

MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA CELSO SUCKOW DA FONSECA
DIRETORIA DE ENSINO (DIREN)
DEPARTAMENTO DE ENSINO SUPERIOR (DEPES)
DEPARTAMENTO DE INFORMÁTICA (DEPIN)
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO (BCC)

DEPARTAMENTO	PLANO DE CURSO DA DISCIPLINA
DEPIN - Departamento Acadêmico de Informática	ENGENHARIA DE SOFTWARE

CÓDIGO	PERÍODO	ANO	SEMESTRE	PRÉ-REQUISITOS
GCC 1521	5º	2012	2	
CRÉDITOS	AULAS/SEMANA			GCC 1312 Engenharia de Requisitos
	TEÓRICA	PRÁTICA	ESTÁGIO	
4	4	0	0	
			TOTAL DE AULAS NO SEMESTRE	
			72	

EMENTA

Engenharia de Software: introdução e conceitos. O processo de Engenharia de Software. Gerência de requisitos. Projeto de software, Teste de software. Qualidade de software. Gerência de configuração de software.

BIBLIOGRAFIA

Bibliografia básica

1. PAULA FILHO, Wilson P. Engenharia de Software: Fundamentos, Técnicas, Métodos e Padrões, 3ª edição, São Paulo: LTC, 2009.
2. PRESSMAN, Roger S., Engenharia de Software – Uma Abordagem Profissional, 7ª edição, São Paulo: Mc Graw Hill, 2011.
3. SOMMERVILLE, Ian, Engenharia de Software, 9ª edição, São Paulo: Pearson Education – Addison-Wesley, 2011.

Bibliografia complementar

1. BEZERRA, Eduardo, Princípios de Análise e Projeto de Sistemas com UML, 2ª edição, Rio de Janeiro: Campus, 2007.
2. PFLEEGER, Shari L., Engenharia de Software: Teoria e Prática, 2ª edição. Prentice Hall – Br, 2003.
3. RIOS, Emerson, MOREIRA FILHO, Trayahú. Teste de Software. 3ª edição. Rio de Janeiro: Alta Books, 2013.
4. SILVEIRA, Paulo, SILVEIRA Guilherme, LOPES, Sérgio, MOREIRA, Guilherme, STEPPAT, Nico, KUNG, Fabio. Introdução à Arquitetura e Design de Software. Rio de Janeiro: Campus, 2012.
5. SCHACH, Stephen R.. Engenharia de Software – Os Paradigmas Clássico e Orientado a Objetos. 7ª edição. Porto Alegre: Bookman, 2009.

OBJETIVO GERAL

A disciplina visa fornecer ao aluno conhecimentos em engenharia de software e nos processos e métodos importantes para o ciclo de vida do desenvolvimento de sistemas computadorizados.

OBJETIVOS ESPECÍFICOS

1. Estudar os aspectos gerais relacionados ao processo de engenharia de software.
2. Estudar processos e métodos para o projeto e desenvolvimento de sistemas computadorizados.

METODOLOGIA

- Aulas expositivas, contando com recursos audiovisuais.
- Resolução de exercícios de fixação e propostos.
- Estudos de casos.

CRITÉRIO DE AVALIAÇÃO

A avaliação semestral envolve duas provas escritas (P1 e P2). As datas das provas são agendadas entre o professor e a turma. A média parcial (MP) será calculada pelo cômputo da média aritmética simples entre a nota P1 e P2:

$$MP = (P1 + P2) / 2$$

O aluno que faltar a uma das duas provas terá direito a uma avaliação alternativa, denominada segunda chamada, versando sobre todos os tópicos abordados no curso, e cuja data também é agendada entre docente e discentes. A nota obtida nessa 2ª chamada substituirá a da avaliação P1 ou P2 onde o aluno não esteve presente. Caso ele falte às duas avaliações, terá atribuído o grau ZERO em uma delas.

Segundo o regimento do CEFET-RJ, caso o aluno obtenha média parcial inferior a 3,0 (três e zero) estará reprovado diretamente. Graus MP maiores ou iguais a 7,0 (sete e zero) aprovam diretamente o aluno. Em situações onde o aluno tenha grau MP entre 3,0 inclusive e 7,0 exclusive, terá direito a uma prova final (PF), que, juntamente com a média parcial gerará uma nova média, denominada média final (MF). Essa média é calculada da seguinte forma:

$$MF = (MP + PF) / 2$$

Para ser aprovado, o aluno deve alcançar uma MF maior ou igual a 5,0 (cinco e zero). Caso contrário, estará reprovado, devendo repetir a componente curricular.

CHEFE DO DEPARTAMENTO

NOME

ASSINATURA

PROFESSOR RESPONSÁVEL PELA DISCIPLINA

NOME

ASSINATURA

PROGRAMA

1. Engenharia de software: introdução e conceitos
 - 1.1. Histórico e definição
 - 1.2. Produtos de software: conceitos, importância e características
 - 1.3. Tipos de aplicações de software
 - 1.4 Mitos do software
2. O processo de engenharia de software
 - 2.1. Aspectos da produção de software
 - 2.2. Etapas do processo de software
 - 2.3. Modelos de ciclo de vida de software
3. Gerência de requisitos
 - 3.1 Controle de mudanças
 - 3.2 Rastreabilidade
4. Projeto de software
 - 4.1. Características de projeto
 - 4.2 Projeto de arquitetura
 - 4.3. Projeto de dados
 - 4.4. Projeto de interface
 - 4.5. Projeto em nível de componente
 - 4.6. Projeto em nível de implantação
 - 4.7. Documentação do projeto
5. Testes
 - 5.1 Verificação e validação
 - 5.2 Testes de desenvolvimento
 - 5.3 Testes de release
 - 5.4 Testes de usuário
6. Qualidade de software
 - 6.1 Qualidade do produto e do processo
 - 6.2 Plano de qualidade
 - 6.3 Padrões de software
 - 6.4 Revisões
 - 6.5 Medições e métricas de software
7. Gerência de configuração de software
 - 7.1 Gerenciamento de mudanças
 - 7.2 Gerenciamento de versões
 - 7.3 Construção de sistemas
 - 7.4 Gerenciamento de releases