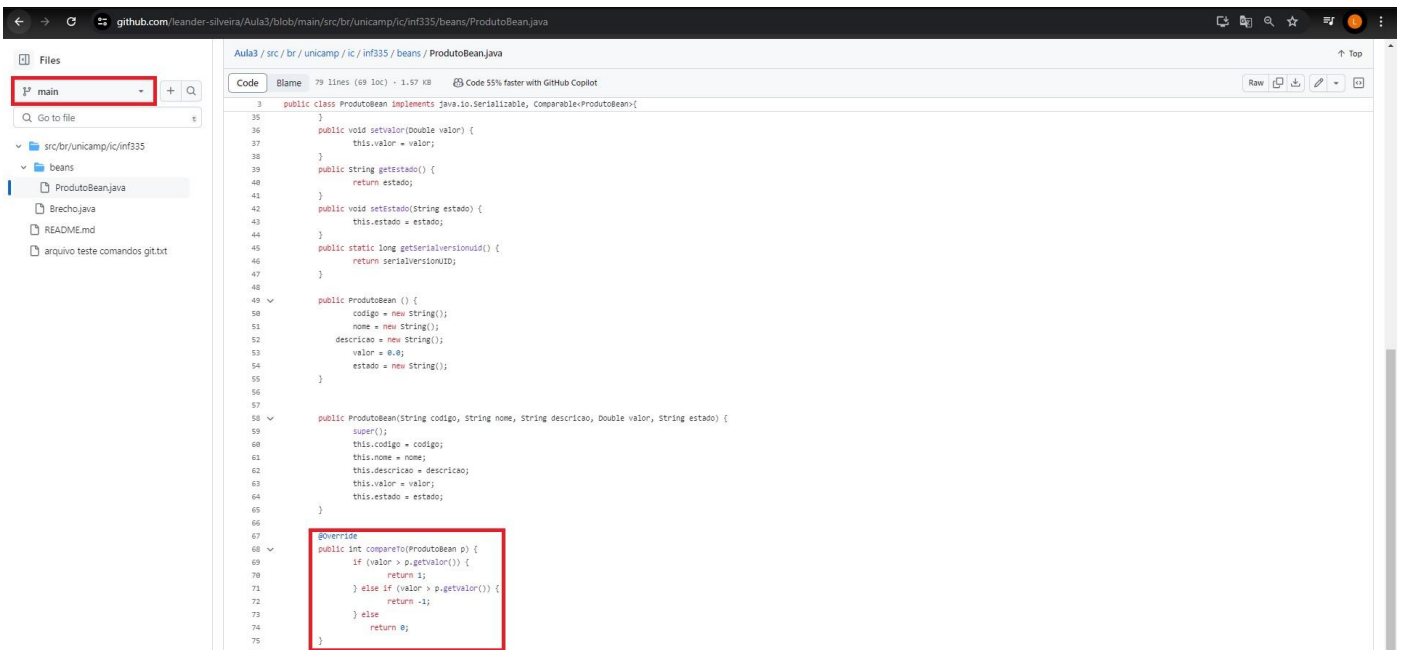
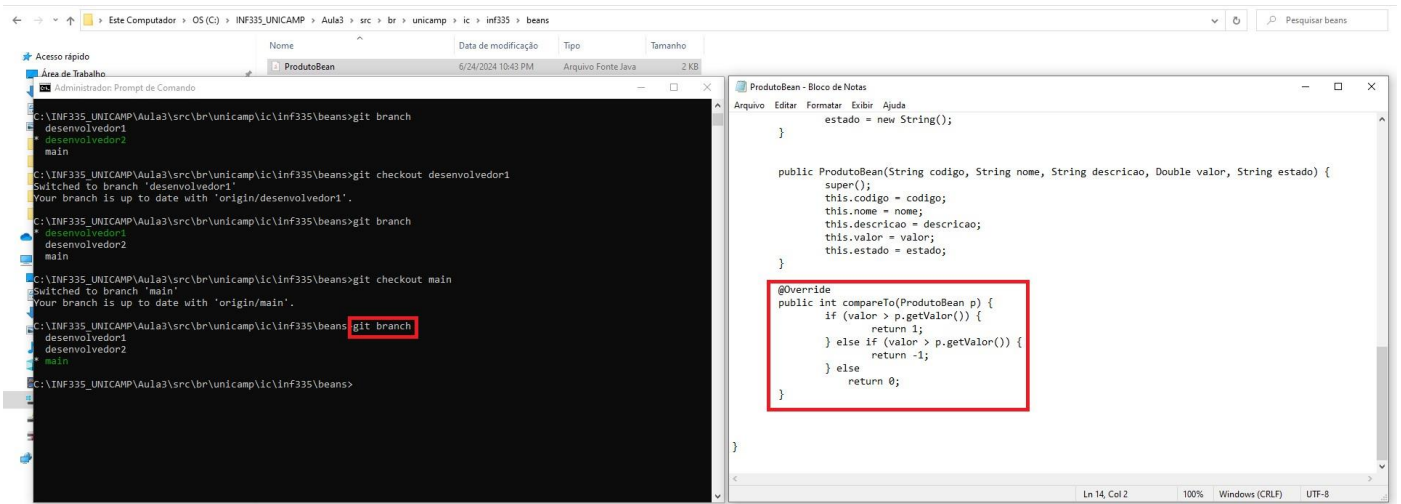
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git branch
desenvolvedor1
desenvolvedor2
main

