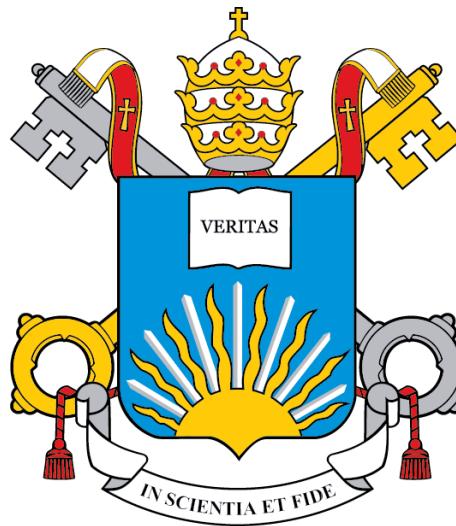


PONTIFÍCIA UNIVERSIDADE CATÓLICA DE GOIÁS
COORDENAÇÃO ENSINO A DISTÂNCIA – CEAD
ESCOLA POLITÉCNICA E DE ARTES
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS



PUC
GOIÁS

PROJETO INTEGRADOR IV – A

Isaías Correia De Moraes
Matrícula: 1132024100768

GOIÂNIA
2025.

Sumário

1. INTRODUÇÃO	1
2. OBJETIVO DO PROJETO	2
3. DESENVOLVIMENTO DO PROJETO	3
3.1. Ambiente de Desenvolvimento Utilizado	4
3.2. Organização do Projeto na IDE Eclipse	4
3.3. Pastas e Estrutura Geral do Projeto	5
4. ESTRUTURA DO CÓDIGO-FONTE	6
4.1. Classe Calculadora	6
4.2. Classe MenuCalculadora	7
4.3. Classe CalculadoraTeste (JUnit)	7
5. FUNCIONALIDADES IMPLEMENTADAS	8
5.1. Soma	8
5.2. Subtração	8
5.3. Multiplicação	8
5.4. Divisão	9
6. INTERFACE DO MENU INTERATIVO	10
7. TESTES AUTOMATIZADOS COM JUNIT 5	11
7.1. Estrutura dos Testes	11
7.2. Casos de Sucesso	11
7.3. Casos de Erro e Exceção	12
8. CONTROLE DE VERSÃO COM GIT E GITHUB	13
8.1. Criação do Repositório e Upload Inicial.....	13
8.2. Commits e Histórico de Versões	14
8.3. Branchs e Merge	14
9. CLONE DE REPOSITÓRIO EXTERNO	15
10. ORGANIZAÇÃO DAS PASTAS DO PROJETO	16
11. PRINTS DE TELA DO PROJETO	17
12. CONCLUSÃO	18
13. REFERÊNCIAS.....	19

1. INTRODUÇÃO

O Projeto Integrador IV – A, foi desenvolvido de forma individual com o propósito de desenvolver e aplicar de forma prática os conhecimentos adquiridos ao longo de aulas e atividades, sobre desenvolvimento em Java, Testes Automatizados com o Framework JUnit 5 e o uso de ferramentas com Git e GitHub. Por isso foi criada uma aplicação de calculadora com as quatro operações básicas (soma, subtração, multiplicação e divisão) e foi feita a implementação de uma mini interface de menu interativo para a facilitação e utilização do programa além disso envolve a implementação funcional, este projeto visa simular um ambiente de trabalho real e colaborativo com organização de códigos, versionamento em repositório remoto, uso de branches e merge, documentação e por último uma apresentação final em vídeo, ao longo da documentação será apresentada a mais detalhe as etapas de desenvolvimento do projeto realizado.

2. OBJETIVO DO PROJETO

O principal objetivo deste projeto é realizar uma aplicação feita na linguagem de programação Java simples que permita aplicar conceitos de desenvolvimento de software, com foco em temas de engenharia de software como Gerência de Configuração de Software e Gerência de Qualidade de Software, o projeto foi a realização de uma implementação de uma calculadora com as quatro operações matemáticas básicas, utilizando como apoio o Java Testes Automatizados com JUnit5 com o intuito de identificar possíveis falhas/erros na aplicação e utilizar também ferramentas de controle de versão por meio do Git e GitHub para gerenciar o código fonte de forma segura e organizada, isso inclui a criação de repositórios ramificações (branches) fusões (merge) commits e também o clone de repositórios externos e o tratamento de exceções em casos de divisão por zero e por fim realizar tudo de forma gerenciada e organizada.

3. DESENVOLVIMENTO DO PROJETO

No desenvolvimento do projeto fui utilizada ferramentas que foram apresentadas ao longo do curso e práticas alinhadas ao mercado de desenvolvimento de software, logo a seguir serão apresentadas as ferramentas que foram adotadas, a organização do projeto na IDE e a estrutura geral das pastas durante a implementação e as etapas que foram seguidas detalhadas.

3.1. Ambiente de Desenvolvimento Utilizado

Para o desenvolvimento da aplicação, foi utilizada a IDE Eclipse amplamente utilizada no mercado de trabalho ferramenta que suporta bibliotecas externas suporte a testes e amplamente utilizada para o desenvolvimento de aplicações Java, a linguagem de programação Java foi a que prevaleceu na sua versão mais recente, e para os testes automatizados foi usado o Framework JUnit 5 para testes unitários e funcionais, o JUnit 5 foi adicionado como biblioteca na IDE no projeto por meio da configuração de “Java Build Path” dentro das propriedades do Eclipse. Foram utilizadas também as ferramentas de versionamento de código o Git para versionar códigos e o GitHub para armazenamento de projetos, permitindo aplicar as práticas de Branches, Merges, Commits, Clone e repositórios tudo de maneira organizada.

