



Teste de Software

Professor: Euclides Paim
euclidespaim@gmail.com



Apresentação da Disciplina

Professor: Euclides Paim
euclidespaim@gmail.com



Apresentação da Disciplina

Informações Gerais

- Aulas expositivas, teóricas em **sala de aula** e práticas em **laboratório**.
- Serão utilizadas as **apresentações em slides** do professor como **material principal** (base da sequência da aula);
- O aluno deverá **manter material para anotação** (caderno) e informações adicionais que não estão nos slides (isto irá ocorrer);
- O aluno deve **praticar e aprofundar, o conhecimento** desenvolvido em aula, através de pesquisas e estudos complementares (internet, biblioteca, vídeos).
- A **participação em aula** contará para a composição da nota.
- No desenvolvimento do assunto haverá explicações para responder dúvidas, também serão feitos exercícios teóricos e práticos de fixação do conteúdo;
- As notas de avaliações serão divulgadas periodicamente.
- **Trabalhos entregues com atraso podem não ser considerados**, ou terão nota reduzida conforme critério do professor;
- Observar as **boas práticas de convivência** e atenção às atividades de aula.



Apresentação da Disciplina

Metodologia

Metodologia:

- **Aulas Expositivas:** Para introdução teórica de cada tópico.
- **Atividades Práticas:** Criação de diagramas UML, modelagem de processos, e prototipagem utilizando ferramentas específicas.
- **Estudos de Caso:** Discussão e análise de exemplos reais para aplicação dos conceitos.
- **Projetos em Equipe:** Desenvolvimento colaborativo de modelos de sistemas, promovendo trabalho em grupo.





Apresentação da Disciplina

Sistema de Avaliação

- **Nota Semestral**

- Composta por **2 Provas** e **4 Listas de exercícios**.
Sendo listas teóricas e práticas (podendo sofrer alterações).
- As notas serão calculadas da seguinte forma:

$$\text{Nota 1} = \frac{(\text{NotaProva} \times 5) + (\text{Lista1} \times 2) + (\text{Lista2} \times 3)}{10}$$

$$\text{Nota Final} = \frac{\text{Nota 1} + \text{Nota 2}}{2}$$

Aprovação: 75% de presença e Média Final \geq 6,0

OBS: Todas as avaliações têm o valor 10.0. A forma de avaliação pode ser modificada de acordo com o andamento das aulas e nível de aprendizado da turma.



Apresentação da Disciplina

Sistema de Avaliação

- **Provas**

- Serão passados aos alunos exercícios/trabalhos a título de preparação para as provas;
- O aluno que resolver as listas de exercícios sem ajuda externa estará preparado para a prova, por isso essas devem ser resolvidas individualmente.
- As questões de prova serão inspiradas pelas listas de exercícios.

- **Listas de exercícios**

- Será incentivado que cada aluno resolva individualmente cada lista.
- Não será tolerada qualquer tipo de cópia. Caso seja detectada, o aluno receberá nota zero;
- As listas devem ser entregues no prazo, atividades atrasadas devem ser devidamente justificadas;
- Qualquer dúvida ou problema na correção das listas, trabalhos, ou na atribuição de frequências será devidamente tratado, sem prejuízo do aluno.



Apresentação da Disciplina

Ementa

Objetivo Geral: Planejar e desenvolver protótipos de testes para as diversas situações e fases de desenvolvimento do software, fornecendo uma visão geral da área de verificação, validação e teste de software; Compreender a importância e a necessidade da sua aplicação em projetos reais.



Apresentação da Disciplina

Ementa

Habilidades: Apropriar-se de conceitos, tipos, funções e forma correta das técnicas de teste. Interpretar e aplicar corretamente os manuais relacionados ao software.

Apresentação da Disciplina

Introdução





Apresentação da Disciplina

Programa da Disciplina

- **Introdução ao Teste de Software:**

- Conceitos e importância dos testes de software.
- Diferença entre testes manuais e automatizados.
- Tipos de testes: unitário, de integração, funcional, de aceitação, de sistema.
- Ciclo de vida do teste de software: planejamento, execução e análise de resultados.