C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>
```

```
public class ProdutoBean {
    private String codigo;
    private String nome;
    private String descricao;
    private Double valor;
    private String estado;

    public ProdutoBean(String codigo, String nome, String descricao, Double valor, String estado) {
        super();
        this.codigo = codigo;
        this.nome = nome;
        this.descricao = descricao;
        this.valor = valor;
        this.estado = estado;
    }

    @Override
    public int compareTo(ProdutoBean p) {
        if (valor > p.getValor()) {
            return 1;
        } else if (valor < p.getValor()) {
            return -1;
        } else {
            return 0;
        }
    }
}
```



2.5. Fazer *merge* das *branches* criada localmente & 2.6. Atualizar o GitHub (*push*)

Antes de fundir a branch *desenvolvedor1* com a *main*. É necessário entrar na branch de destino (*main*) que deseja adicionar as alterações realizadas na branch origem (*desenvolvedor1*). Para isso é utilizado o comando `git checkout main`.

```
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335>git branch
* desenvolvedor1
  main

C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

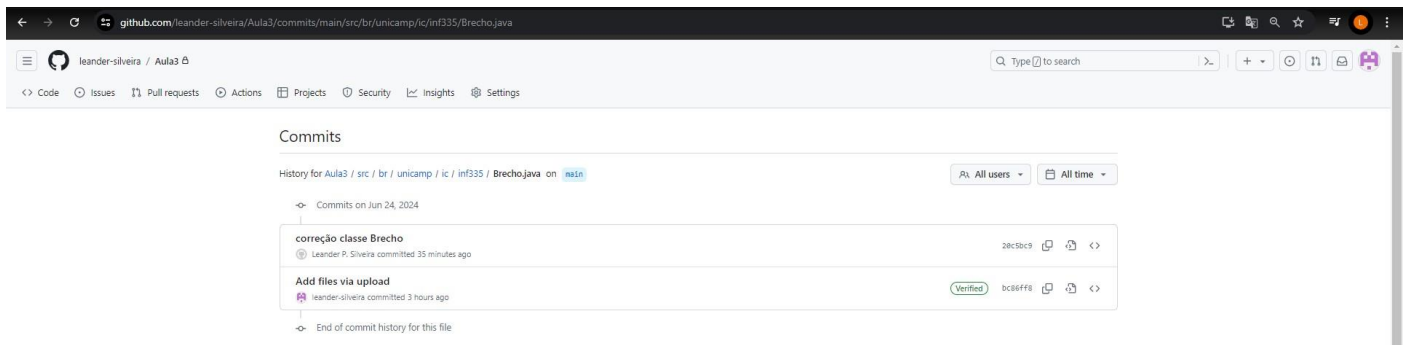
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335>git branch
desenvolvedor1
* main
```

Após estar na branch de destino (*main*). É executado o comando `git merge desenvolvedor1`. Em seguida, executado `git push` para enviar para o GitHub.