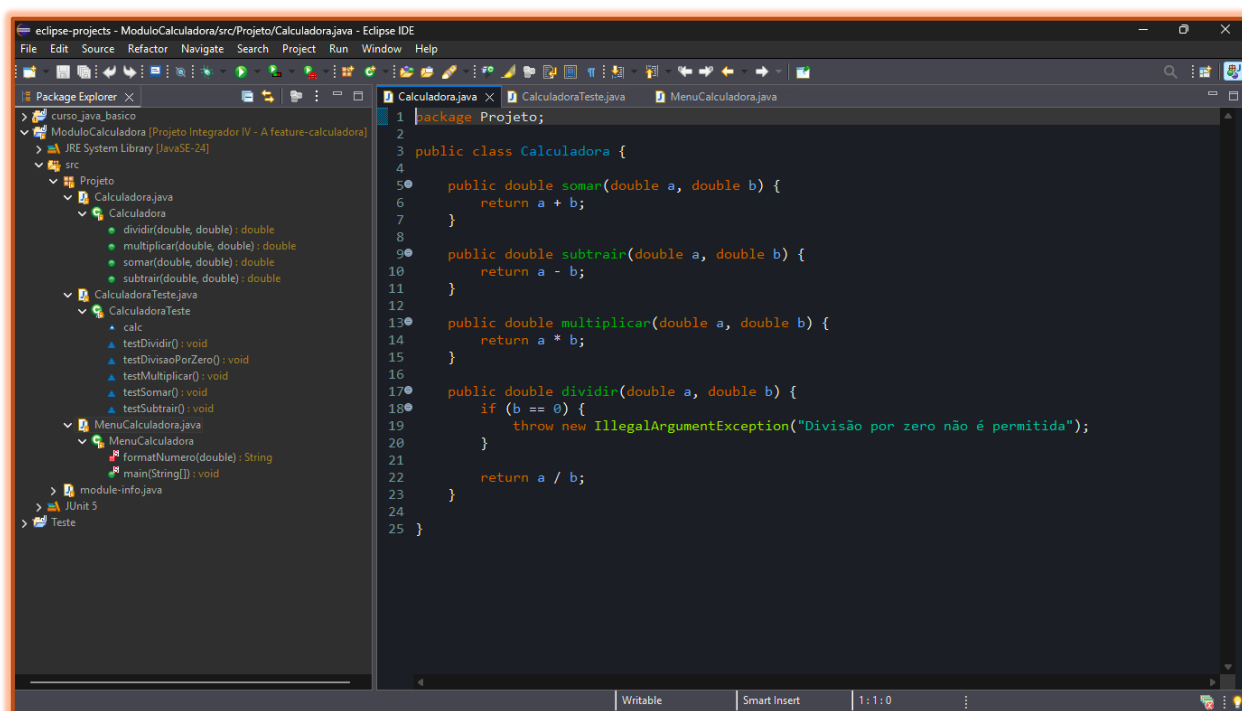


Figura 1: Imagem da IDE Eclipse sendo utilizada

3.2. Organização do Projeto na IDE Eclipse

Dentro da IDE Eclipse, foi criado o projeto Java com a seguinte configuração o nome do projeto foi “ModuloCalculadora”, depois para manter uma lógica estável e modular foi adicionado um pacote chamado Projeto e dentro deste pacote foi implementado três classes principais a primeira chamada “Calculadora.java” que é a responsável pelas operações básicas como soma, subtração, multiplicação e divisão com o tratamento de erro para divisão por zero, a segunda classe foi a “CalculadoraTeste” classe que contém os testes automatizados com o JUnit 5 cobrindo diversos cenários como testes de números positivos, negativos e zero, já a terceira e última classe chamada “MenuCalculadora.java” uma classe que fornece uma mini interface simples por via terminal com um menu interativo validação de entrada e formatação de saída.

