

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)
CURSO SUPERIOR DE TECNOLOGIA EM SISTEMAS PARA INTERNET (CST-SI)

DEPARTAMENTO	PLANO DE CURSO DA DISCIPLINA
DEPIN - Departamento Acadêmico de Informática	MANUENÇÃO E EVOLUÇÃO DE SOFTWARE

CÓDIGO	PERÍODO	ANO	SEMESTRE	PRÉ-REQUISITOS
GCC 1937	N/A	2016	1	GCC 1311 Programação Orientada a Objetos
CRÉDITOS	AULAS/SEMANA			TOTAL DE AULAS NO SEMESTRE
4	TEÓRICA	PRÁTICA	ESTÁGIO	72
	4	0	0	GCC 1521 Engenharia de Software

EMENTA

Conceitos e taxonomia. Modelos de manutenção e evolução. Engenharia reversa. Reengenharia. Sistemas legados. Análise de impacto de mudanças. Refatoração. Compreensão de programas. Reuso e engenharia de domínio.

BIBLIOGRAFIA

Bibliografia básica

1. FEATHERS, Michael C. Working effectively with legacy code. Upper Saddle River, NJ: Prentice Hall, 2005. xxi, 434. ISBN 9780131177055.
2. KERIEVSKY, Joshua. Refatoração para padrões. Porto Alegre: Bookman, 2008. xviii, 400 p., il. ISBN 9788577802449.
3. FOWLER, Martin. Refatoração: aperfeiçoando o projeto de código existente. Porto Alegre: Bookman, 2004. 365p., il. ISBN 9798536303955.

Bibliografia complementar

1. PRESSMAN, Roger S., 1947-. Engenharia de software: uma abordagem profissional. 7.ed. Porto Alegre, RS: AMGH Ed., 2011. 780 p., il. Bibliografia: p. [751]-771. ISBN 9788563308337.
2. PRESSMAN, Roger S.; LOWE, David Brian., Engenharia web. Rio de Janeiro: LTC, c2009. 416 p. ISBN 9788521616962.
3. SOMMERVILLE, Ian. Engenharia de software. 9.ed. São Paulo: Pearson, 2011. xiii, 529p., il. ISBN 9788579361081.
4. PAULA FILHO, Wilson de Pádua. Engenharia de software: fundamentos, métodos e padrões. 3.ed. Rio de Janeiro: LTC Ed., 2009. xiii, 1248p, il. Bibliografia p. [1235]-1244. ISBN 9788521616504.
5. WAZLAWICK, Raul Sidnei. Engenharia de software: conceitos e práticas. Rio de Janeiro: Elsevier, 2013. xxii, 343 p., il. Bibliografia: p.[355]-338. ISBN 9788535260847.

OBJETIVO GERAL

Introduzir conceitos, técnicas e ferramentas pertinentes a manutenção e evolução de software habilitando o aluno a trabalhar com sistemas que se encontram em estágio produtivo.

METODOLOGIA

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

CRITÉRIO DE AVALIAÇÃO

A avaliação semestral envolve duas provas escritas (P1 e P2). As datas das provas são agendadas conforme o calendário da Instituição. 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. Conceitos e taxonomia
2. Modelos de manutenção e evolução
 - 2.1. IEEE Maintenance Process
 - 2.2. ISO 14764
3. Engenharia reversa e Reengenharia
 - 3.1. Processo
 - 3.2. Técnicas
 - 3.3. Ferramentas
4. Sistemas legados
 - 4.1. Encapsulamento
 - 4.2. Migração
5. Análise de impacto de mudanças
 - 5.1. Baseado em dependências
 - 5.2. Propagação de mudanças
6. Refatoração
7. Compreensão de programas
 - 7.1. Modelos de Cognição
 - 7.2. Visualização de programa
8. Reuso e engenharia de domínio
 - 8.1. Modelos de Eng. De Domínio
 - 8.2. Modelo de maturidade de reuso
 - 8.3. Reuso e economia de recursos