```
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335>git merge desenvolvedor1
Updating 4afb100..20c5bc9
Fast-forward
 src/br/unicamp/ic/inf335/Brecho.java | 35 ++++++
 1 file changed, 21 insertions(+), 14 deletions(-)

C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335>git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/leander-silveira/Aula3
 4afb100..20c5bc9  main -> main
```

The screenshot shows the GitHub web interface for the repository `leander-silveira / Aula3`. The file `src/br/unicamp/ic/inf335/Brecho.java` is selected. The interface displays the commit history for this file, showing three commits: two from 3 hours ago and one from 34 minutes ago. The 34-minute commit, titled "correção classe Brecho", is the current state of the file. The code editor shows the Java code for the `Brecho` class, which includes a `main` method that creates a list of `Produto` objects and prints their details. The code is highlighted with syntax coloring, and the interface includes standard GitHub navigation elements like the file explorer, search bar, and commit history table.



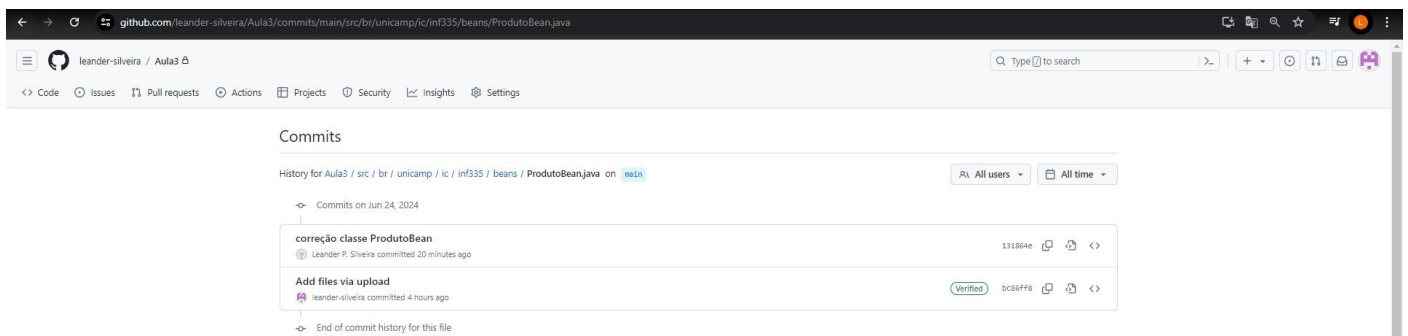
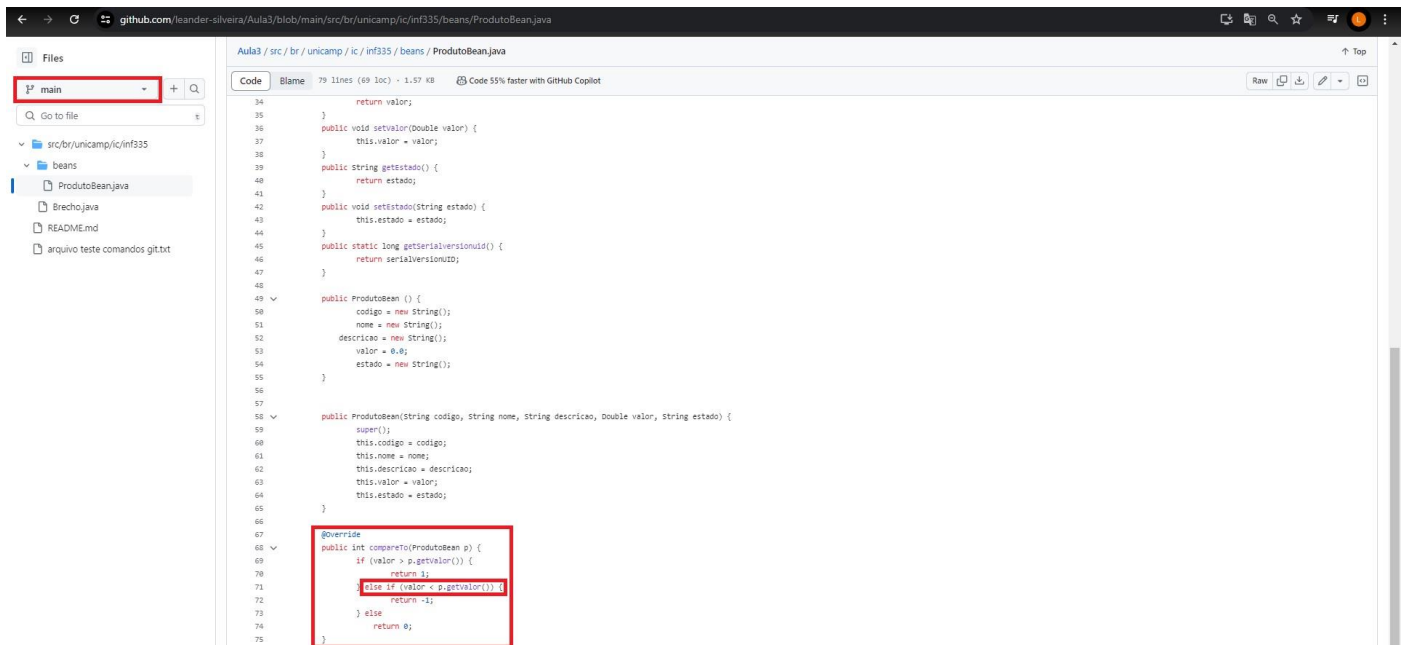
Para fundir a branch desenvolvedor2 com a main, foi utilizado os mesmos passos descritos nos tópicos 2.5 e 2.6. A seguir são apresentados os prints.