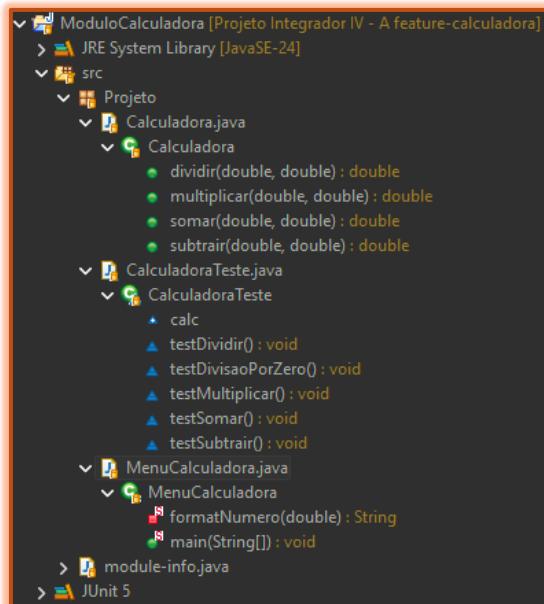


Figura 2: Estrutura do Projeto Organizado

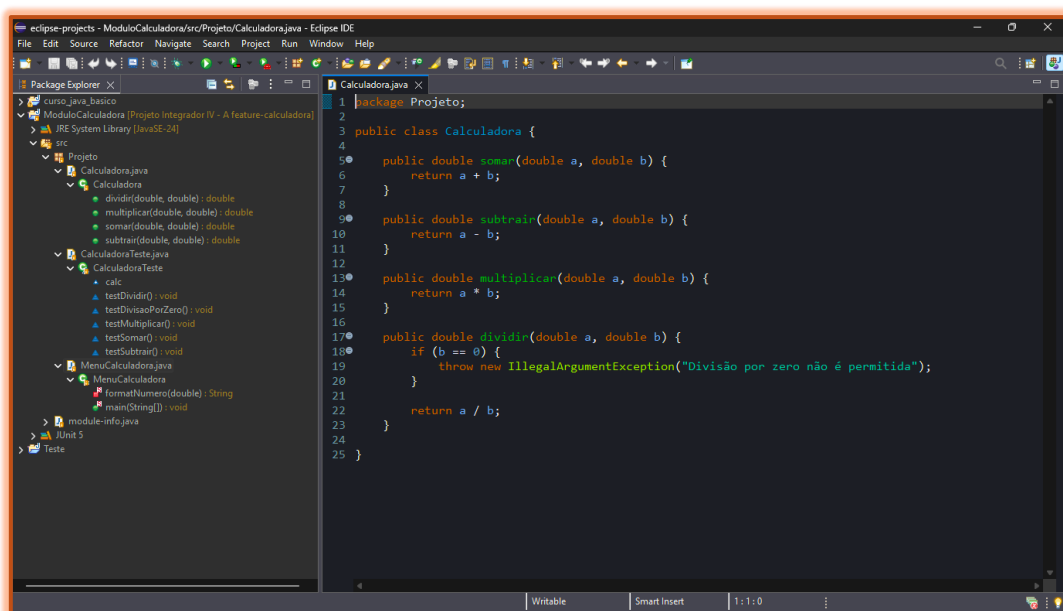


Figura 3: Classe Java Calculadora.java

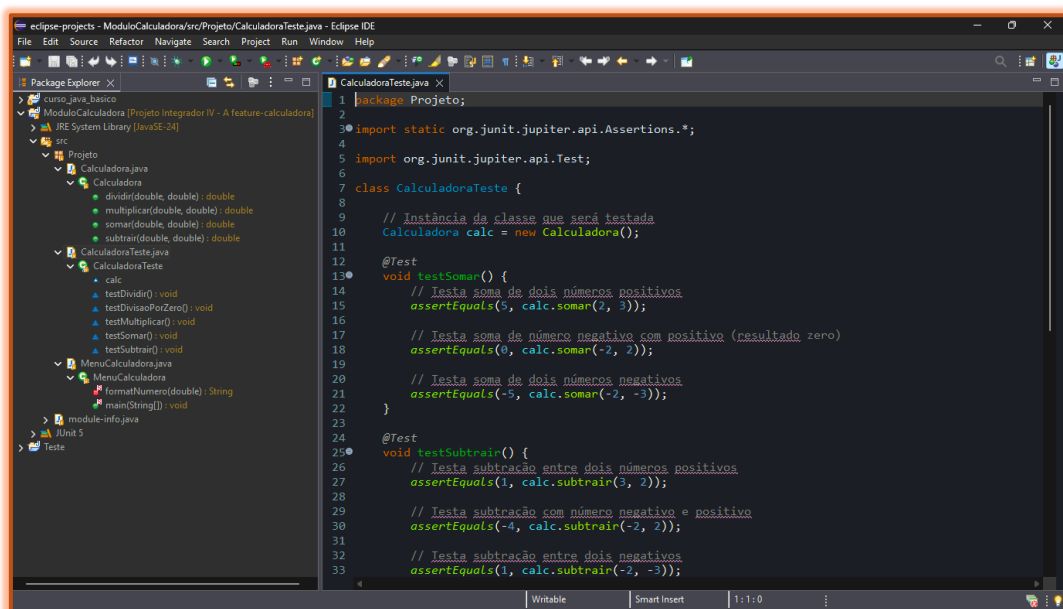


Figura 4: Classe Java CalculadoraTeste.java

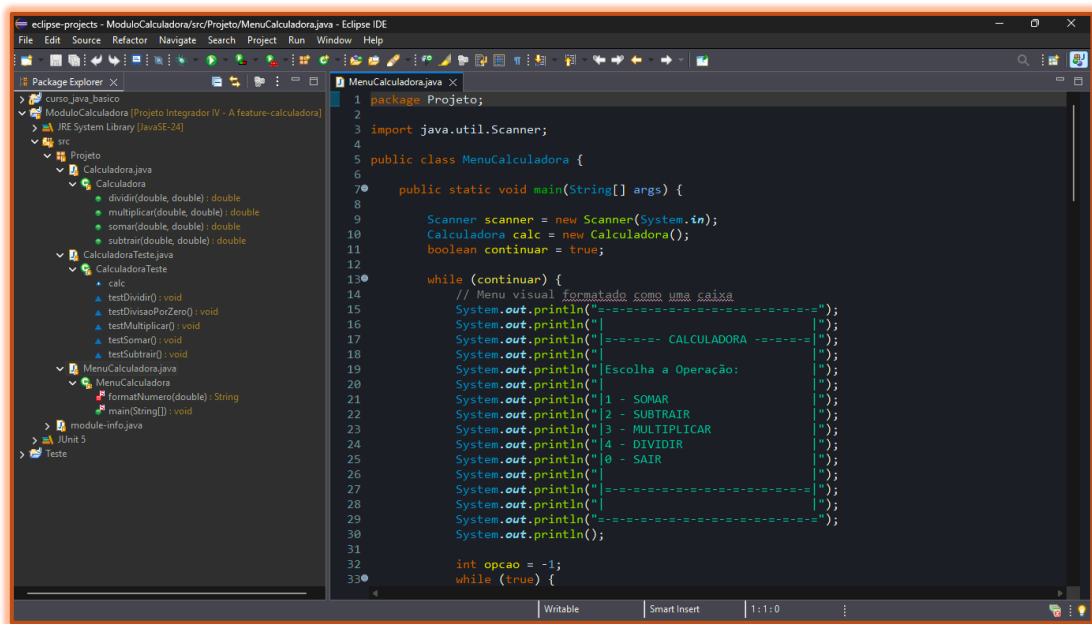


Figura 5: Classe Java MenuCalculadora.java

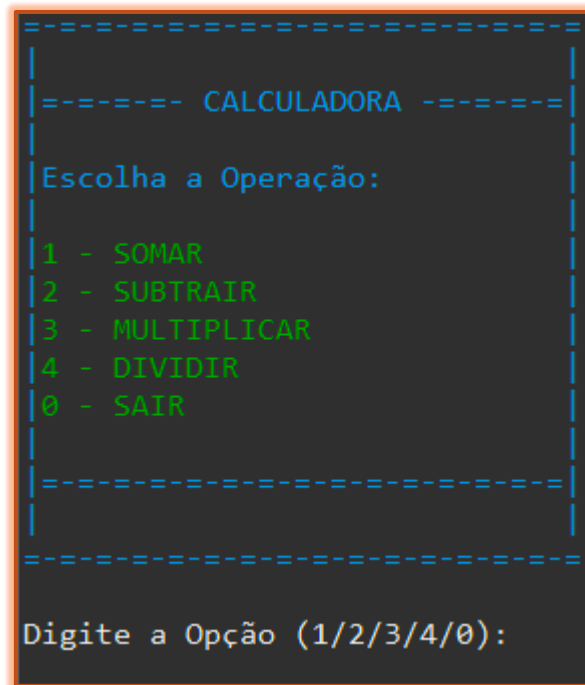


Figura 6: Menu Via Terminal

3.3. Pastas e Estrutura Geral do Projeto

Além do código-fonte feito pelo Eclipse, o projeto foi organizado de maneira bem organizada em uma pasta principal chamada Projeto Integrador IV – A, dentro dela foram criadas as seguintes subpastas Modulo Calculadora contendo o projeto em Java e as classes, a outra pasta foi a Imagens do Projeto pasta destinada a prints de tela ao longo do projeto, e a outra pasta Documentação pasta criada para estar armazenado este relatório e os demais arquivos e por último outra subpasta chamada apresentação do projeto aonde que mostra o vídeo apresentando o projeto, essa organização facilita pastas e subpastas ao encontrar algum arquivo específico.

4. ESTRUTURA DO CÓDIGO-FONTE

A aplicação Java foi estruturada com base em três classes principais todas organizadas dentro do pacote chamado Projeto no projeto Java “ModuloCalculadora” foi feita a modularização para não haver dificuldades na hora de entender o código facilitando a reutilização do código e aplicação dos testes unitários a seguir serão descritas as classes e suas responsabilidades principais.

4.1. Classe Calculadora

Está e a classe principal responsável por executar as quatro operações matemáticas básicas como soma, subtração, multiplicação e divisão ela possui quatro métodos públicos a estrutura e clara e simples de entender.

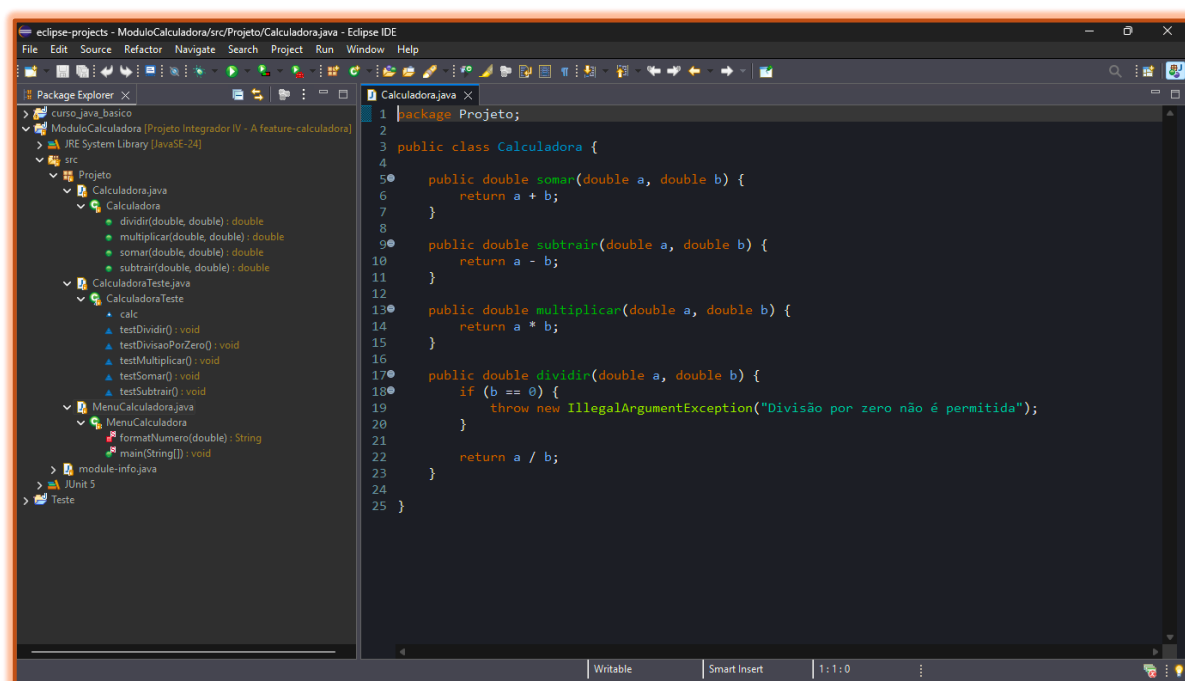


Figura 7: Classe Java Calculadora

4.2. Classe MenuCalculadora

A classe MenuCalculadora é responsável por fornecer uma mini interface via linha de terminal, uma interface de interação entre o usuário e a calculadora, utilizando o console por meio de entrada e saída ele apresenta uma aparência bonita com bordas, cores ANSI, e opções numeradas tornando uma experiência mais amigável mesmo sendo uma aplicação em modo de texto, o menu principal poderá escolher cinco opções como somar, subtrair, multiplicar e dividir ou sair do programa.

