



## **Trabalho A3 – Teste de Software** **UC Gestão e Qualidade de Software**

**Professor Thiago Dotto**

### **Descrição:**

Modelagem de sistema para a prefeitura de Salvador. O domínio do sistema será escolhido pela equipe e deve ser aprovado pelo professor.

OBS: Alunos que tiverem aulas de Eng. de Software e o trabalho da A3 for diferente dessa proposta, podem aproveitar o mesmo trabalho, alinhar com antecedência com o Professor.

### **Objetivos:**

O objetivo do trabalho é oferecer a oportunidade dos alunos da UC de Gestão de Qualidade de Software praticarem a especificação de requisitos de software, testes de software, análise de ferramentas e implementação de Casos de Teste.

### **Data:**

O trabalho será apresentado conforme cronograma da disciplina.

### **Avaliação:**

O trabalho será dividido em três entregas

1. Engenharia de Requisitos: construção do documento de especificação de requisitos de sistema e casos de uso (funcionais e não funcionais);
2. Especificação de testes: construção do documento de especificação de testes do sistema conforme modelo enviado pelo professor e apresentação em sala de aula. Escrita de pelo menos **10 casos** de teste;
3. Estudo de uma ferramenta de teste de software (ver sugestões a seguir), implementação de pelo menos 3 casos de teste e apresentação em sala de aula;

O trabalho é referente a avaliação A3 que vai ter sua pontuação dividida em:

- 20 pontos, referentes as entregas 1 e 2 e 3 desse trabalho.
- 15 pontos vão ser de atividades em sala de aula.
- 05 pontos com o professor da aula teórica.

Compondo os 40 pontos totais para a entrega da A3.

### **Grupos:**

O trabalho deverá ser desenvolvido em grupos de no mínimo 5 e no máximo 6 alunos.

### **Cronograma:**

21/03/24: Criação dos grupos, escolha e apresentação dos eixos temáticos para discussão com as equipes;

18/04/24: 1ª entrega:

16/05/24: 2ª entrega:

13/06/24: 3ª entrega:

**1ª Entrega:** Configuração do Ambiente DevOps com GitActions e Planejamento

Objetivos:

- Especificação de Requisitos



## **Trabalho A3 – Teste de Software**

### **UC Gestão e Qualidade de Software**

- Configurar um ambiente de desenvolvimento e operações (DevOps) com ênfase na integração contínua usando GitActions. Planejar o projeto de software, incluindo estimativas e estratégias de desenvolvimento da pipeline.

#### **Tarefas:**

Configurar um ambiente de integração contínua utilizando GitHub Actions, automatizando fluxos de trabalho de CI/CD.

Implementar a validação de conventional commits no GitHub Actions para garantir padrões de commits.

Integrar a geração automática de versão semântica com base nos conventional commits.

#### **Entregáveis:**

Entrega do documento de Especificação de Requisitos

Configuração do GitHub Actions, regras de conventional commits, plano de projeto com estratégias de testes unitários.

#### **2ª entrega:**

##### **Objetivos:**

- Especificação de testes: construção do documento de especificação de testes do sistema conforme modelo enviado pelo professor e apresentação em sala de aula. Escrita de pelo menos **10 casos** de teste
- Desenvolver uma versão inicial do software.
- Aplicar conceitos de testes unitários e de integração, utilizando Mocks.

##### **Tarefas:**

- Implementar funcionalidades chave do software.
- Desenvolver e executar testes unitários e de integração, aplicando TDD (Test-Driven Development) e utilizando Mockito para mocks.

##### **Entregáveis:**

Código-fonte, testes unitários e de integração com mocks.

#### **3ª entrega:**

##### **Objetivos:**

- Apresentação da ferramenta de teste e dos casos de teste implementados;
- Implementar testes de validação e sistema.
- Aplicar técnicas de garantia de qualidade e revisão.

##### **Tarefas:**

- Realizar testes de caixa branca e caixa preta.
- Aplicar medidas, métricas e indicadores para avaliar a qualidade do software.
- Conduzir revisões de código e testes automatizados.

##### **Entregáveis:**



## **Trabalho A3 – Teste de Software**

### **UC Gestão e Qualidade de Software**

Relatório de testes, métricas de qualidade.

#### **Cenário:**

A prefeitura de Salvador precisa construir um sistema para auxiliar a população. Foram selecionados grande eixos temáticos para a apresentação de soluções que visem auxiliar na escolha desse sistema.

Eixos Temáticos:

- Trânsito
- Saúde
- Educação
- Emprego e Renda
- Esportes
- Cultura
- Ordem pública e manutenção da cidade
- Turismo

**Cada grupo deve selecionar uma das ferramentas sugeridas:**

- **JMeter**
- **Selenium**
- **Cypress**
- **Cucumber**
- **TestingWhiz (Community version)**
- **etc**

#### **Observações Gerais**

- Cada equipe deverá escolher um eixo temáticos e discutir com o professor os cenários para cada solução dentro do eixo escolhido.
- O trabalho deve ser desenvolvido seguindo os *templates* disponibilizados pelo professor e postado no Ulife (basta 1 componente do grupo postar). O trabalho deve ser apresentado em sala de aula nas datas programadas.
- A ferramenta de teste, assim como os tipos de testes, que a equipe for apresentar deverão ser discutidos e aprovados pelo professor.
- Integração Contínua com GitActions: Os alunos aprenderão a configurar e utilizar o GitHub Actions para automatizar testes, builds e deploy, garantindo um fluxo de trabalho eficiente e contínuo.
- Conventional Commits e Versionamento Semântico: A prática de conventional commits será usada para padronizar mensagens de commit, facilitando a geração automática de versão semântica e melhorando a rastreabilidade e a organização do projeto.
- Mocks com Mockito: A biblioteca Mockito será utilizada para criar mocks eficientes em testes unitários, permitindo testar comportamentos específicos do código de forma isolada.



## **Trabalho A3 – Teste de Software**

### **UC Gestão e Qualidade de Software**

- Colaboração e Gestão de Configuração: O trabalho será realizado em equipes, usando ferramentas de controle de versão como Git.
- Manutenção e Reengenharia: Aspectos de manutenção de software e potenciais necessidades de reengenharia devem ser considerados durante o desenvolvimento.
- Melhoria Contínua: Os alunos são encorajados a refletir sobre as práticas de melhoria do processo de software, incluindo referências a CMMI e MPS.BR.
- Feedback Contínuo: Será promovido um ambiente de feedback contínuo entre as equipes e o instrutor.

***Boa Sorte!***