```
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

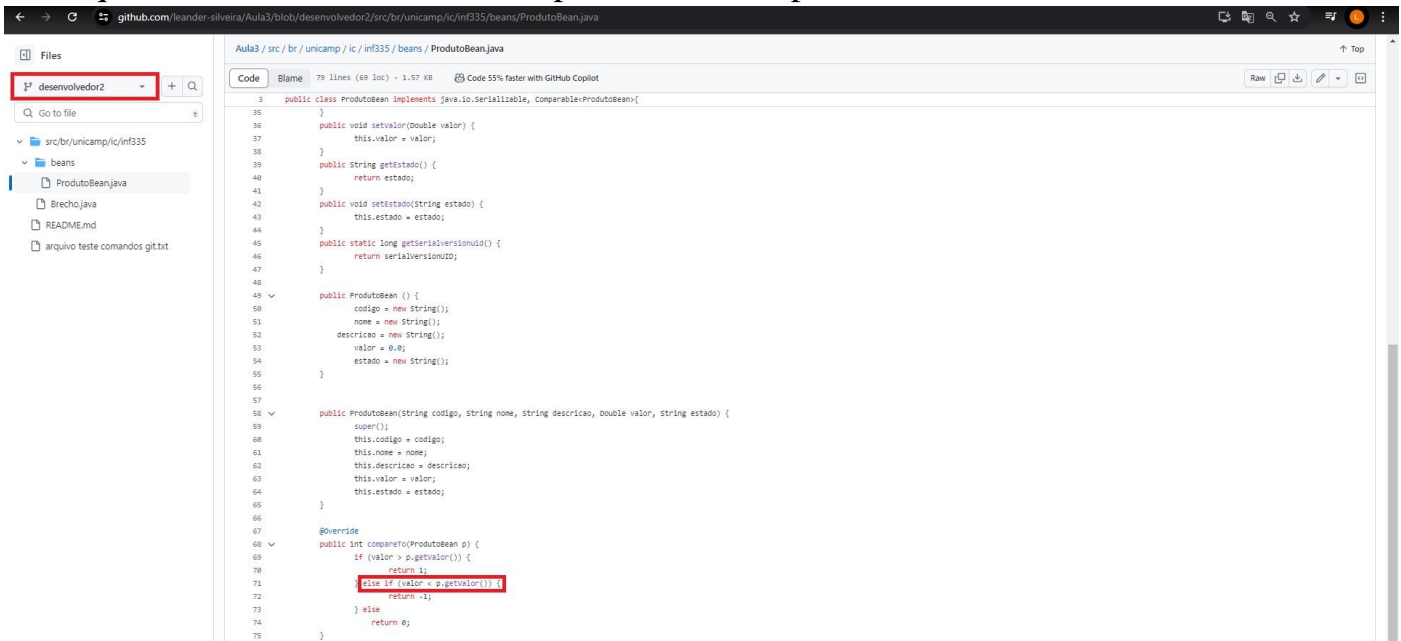
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git branch
desenvolvedor1
desenvolvedor2
* main