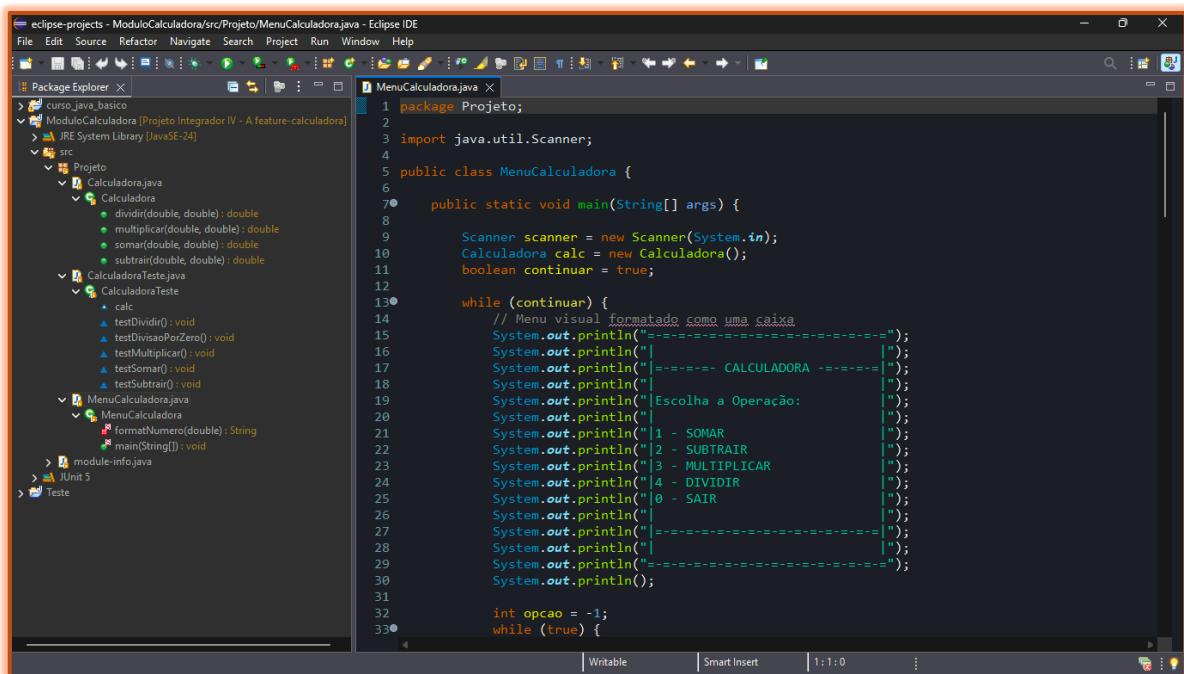


Figura 8: Classe Java MenuCalculadora

4.3. Classe CalculadoraTeste (JUnit)

Nesta classe CalculadoraTeste contém os testes automatizados da aplicação, utilizando a biblioteca JUnit 5, esses testes têm o principal objetivo e verificar o comportamento dos métodos da classe calculadora em diferentes situações, garantindo que os resultados fiquem com sucesso e se der algum erro que os erros sejam tratados de maneira certa, para cada método foi criado um teste para as operações.

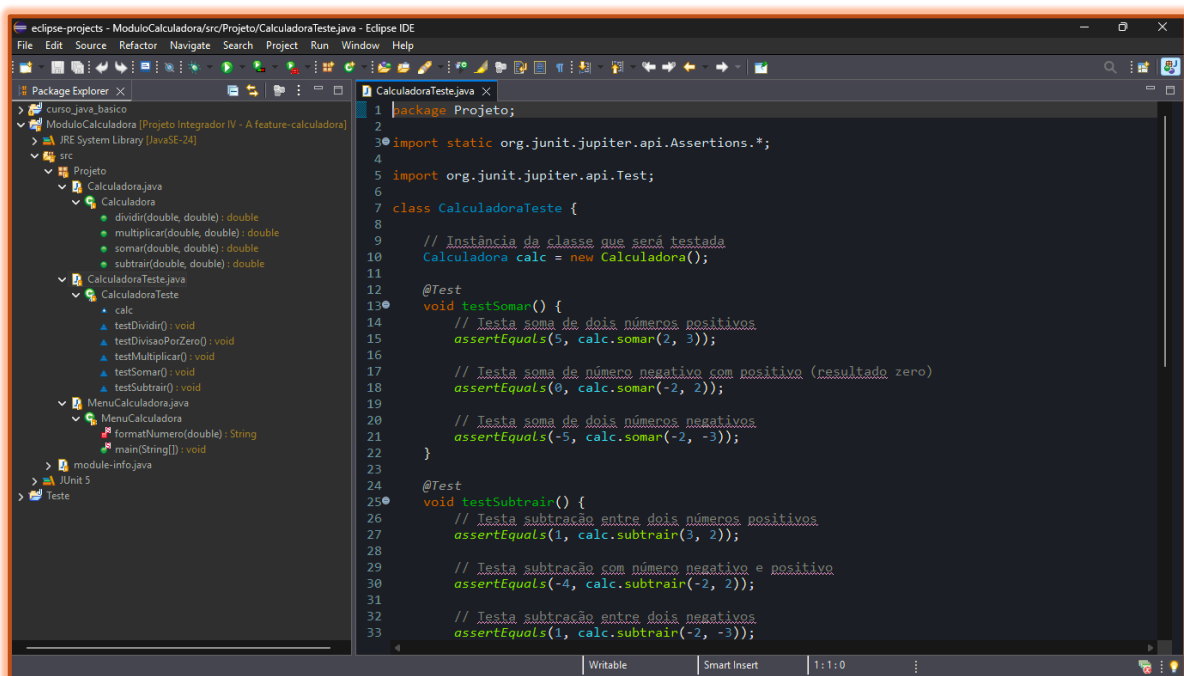


Figura 9: Classe Java CalculadoraTeste

5. FUNCIONALIDADES IMPLEMENTADAS

As principais funcionalidades do projeto consistem nas operações matemáticas básicas implementadas na classe calculadora, cada operação foi desenvolvida com um método separado, recebendo dois parâmetros do tipo double, e um resultado correspondente. A seguir são descritas as quatro funcionalidades principais.

5.1. Soma

A funcionalidade de soma foi implementada por meio do método somar(double a, double b) esse método realiza a adição simples entre dois valores recebidos, como parâmetro e retorna o resultado da operação a operação foi testada com diferentes cenários a soma entre números positivos soma entre números negativos soma de números envolvendo zeros.

5.2. Subtração

A operação de subtração é realizada pelo método subtrair(double a, double b) que retorna a diferença entre os dois valores passando como argumento nos testes foram considerados vários pontos como subtração de números positivos, negativos e zeros e situações envolvendo cada tipo diferente de número.

5.3. Multiplicação

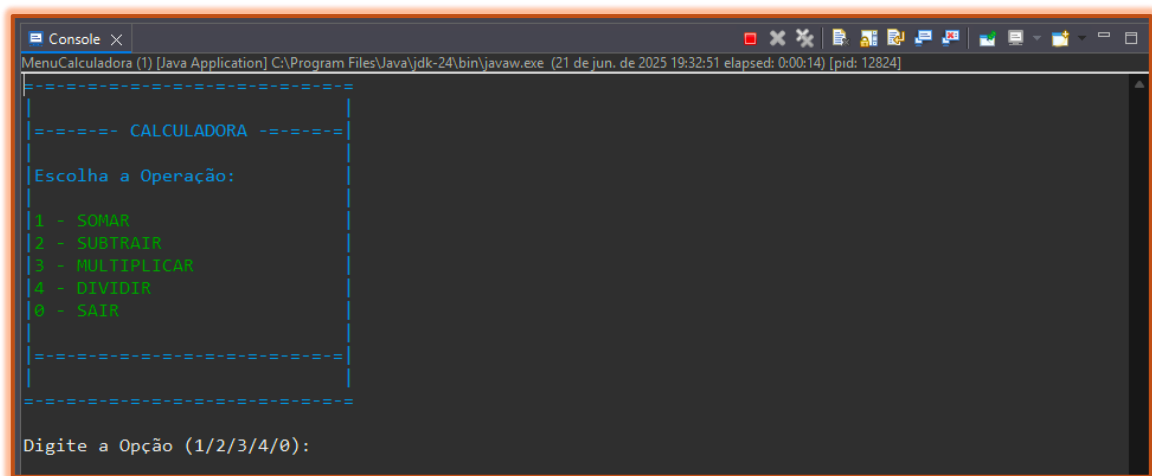
A multiplicação foi implementada com o método multiplicar(double a, double b) que vai retornar o produto entre dois valores fornecidos, durante os testes foram feitos o teste automatizados e diferentes cenários e foram testadas com números positivos negativo e zeros também.

