

GESTÃO E QUALIDADE DE SOFTWARE – GQS

Lab01

Prof. Calvetti



Gestão e Qualidade de Software

Tópicos:

- Bibliografia Básica;
- Bibliografia Complementar;
- Projeto A3;
- Fechamento.

Gestão e Qualidade de Software

Bibliografia Básica:

- PRESSMAN, Roger; MAXIM, Bruce. Engenharia de Software. Uma abordagem profissional. 8a. Ed. Bookman, 2016.
Parte III – Gestão da qualidade; Capítulo 19 – Conceitos de qualidade, pg. 412
<https://integrada.minhabiblioteca.com.br/#/books/9788580555349/cfi/3!/4/2@100:0.00>
- SOMMERVILLE, Ian. Engenharia de Software. 9. ed. São Paulo: Pearson Prentice Hall, 2011.
<https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca&page= 14§ion=0#/legacy/276>
- GONÇALVES, Priscila de Fátima.; BARRETO, Jeanine dos Santos.; ZENKER, Aline Maciel.; FAGUNDES, Rubem. Testes de software e gerência de configuração. Soluções Educacionais Integradas, 2019.
<https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca&page= 14§ion=0#/legacy/276>

Gestão e Qualidade de Software

Bibliografia Complementar:

- PFLEEGER, Shari Lawrence. Engenharia de software: teoria e prática. 2. ed. São Paulo: Prentice Hall, 2004.
<https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca#/legacy/106>
- BRAGA, Pedro Henrique Cacique. Teste de Software. São Paulo: Pearson Education, 2016.
<https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca#/legacy/150962>
- GALLIOTTI, Giocondo MARINO. Qualidade de Software. São Paulo: Pearson Education, 2016.
<https://bv4.digitalpages.com.br/?term=engenharia%2520de%2520software&searchpage=1&filtro=todos&from=busca#/legacy/124148>
- TONINI, Antonio Carlos. Métricas de Software. 2004.
<https://slideplayer.com.br/slide/108276/>

Gestão e Qualidade de Software

O que é...

- Projeto A3?

Projeto A3:

- Tema:

Refatoração de Código e Boas Práticas do *Clean Code*;

- Objetivo:

Este trabalho tem como objetivo avaliar a capacidade dos alunos de compreender e aplicar os princípios do *Clean Code* por meio da refatoração de um código-fonte legado. O foco será melhorar a legibilidade, manutenibilidade e eficiência do código sem alterar sua funcionalidade.

- Descrição:

- Os alunos devem escolher um código-fonte legado com diversas más práticas de programação;
- O código pode ser criado pelos próprios alunos, ou reutilizado de semestres anteriores, ou um código que possa ser utilizado e publicado em repositórios de códigos públicos;
- Os alunos deverão identificar problemas e realizar uma refatoração utilizando as melhores práticas de código limpo;
- Idealmente, o código-fonte deve ser (ou se tornar) orientado a objeto e o grupo pode pesquisar e implementar pelo menos um *Design Pattern* nesse código.

Continua...

Projeto A3 (continuação):

- Critérios de Avaliação:

- **Legibilidade:** Código mais organizado, nomes de variáveis e funções claros e significativos;
- **Estrutura:** Redução de repetições, modularização adequada e menor complexidade;
- **Comentário e Documentação:** Comentários explicativos apenas quando necessário e apropriada documentação;
- **Boas práticas:** Aplicação de princípios SOLID, DRY, KISS e YAGNI quando aplicável;
- **Testes:** Geração de testes unitários para garantir que o código continua funcionando corretamente após a refatoração;
- **Versionamento:** O código-fonte deve estar em um repositório *gitHub* que possa ser acessado publicamente e precisa ter registrado atualizações entre os meses de setembro e novembro.

Continua...

Projeto A3 (continuação):

- Entregas:

1. Os alunos deverão entregar um relatório em PDF até 27/11/2025 contendo:

- i. A descrição das deficiências do código identificadas, acompanhadas pelo *link* do repositório *GitHub* público com o código original, para uso do grupo todo. Esse repositório precisa ser criado e alimentado com a versão inicial do código até 30/09/2025;
- ii. As justificativas para as mudanças feitas no código, acompanhadas pelo *link* do mesmo repositório *GitHub* público com o código refatorado e com *commits* a partir de 01/10/2025 até 27/11/2025. Cada aluno do grupo está obrigado a fazer, pelo menos, um dos *commit* em seu nome;
- iii. A descrição dos testes unitários implementados, acompanhadas pelo *link* no mesmo repositório citado em i e ii. No repositório deve haver uma pasta */testes* com os testes unitários elaborados pelo grupo;
- iv. Conclusão sobre a importância do *Clean Code* na manutenção de *software*.

Continua...

Projeto A3 (continuação):

- Entregas:

2. Os alunos deverão entregar uma apresentação sobre o trabalho até 27/11/2025 contendo:

i. No padrão PPTX, 5 (cinco) *slides* resumindo o que foi feito no trabalho e como foi a experiência de fazer o trabalho em grupo, além de uma conclusão sobre a importância do *Clean Code* na manutenção d *software* escolhido.

3. Os alunos deverão apresentar presencialmente o trabalho, em sala, na aula de 28/11/2025, a partir das 19h:

i. Baseada no PPTX entregue no item 2.i, com no máximo 5 (cinco) minutos, sobre o trabalho realizado;

ii. Todos os alunos do grupo deverão estar presentes e apresentarem pessoalmente sua contribuição ao trabalho para os espectadores.

Continua...

Projeto A3 (continuação):

- Avaliações:

1. Todos os artefatos produzidos pelo grupo e entregues no seu repositório *GitHub* público, até o prazo limite estipulado:

- i. Até 30 pontos (em grupo), dos 40 pontos da A3 possíveis.

2. Apresentação presencial em sala de aula:

- i. Até 10 pontos (individual), dos 40 pontos da A3 possíveis.

Gestão e Qualidade de Software

Fechamento:

- Resumo;
- Problemas e Pontos a Ponderar; e
- Leituras e Fontes de Informação Complementares.

Obrigado!