C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git merge desenvolvedor2
Updating 20c5bc9..131864e
Fast-forward
 src/br/unicamp/ic/inf335/beans/ProdutoBean.java | 2 +-
 1 file changed, 1 insertion(+), 1 deletion(-)

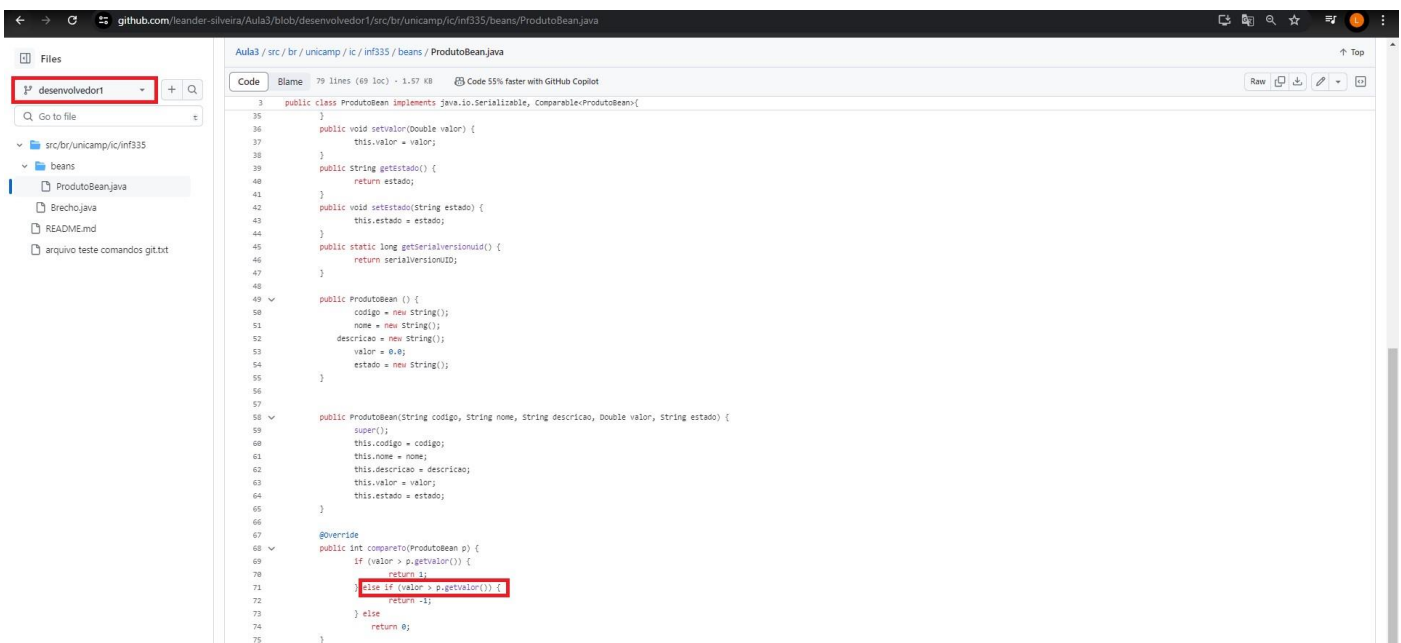
C:\INF335_UNICAMP\Aula3\src\br\unicamp\ic\inf335\beans>git push
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/leander-silveira/Aula3
 20c5bc9..131864e main -> main
```



É interessante destacar um exemplo da vantagem do GitHub para o desenvolvimento paralelo de software. Observe que a modificação da classe ProdutoBean.java realizada na branch desenvolvedor2 não é replicada na branch desenvolvedor1 de forma automática, demonstrando assim que são desenvolvimentos completamente independentes.

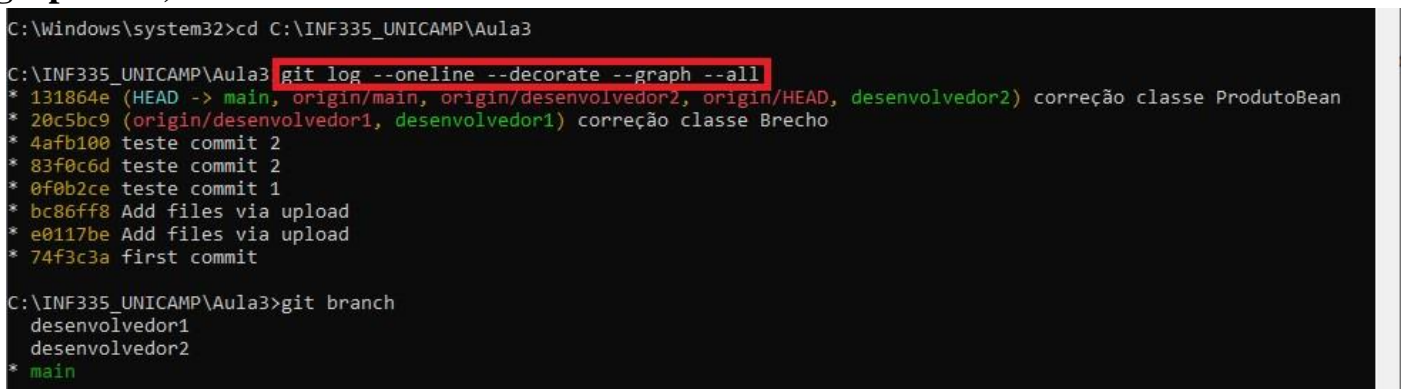


The screenshot shows the GitHub web interface for the repository 'leander-silveira/Aula3'. The file 'ProdutoBean.java' is open, and the 'desenvolvedor2' branch is selected. The code shows a class implementing Serializable and Comparable. A red box highlights a change in the compareTo method: 'else if (valor < p.getvalor()) {'.



The screenshot shows the GitHub web interface for the repository 'leander-silveira/Aula3'. The file 'ProdutoBean.java' is open, and the 'desenvolvedor1' branch is selected. The code shows a class implementing Serializable and Comparable. A red box highlights a change in the compareTo method: 'else if (valor > p.getvalor()) {'.

3. Print com log de alterações/commits no repositório local (git log --oneline --decorate --graph --all)



```
C:\Windows\system32>cd C:\INF335_UNICAMP\Aula3
C:\INF335_UNICAMP\Aula3>git log --oneline --decorate --graph --all
* 131864e (HEAD -> main, origin/main, origin/desenvolvedor2, origin/HEAD, desenvolvedor2) correção classe ProdutoBean
* 20c5bc9 (origin/desenvolvedor1, desenvolvedor1) correção classe Brecho
* 4afb100 teste commit 2
* 83f0c6d teste commit 2
* 0f0b2ce teste commit 1
* bc86ff8 Add files via upload
* e0117be Add files via upload
* 74f3c3a first commit
C:\INF335_UNICAMP\Aula3>git branch
desenvolvedor1
desenvolvedor2
* main
```