

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



PROPOSTA DO PROJETO INTEGRADOR IV-A

ALISSON RODRIGUES ALVES

GOIÂNIA,
2025

SUMÁRIO

1	OBJETIVO	3
2	DESENVOLVIMENTO	3
3	RESULTADOS	3
3.1	Etapas	4
4	CRONOGRAMA	4
5	SUGESTÕES BIBLIOGRÁFICAS	5

1 OBJETIVO

Desenvolvimento de um projeto Java, focado em testes automatizados (unitários e funcionais) e no gerenciamento de código utilizando práticas de controle de versão (versionamento) com GitHub. O objetivo é aplicar JUnit 5 para garantir a qualidade do código e configurar um fluxo de trabalho de desenvolvimento colaborativo usando GitHub com práticas de branching e merge. Desta forma, os objetivos específicos são:

- Aplicar testes unitários e funcionais automatizados utilizando JUnit 5, conceitos de verificação e validação de software.
- Configurar e gerenciar um repositório Git, implementando práticas de branching e merge.
- Simular o desenvolvimento colaborativo, aplicando boas práticas de controle de versão e qualidade de software.

2 DESENVOLVIMENTO

O trabalho envolve a criação de funcionalidades, aplicação de testes automatizados (unitários e funcionais) com JUnit, e o envio do código para um repositório remoto no GitHub. Sendo assim, você deverá desenvolver um módulo simples de software, como uma **calculadora** (incluindo as quatro operações como métodos, a saber: adição, subtração, multiplicação e divisão).

Para cada funcionalidade implementada, criar também os testes unitários com JUnit cobrindo diferentes cenários de uso (incluindo casos de sucesso e de falha). Dica:

- Somar: Testar a soma de números positivos, negativos e zero.
- Subtrair: Testar a subtração de números positivos, negativos e zero.
- Multiplicar: Testar a multiplicação de números positivos, negativos e zero.
- Dividir: Testar a divisão de números positivos, negativos e divisão por zero (esperar exceção).

Após, você deverá subir o seu projeto Java para o GitHub. Veja na seção Etapas a descrição mais detalhada sobre esses passos.

3 RESULTADOS

O entregável desta atividade deve conter o link do Github, o projeto calculadora com a classe Calculadora e a(as) classe(s) de testes, um arquivo de texto com as devidas explicações e os prints de telas exibindo que os seguintes pontos foram realizados:

- Implementação da classe calculadora.
- Implementação da classe de testes, vinculada ao JUnit 5.
- Criação da conta no GitHub;
- Criação de pelo menos do repositório na conta criada;
- Criação de pelo da branch (ramificação) do projeto calculadora;

- Clone de algum repositório preexistente do GitHub (escolha livre);
- Realizar commits de projeto calculadora com o código-fonte.
- Realizar a fusão (merge) de arquivos de branches diferentes no próprio projeto;

Além disso, o entregável deverá conter um vídeo (áudio + imagem) de demonstração dos itens solicitados. Grave a explicação de cada um dos itens solicitados como se estive explicando para um novo desenvolvedor da equipe recém-chegado na empresa. Dica para a gravação de vídeo: se utilizar o Windows 10 ou 11, nativamente use o atalho das teclas “wiki + G” para abrir a ferramenta de gravação do Windows, mas outros aplicativos de gravação de tela podem ser utilizados.

**** Atividade individual.**

3.1 Etapas

1. Configuração Inicial do Repositório GitHub:

- Criar uma conta do GitHub caso não possua.
- Cada aluno deve criar um repositório privado no **GitHub**.
- O repositório deve conter o código base do projeto, já configurado com **JUnit** para testes automatizados.

2. Criação de uma Branch de Funcionalidade:

- No repositório GitHub, o aluno deve criar uma branch da funcionalidade criada.

3. Desenvolvimento da Funcionalidade:

- O aluno deverá desenvolver a nova funcionalidade (ex: uma calculadora) dentro da branch criada.
- Fazer commits regulares durante o desenvolvimento para manter o histórico de progresso.

4. Testes Automatizados com JUnit:

- Durante o desenvolvimento, o aluno deve implementar **testes unitários e funcionais** para garantir que a funcionalidade esteja funcionando corretamente.
- Utilizar o **JUnit** para testar todas as partes relevantes do código.

5. Envio do Código para o GitHub:

- Após finalizar a implementação e os testes, o aluno deve enviar o código da branch para o repositório no **GitHub**.
- O aluno deve garantir que todos os arquivos relevantes (código fonte e testes) sejam incluídos no commit.

4 CRONOGRAMA

Data	Atividade
23/06/2025	Entrega do Projeto Integrador.

5 SUGESTÕES BIBLIOGRÁFICAS

Github. Disponível em: <https://github.com/>. Acesso em: 22 out. 2024.

Github Desktop. Disponível em: <https://desktop.github.com/> Acesso em: 22 out. 2024.

Para saber mais sobre o GitHub, pesquise sobre o vídeo com o seguinte título na internet:

“Como usar Git e GitHub da Forma mais Fácil Possível”, do canal Hashtag Programação. Esse vídeo possui aproximadamente 40 minutos de duração.