5.4. Divisão

Na divisão é tratada o seguinte método dividir(double a, double b) que vai retornar o resultado da divisão do primeiro número, pelo segundo diferente de outras operações este método possui uma validação especial se o valor b for zero vai retornar uma exceção impedindo que não pode fazer a divisão por zero.

6. INTERFACE DO MENU INTERATIVO

A interface do sistema foi desenvolvida no estilo console modo texto através da classe MenuCalculadora, mesmo sendo uma aplicação via comando, foram aplicadas formatação visual com o uso de bordas centralizadas, alinhamento, utilização de cores ANSI tornando mais atrativo para os usuários, o menu exibe as seguintes operações como o de somar, subtrair, multiplicar, dividir e sair do programa. Totalmente após escolher umas das opções o usuário é solicitado a informar dois números o sistema permite que sejam inseridos números com virgulas além disso cada método permite que nas opções sejam digitadas s ou n tanto minúscula ou maiúscula foram utilizados laços de repetição, condicionais e dentre outros comandos específicos.



```
Console X
MenuCalculadora (1) [Java Application] C:\Program Files\Java\jdk-24\bin\javaw.exe (21 de jun. de 2025 19:32:51 elapsed: 0:00:14) [pid: 12824]

=====
|===== CALCULADORA =====|
|Escolha a Operação:         |
|1 - SOMAR                   |
|2 - SUBTRAIR                 |
|3 - MULTIPLICAR              |
|4 - DIVIDIR                  |
|0 - SAIR                     |
|=====                     |
|=====                     |
|
|Digite a Opção (1/2/3/4/0): |
|=====                     |
|=====                     |
```

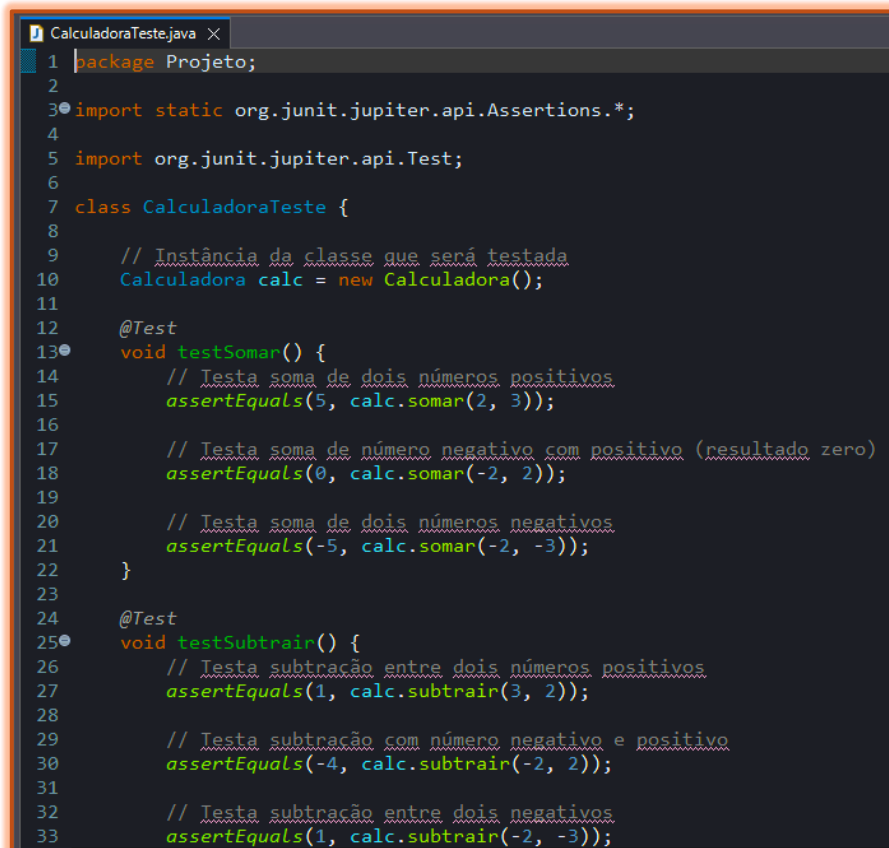
Figura 10: Menu pelo Terminal

7. TESTES AUTOMATIZADOS COM JUNIT 5

Para garantir a confiabilidade das operações matemáticas implementadas na classe calculadora, foram criados testes automatizados utilizando a biblioteca JUnit 5 esses testes são executados, para verificar se cada método retorna o resultado esperado tanto de sucesso quanto de erro. A seguir serão descritos os principais aspectos dos testes realizados.

7.1. Estrutura dos Testes

A classe dos testes foi nomeada CalculadoraTeste e contém cinco métodos principais de verificação cada uma das operações de calculadora somar, subtrair, multiplicar e dividir e tratamento de divisão por zero foi utilizada anotações @Test do JUnit 5 para identificar os métodos como testes e as asserções também utilizadas, esses testes ajudam a garantir que as funcionalidades estejam funcionando tudo em normal processo de maneira corretamente.



```

1 package Projeto;
2
3 import static org.junit.jupiter.api.Assertions.*;
4
5 import org.junit.jupiter.api.Test;
6
7 class CalculadoraTeste {
8
9     // Instância da classe que será testada
10    Calculadora calc = new Calculadora();
11
12    @Test
13    void testSomar() {
14        // Testa soma de dois números positivos
15        assertEquals(5, calc.somar(2, 3));
16
17        // Testa soma de número negativo com positivo (resultado zero)
18        assertEquals(0, calc.somar(-2, 2));
19
20        // Testa soma de dois números negativos
21        assertEquals(-5, calc.somar(-2, -3));
22    }
23
24    @Test
25    void testSubtrair() {
26        // Testa subtração entre dois números positivos
27        assertEquals(1, calc.subtrair(3, 2));
28
29        // Testa subtração com número negativo e positivo
30        assertEquals(-4, calc.subtrair(-2, 2));
31
32        // Testa subtração entre dois negativos
33        assertEquals(1, calc.subtrair(-2, -3));
34    }
35 }

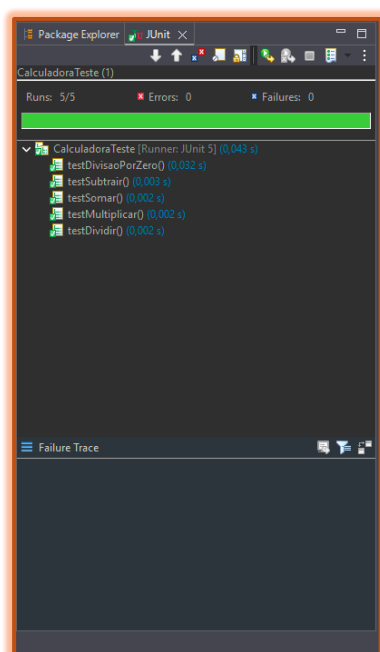
```

Figura 11: Classe Java CalculadoraTeste

7.2. Casos de Sucesso

Os casos de sucesso testam as operações matemáticas com diferentes tipos de entrada validando se os cálculos realmente retornam os valores corretos, por exemplo a soma verifica a soma de dois números e se retorna o resultado correto, a subtração também verifica se a subtração de dois números realmente retorna o resultado correto e todos os outros quanto a multiplicar verificando se o produto retorna correto e a divisão está sendo correta e verifica a exceção por zero.

7.3. Casos de Erro e Exceção



Além dos testes de sucesso foi implementado um teste específico para verificar o comportamento da divisão por zero, um cenário que pode causar falha em tempo de execução, se não for tratado corretamente o método dividir da classe calculadora foi projetado para lançar uma exceção sempre que o segundo parâmetro divisor for igual a zero.

