

MINISTÉRIO DA EDUCAÇÃO
SECRETARIA DA EDUCAÇÃO SUPERIOR

**CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA
CELSO SUCKOW DA FONSECA
Campus Nova Friburgo**

CURSO DE BACHARELADO EM SISTEMAS DE INFORMAÇÃO

DEPARTAMENTO
Coordenação do Curso de Bacharelado em Sistemas de Informação

PLANO DE CURSO DA DISCIPLINA
TESTE E MANUTENÇÃO DE SOFTWARE

CÓDIGO
GSI9504NF

CRÉDITOS
4

PERÍODO
5º

ANO
2016

SEMESTRE
2º

PRÉ-REQUISITOS
Projeto e Arquitetura de Software
Programação de Aplicações Corporativas

DISTRIBUIÇÃO DA CARGA HORÁRIA (AULAS/SEMANA)				
PRESENCIAL			SEMI- PRESENCIAL	TOTAL AULAS/SEMANA
TEÓRICA	PRÁTICA	ESTÁGIO		
3	2	0	0	5

TOTAL DE AULAS NO SEMESTRE
90

EMENTA
Fundamentos de testes de software; Atributos de qualidade de software e relação com testes. Ciclo de vida de teste de software. Estratégias e métodos para projeto de casos de teste. Níveis de teste. Políticas, metas, planos e documentação de testes. Cobertura de testes e medição de defeitos. Testes no ambiente organizacional. Controle e monitoramento de testes. Gestão de defeitos. Automação de testes. Manutenção e evolução. Modelos de manutenção e evolução. Manutenção e refatoração em sistemas legados. Automação de testes em sistemas legados.

BIBLIOGRAFIA
BIBLIOGRAFIA BÁSICA <ol style="list-style-type: none">FEATHERS, Michael C. Working effectively with legacy code. Upper Saddle River, NJ: Prentice Hall, 2005.FREEMAN, S.; PRYCE, N. Growing object-oriented software, guided by tests. Boston: Addison-Wesley, 2011.SOMMERVILLE, Ian. Engenharia de Software. São Paulo: Pearson, 2007. BIBLIOGRAFIA COMPLEMENTAR <ol style="list-style-type: none">BECK, K. TDD: desenvolvimento guiado por testes. Porto Alegre: Bookman, 2010.FOWLER, M. Refatoração: aperfeiçoando o projeto de código existente. Porto Alegre: Bookman, 2004.MCCONNELL, S. Code Complete: um guia prático para construção de software. 2.ed. Porto Alegre: Bookman, 2005.PRESSMAN, R. S. Engenharia de Software. São Paulo: Pearson Education, 2005.SILVERMAN, Richard E. Git: guia prático. São Paulo: Novatec, 2013.TELES, V. M. Extreme Programming: aprenda como encantar seus usuários desenvolvendo software com agilidade e alta qualidade. São Paulo: Novatec, 2006.

OBJETIVOS GERAIS

1. Introduzir princípios de teste de software e de manutenção de software;
2. Projetar e desenvolver testes de software;
3. Combinar a manutenção de software com o controle de qualidade;
4. Experimentar ferramentas de apoio ao teste e à manutenção de software;
5. Realçar aspectos técnicos do estado da prática sobre testes e manutenção de software.

METODOLOGIA

Aulas expositivas dialogadas e práticas.

CRITÉRIO DE AVALIAÇÃO

Provas escritas individuais e trabalhos individuais ou em grupo.

CHEFE DO DEPARTAMENTO

NOME	ASSINATURA
DACY CÂMARA LOBOSCO	

PROFESSOR RESPONSÁVEL PELA DISCIPLINA

NOME	ASSINATURA
THIAGO DELGADO PINTO	

PROGRAMA

1. Fundamentos de testes de software
 - 1.1. Conceitos Fundamentais
 - 1.2. Objetivos dos testes
 - 1.3. Erro, defeito e falha
2. Testes e Qualidade
 - 2.1. Qualidade de Software
 - 2.2. Atributos da Qualidade
 - 2.3. Relação da qualidade com os testes
 - 2.4. Dimensões de Teste
3. Processo de Testes
 - 3.1. Fluxo básico de atividades
 - 3.2. Níveis de Teste
 - 3.3. Análise de Riscos
 - 3.4. Ambiente de Teste
 - 3.5. Gestão de defeitos
4. Projeto de Casos de Teste
5. Estratégias de Teste
 - 5.1. Técnicas de teste estrutural
 - 5.2. Técnicas de teste funcional
 - 5.3. Automação de testes
6. Manutenção e Evolução
 - 6.1. Modelos de Manutenção

- 6.2. Modelos de Evolução
- 6.3. Sistemas Legados
- 6.4. Refatoração
- 6.5. Automação de testes em sistemas legados