- **Planejamento de Testes:**

- Definição de objetivos e escopo dos testes.
- Elaboração de planos de testes: objetivos, critérios de entrada e saída.
- Identificação de requisitos de teste e critérios de sucesso.
- Ferramentas para planejamento de testes.

- **Testes de Unidade (Unit Testing):**

- Definição de testes unitários e sua importância.
- Criação de testes unitários para funções/métodos.
- Uso de frameworks de testes: JUnit (Java), NUnit (.NET), PyTest (Python).
- Testando código isolado e simulações (mocking).



Apresentação da Disciplina

Programa da Disciplina

- **Testes de Integração:**
 - Definição de testes de integração e sua importância.
 - Testando a interação entre módulos ou componentes.
 - Técnicas de integração contínua e testes automatizados.
 - Ferramentas para testes de integração: Jenkins, Travis CI.
- **Testes Funcionais e de Interface:**
 - Testes funcionais: objetivo e metodologias.
 - Ferramentas para automação de testes de interface: Selenium, Cypress.
 - Criação de scripts de teste para interfaces gráficas.
 - Testes de compatibilidade de navegadores.
- **Testes de Performance e Carga:**
 - Definição de testes de performance e carga.
 - Ferramentas para testes de carga: JMeter, LoadRunner.
 - Medição de tempo de resposta e capacidade de tráfego.
 - Análise de resultados e identificação de gargalos.



Apresentação da Disciplina

Programa da Disciplina

- **Testes de Aceitação (UAT - User Acceptance Testing):**
 - O que é o teste de aceitação e como ele se aplica no ciclo de vida do software.
 - Colaboração com os stakeholders para definir os critérios de aceitação.
 - Como documentar e validar os testes de aceitação.
 - Simulações de testes com usuários reais ou fictícios.
- **Testes Automatizados:**
 - Introdução aos testes automatizados.
 - Vantagens e desvantagens da automação de testes.
 - Implementação de testes automatizados em diferentes camadas da aplicação.
- **Análise de Defeitos e Relatórios de Testes:**
 - Identificação e documentação de falhas durante os testes.
 - Classificação de defeitos: severidade, prioridade.
 - Como escrever relatórios de teste eficientes.
 - Ferramentas para rastreamento de defeitos: JIRA, Bugzilla, Redmine.



Apresentação da Disciplina

Programa da Disciplina

- **Cultura de Qualidade e Melhoria Contínua:**
 - A importância da qualidade de software no ciclo de vida do desenvolvimento.
 - Processos ágeis e testes: Scrum, Kanban.
 - Melhoria contínua em processos de testes.
 - Como implementar uma cultura de testes no time de desenvolvimento.
- **Projeto Final:**
 - Planejamento e execução de testes em um projeto de software real ou simulado.
 - Aplicação de diferentes tipos de testes: unitários, funcionais, de integração e de aceitação.
 - Apresentação do projeto e relatório final sobre a execução dos testes.



Apresentação da Disciplina

Programa da Disciplina

- **Atitudes esperadas**

- Tratar todos com respeito;
- Persistir, pesquisar, aprofundar conhecimentos;
- Participar das aulas de forma organizada;
- Entender o sistema de avaliação;
- Utilizar fontes complementares de informação;
- Pontualidade;
- Responsabilidade com a realização das atividades;
- Compartilhar conhecimento.





Referências

Referências Básicas

DELAMARO, Marcio; JINO, Mario; MALDONADO, Jose. **Introdução ao teste de software**. Elsevier Brasil, 2013.
WAZLAWICK, Raul. **Engenharia de software: conceitos e práticas**. Elsevier Editora Ltda., 2019.

Referências Complementares

KOSCIANSKI, André; DOS SANTOS SOARES, Michel. **Qualidade de Software-2ª Edição: Aprenda as metodologias e técnicas mais modernas para o desenvolvimento de software**. Novatec Editora, 2007.

Referências na Internet

<https://www.ibm.com/br-pt/topics/software-testing>

https://pt.wikipedia.org/wiki/Teste_de_software