Figura 12: Testes Automatizados com JUnit 5

8. CONTROLE DE VERSÃO COM GIT E GITHUB

Foi utilizada ferramentas como versionamento de códigos, ferramentas famosas no mercado de trabalho de TI, conhecidas como Git e GitHub, a ferramenta utilizada para digitar os comandos como fazer Branches, Merges, Commitar, Clonagem de repositórios, já o GitHub ferramentas online responsáveis por armazenar códigos fontes e utilizando estas duas ferramentas em conjunto o Git para digitar os comandos e o GitHub para disponibilizar os repositórios.

8.1. Criação do Repositório e Upload Inicial

Para armazenar o repositório versionar o projeto foi criado uma conta no GitHub e logo depois foi criado um repositório chamado Projeto_Integrador_IV-A o repositório é público tendo nele imagens, códigos, apresentação relacionado ao projeto.

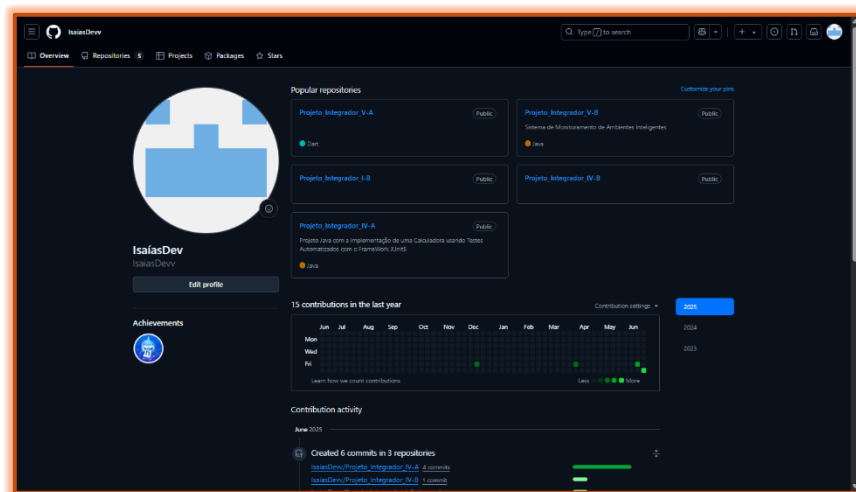


Figura 13: Perfil do GitHub e os Seguintes Repositórios

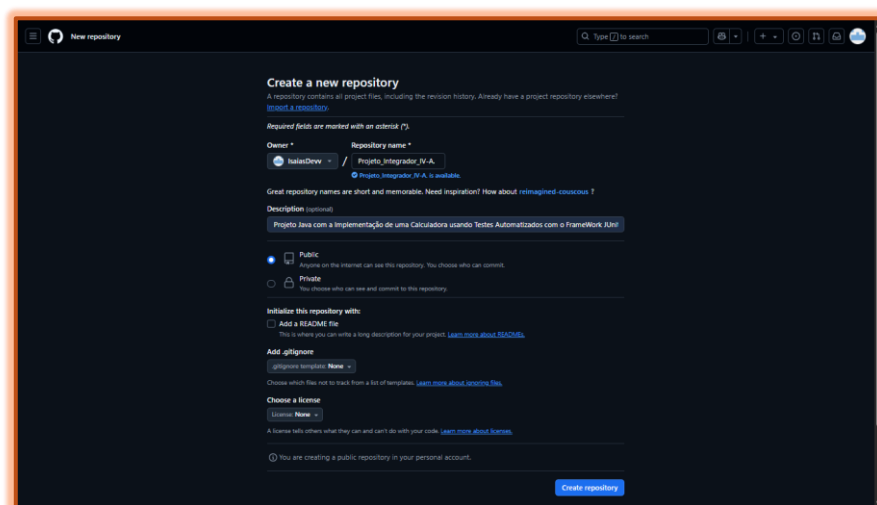


Figura 14: Criação do Repositório do Projeto

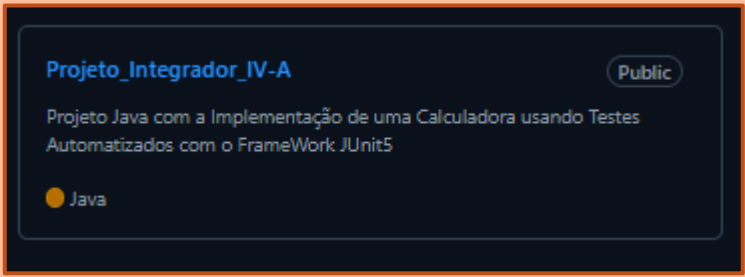


Figura 15: Repositório já Criado

8.2. Commits e Histórico de Versões

Para fazer a simulação de um desenvolvimento colaborativo foi criada uma Branch separada a partir da Branch principal main, a outra Branch foi chamada de feature-calculadora nessa nova Branch, foi desenvolvido o código do menu como implementação de cores do ANSI e bordas alinhadas, durante o desenvolvimento do projeto foram feitos diversos Commits para registrar o progresso e manter um histórico de alterações cada Commit feito com mensagem claras e objetivas descrevendo as modificações realizadas, criação da classe calculadora implementação das cores ANSI adição dos testes automatizados melhoria visuais na interface e tratamento de exceções.

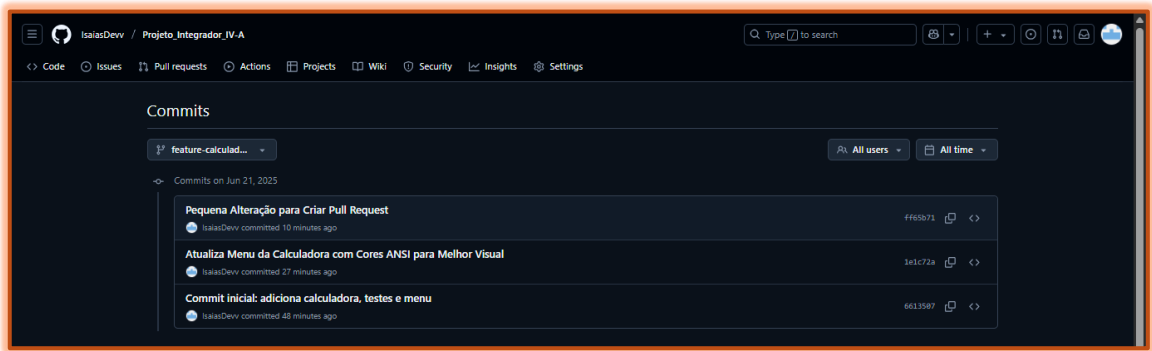


Figura 16: Histórico de Progresso dos Commits Realizados

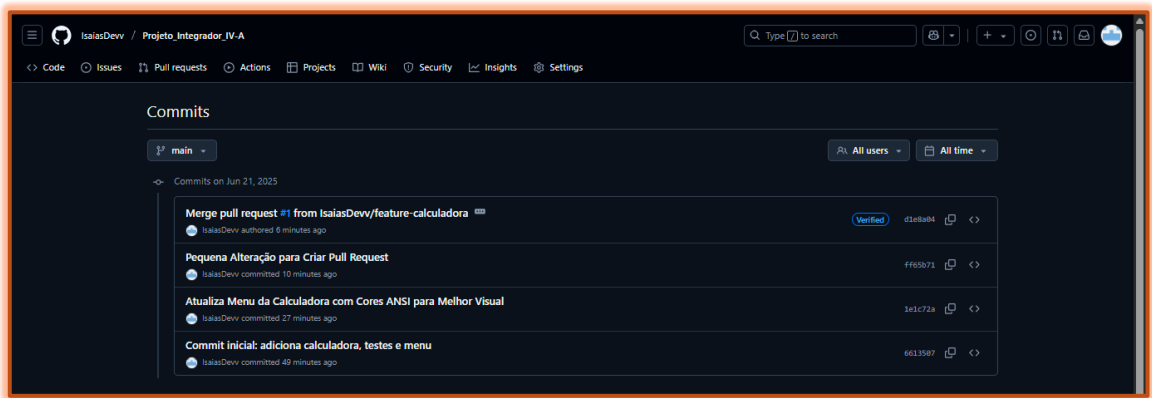


Figura 17: Histórico de Progresso dos Commits Realizados

8.3. Branchs e Merge

Após o término da implementação e testes na Branch feature-calculadora foi realizado o processo de merge para integrar a Branch principal main, o merge foi feito diretamente na interface do GitHub garantido que todas as alterações fossem revisadas antes da unificação, essa prática simula a rotina de equipes de desenvolvimento em que cada funcionalidade é criada em uma Branch separada e posteriormente unida na principal.

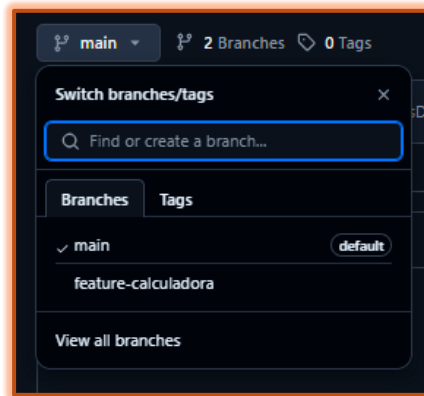


Figura 18: Branchs Criadas

9. CLONE DE REPOSITÓRIO EXTERNO

Como parte da proposta do projeto integrador foi realizado o clone de um repositório publico no GitHub com o objetivo de praticar o uso do Git em ambientes remotos e simular o reaproveitamento de código de terceiros, o repositório clonado foi selecionado livremente apenas para fins demonstrativos, o processo foi executado utilizando o comando Git Clone, ao finalizar o clone os arquivos do projeto foram baixados para a maquina local onde foi possível acessar, visualizar e até mesmo modificar o código caso desejado, esse processo também simula um ambiente de colaboração entre desenvolvedores.

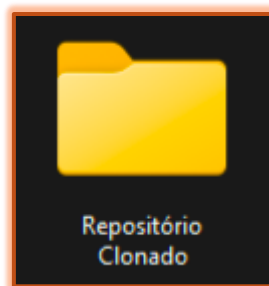


Figura 19: Pasta Criada para Clonar o Repositório

```
MINGW64/c/Users/icorr/OneDrive/Imagens/Repositório Clonado

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Imagens/Repositório Clonado
$ git clone https://github.com/loiane/curso-java-basico.git
Cloning into 'curso-java-basico'...
remote: Enumerating objects: 2996, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (22/22), done.
remote: Total 2996 (delta 13), reused 2 (delta 2), pack-reused 2972 (from 3)
Receiving objects: 100% (2996/2996), 10.63 MiB | 17.52 MiB/s, done.
Resolving deltas: 100% (1417/1417), done.

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Imagens/Repositório Clonado
$
```

Figura 20: Comando Sendo Aplicado para Clonar Repositório Git Clone

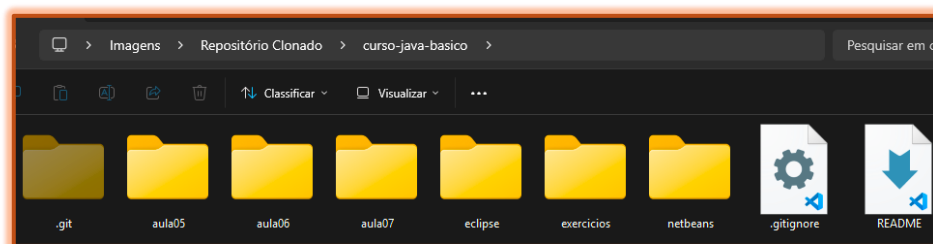


Figura 21: Depois de ter executado o Comando Cria as pastas do Repositório Original na Máquina do Usuário

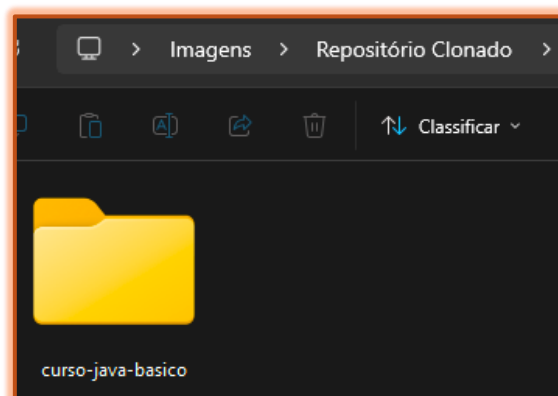


Figura 22: Pasta depois de Criada

10. ORGANIZAÇÃO DAS PASTAS DO PROJETO

Durante o desenvolvimento do Projeto Integrador IV – A foi adotada uma estrutura de pastas clara e organizada, para facilitar o gerenciamento dos arquivos e manter uma separação lógica entre o componente do projeto, abaixo está a descrição a organização adotada:

- Projeto Integrador IV - A

```

└─ ModuloCalculadora
|   └─ src/Projeto/
|       └─ Calculadora.java
|       └─ MenuCalculadora.java
|       └─ CalculadoraTeste.java
|       └─ .classpath
|       └─ .project
|
└─ Imagens do Projeto
|   └─ tela_menu.png
|   └─ execução_teste.png
|   └─ estrutura_git.png
|
└─ Documentação
|   └─ Projeto Integrador IV – A
  
```


11. PRINTS DE TELA DO PROJETO

```

MINGW64/c/Users/icorr/OneDrive/Documentos/Projeto Integrador IV - A

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A
$ git init
Initialized empty Git repository in C:/Users/icorr/OneDrive/Documentos/Projeto Integrador IV - A/.git/

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (master)
$ git add .

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (master)
$ git status
On branch master

No commits yet

Changes to be committed:
  (use "git rm --cached <file>..." to unstage)
        new file:   Modulo Calculadora/.classpath
        new file:   Modulo Calculadora/.project
        new file:   Modulo Calculadora/.settings/org.eclipse.core.resources.prefs
        new file:   Modulo Calculadora/.settings/org.eclipse.jdt.core.prefs
        new file:   Modulo Calculadora/bin/Projeto/Calculadora.class
        new file:   Modulo Calculadora/bin/Projeto/CalculadoraTeste.class
        new file:   Modulo Calculadora/bin/Projeto/MenuCalculadora.class
        new file:   Modulo Calculadora/bin/module-info.class
        new file:   Modulo Calculadora/src/Projeto/Calculadora.java
        new file:   Modulo Calculadora/src/Projeto/CalculadoraTeste.java
        new file:   Modulo Calculadora/src/Projeto/MenuCalculadora.java
        new file:   Modulo Calculadora/src/module-info.java

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (master)
$ git commit -m "Commit inicial: adiciona calculadora, testes e menu"
[master (root-commit) 6613507] Commit inicial: adiciona calculadora, testes e menu
12 files changed, 306 insertions(+)
create mode 100644 Modulo Calculadora/.classpath
create mode 100644 Modulo Calculadora/.project
create mode 100644 Modulo Calculadora/.settings/org.eclipse.core.resources.prefs
create mode 100644 Modulo Calculadora/.settings/org.eclipse.jdt.core.prefs
create mode 100644 Modulo Calculadora/bin/Projeto/Calculadora.class
create mode 100644 Modulo Calculadora/bin/Projeto/CalculadoraTeste.class
create mode 100644 Modulo Calculadora/bin/Projeto/MenuCalculadora.class
create mode 100644 Modulo Calculadora/bin/module-info.class
create mode 100644 Modulo Calculadora/src/Projeto/Calculadora.java
create mode 100644 Modulo Calculadora/src/Projeto/CalculadoraTeste.java
create mode 100644 Modulo Calculadora/src/Projeto/MenuCalculadora.java
create mode 100644 Modulo Calculadora/src/module-info.java

```

Figura 23: Comandos Feitos pelo Git para realizar Upload do Repositório

```

MINGW64/c/Users/icorr/OneDrive/Documentos/Projeto Integrador IV - A
create mode 100644 Modulo Calculadora/src/Projeto/MenuCalculadora.java
create mode 100644 Modulo Calculadora/src/module-info.java

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (master)
$ git remote add origin https://github.com/IsaiasDevv/Projeto_Integrador_IV-A.git

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (master)
$ git branch
* master

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (master)
$ git branch -M main

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (main)
$ git push -u origin main
Enumerating objects: 20, done.
Counting objects: 100% (20/20), done.
Delta compression using up to 8 threads
Compressing objects: 100% (19/19), done.
Writing objects: 100% (20/20), 8.00 KiB | 2.00 MiB/s, done.
Total 20 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
To https://github.com/IsaiasDevv/Projeto_Integrador_IV-A.git
 * [new branch]    main -> main
branch 'main' set up to track 'origin/main'.

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (main)
$ git checkout -b feature-calculadora
Switched to a new branch 'feature-calculadora'

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git branch
* feature-calculadora
  main

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git status
On branch feature-calculadora
Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Modulo Calculadora/bin/Projeto/MenuCalculadora.class
        modified:   Modulo Calculadora/src/Projeto/MenuCalculadora.java

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

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

```

Figura 24: Comandos Realizados no Git

```

MINGW64:/c/Users/icorr/OneDrive/Documentos/Projeto Integrador IV - A

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

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git add .
warning: in the working copy of 'Modulo Calculadora/bin/.gitignore', LF will be replaced by CRLF
the next time Git touches it

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git status
On branch feature-calculadora
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   Modulo Calculadora/bin/.gitignore
        modified:   Modulo Calculadora/bin/Projeto/MenuCalculadora.class
        modified:   Modulo Calculadora/src/Projeto/MenuCalculadora.java

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git commit -m "Atualiza Menu da Calculadora com Cores ANSI para Melhor Visual"
[feature-calculadora 1e1c72a] Atualiza Menu da Calculadora com Cores ANSI para Melhor Visual
3 files changed, 16 insertions(+), 9 deletions(-)
create mode 100644 Modulo Calculadora/bin/.gitignore

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git checkout main
Switched to branch 'main'
Your branch is up to date with 'origin/main'.

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (main)
$ git merge feature-calculadora
Updating 6613507..1e1c72a
Fast-forward
  Modulo Calculadora/bin/.gitignore      | 1 +
  .../bin/Projeto/MenuCalculadora.class  | Bin 5013 -> 5261 bytes
  .../src/Projeto/MenuCalculadora.java   | 24 ++++++-----
  3 files changed, 16 insertions(+), 9 deletions(-)
  create mode 100644 Modulo Calculadora/bin/.gitignore

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (main)
$ git push origin main
Enumerating objects: 18, done.
Counting objects: 100% (18/18), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (10/10), 3.88 KiB | 3.88 MiB/s, done.
Total 10 (delta 4), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/IsaiasDevv/Projeto_Integrador_IV-A.git
  6613507..1e1c72a  main -> main

```

Figura 25: Comandos feitos pelo Git Criação de Branches

```

MINGW64:/c/Users/icorr/OneDrive/Documentos/Projeto Integrador IV - A
6613507..1e1c72a  main -> main

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (main)
$ git checkout feature-calculadora
Switched to branch 'feature-calculadora'

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git push -u origin feature-calculadora
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-calculadora' on GitHub by visiting:
remote:   https://github.com/IsaiasDevv/Projeto_Integrador_IV-A/pull/new/feature-calculadora
remote:
To https://github.com/IsaiasDevv/Projeto_Integrador_IV-A.git
 * [new branch]   feature-calculadora -> feature-calculadora
branch 'feature-calculadora' set up to track 'origin/feature-calculadora'.

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git checkout feature-calculadora
Already on 'feature-calculadora'
Your branch is up to date with 'origin/feature-calculadora'.

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git status
On branch feature-calculadora
Your branch is up to date with 'origin/feature-calculadora'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Modulo Calculadora/src/Projeto/MenuCalculadora.class
        modified:   Modulo Calculadora/src/Projeto/MenuCalculadora.java

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

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git add .

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git commit -m "Pequena Alteração para Criar Pull Request"
[feature-calculadora ff65b71] Pequena Alteração para Criar Pull Request
2 files changed, 3 insertions(+)

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git push origin feature-calculadora
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 1.15 KiB | 1.15 MiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/IsaiasDevv/Projeto_Integrador_IV-A.git

```

Figura 26: Comandos feitos pelo Git Commits Realizados

```

MINGW64/c:/Users/icorr/OneDrive/Documentos/Projeto Integrador IV - A
Switched to branch 'feature-calculadora'

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git push -u origin feature-calculadora
Total 0 (delta 0), reused 0 (delta 0), pack-reused 0 (from 0)
remote:
remote: Create a pull request for 'feature-calculadora' on GitHub by visiting:
remote:   https://github.com/IsaiasDevv/Projeto_Integrador_IV-A/pull/new/feature-calculadora
remote:
To https://github.com/IsaiasDevv/Projeto_Integrador_IV-A.git
 * [new branch]      feature-calculadora -> feature-calculadora
branch 'feature-calculadora' set up to track 'origin/feature-calculadora'.

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git checkout feature-calculadora
Already on 'feature-calculadora'
Your branch is up to date with 'origin/feature-calculadora'.

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git status
On branch feature-calculadora
Your branch is up to date with 'origin/feature-calculadora'.

Changes not staged for commit:
  (use "git add <file>..." to update what will be committed)
  (use "git restore <file>..." to discard changes in working directory)
        modified:   Modulo Calculadora/bin/Projeto/MenuCalculadora.class
        modified:   Modulo Calculadora/src/Projeto/MenuCalculadora.java

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

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git add .

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git commit -m "Pequena Alteração para Criar Pull Request"
[feature-calculadora ff65b71] Pequena Alteração para Criar Pull Request
2 files changed, 3 insertions(+)

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ git push origin feature-calculadora
Enumerating objects: 17, done.
Counting objects: 100% (17/17), done.
Delta compression using up to 8 threads
Compressing objects: 100% (8/8), done.
Writing objects: 100% (9/9), 1.15 KiB | 1.15 MiB/s, done.
Total 9 (delta 5), reused 0 (delta 0), pack-reused 0 (from 0)
remote: Resolving deltas: 100% (5/5), completed with 5 local objects.
To https://github.com/IsaiasDevv/Projeto_Integrador_IV-A.git
 1e1c72a..ff65b71 feature-calculadora -> feature-calculadora

icorr@NOTEBOOK-GAMER MINGW64 ~/OneDrive/Documentos/Projeto Integrador IV - A (feature-calculadora)
$ |

```

Figura 27: Enviando as Informações para o GitHub

12. CONCLUSÃO

O desenvolvimento deste Projeto Integrador IV – A proporcionou uma importante oportunidade de aplicar na prática diversos conceitos fundamentais na área de análise e desenvolvimento de sistemas por meio de construção de um modulo calculadora Java, aplicar melhor sobre temas de programação orientada a objetos, e aplicar também testes automatizados usando o JUnit 5 como ferramenta auxiliadora, e também usando versionamento de código com a ferramenta Git e GitHub que possibilita um controle de versão de forma gerenciada e controlada e organizada permitindo fazer Commits Branches Merges e dentre outros comandos importantes, com isso este projeto permitiu adquirir mais conhecimentos e colocar em pratica tudo que foi visto em aula e vídeo aulas com os professor.

13. REFERÊNCIAS

Git: <https://git-scm.com/>

GitHub: <https://github.com/>

GitHub Desktop: <https://desktop.github.com/download/>

JUnit 5: <https://junit.org/>

IDE Eclipse: <https://eclipseide.org/>

Java: <https://www.oracle.com/br/java/>

