

**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
<b>DEPIN - Departamento Acadêmico de Informática</b>	<b>ENGENHARIA DE REQUISITOS</b>

CÓDIGO	PERÍODO	ANO	SEMESTRE	PRÉ-REQUISITOS
<b>GTSI 1231</b>	3º	2012	2	
CRÉDITOS	AULAS/SEMANA			Nenhum
	TEÓRICA	PRÁTICA	ESTÁGIO	
4	4	0	0	
			TOTAL DE AULAS NO SEMESTRE	
			72	

EMENTA
<p>Engenharia de Software – conceitos básicos. Requisitos de Software. Processos de engenharia de requisitos. Regras de Negócio. Modelo de Casos de Uso. Diagrama de atividades. Especificação de requisitos de software.</p>

BIBLIOGRAFIA
<p>Bibliografia básica</p> <ol style="list-style-type: none"> <li>1. FILHO, Wilson P. P., Engenharia de Software: Fundamentos, Técnicas, Métodos e Padrões. São Paulo: LTC, 2003.</li> <li>2. PRESSMAN, Roger S., Engenharia de Software. São Paulo: Mc Graw Hill, 2006.</li> <li>3. SOMMERVILLE, Ian, Engenharia de Software. São Paulo: Pearson Education–Addison-Wesley, 2007.</li> </ol> <p>Bibliografia complementar</p> <ol style="list-style-type: none"> <li>1. BEZERRA, Eduardo, Princípios de Análise e Projeto de Sistemas com UML. Rio de Janeiro: Campus, 2007.</li> <li>2. PETERS, James F. e PEDRYCZ, Witold, Engenharia de Software: Teoria e Prática, Rio de Janeiro: Campus, 2001.</li> <li>3. BLAHA, Michael; RUMBAUGH, James. Modelagem e projetos baseados em objetos com UML 2. 2.ed.rev.atual. Rio de Janeiro: Elsevier, 2006.</li> <li>4. MCLAUGHLIN, Brett; POLLICE, Gary; WEST, David. Use a cabeça análise e projeto orientado ao objeto. Rio de Janeiro: Alta Books, 2010.</li> <li>5. LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. Porto Alegre, Ed. Bookman, 2007.</li> </ol>

OBJETIVO GERAL
<p>Esta disciplina tem como objetivo apresentar ao aluno o desenvolvimento de software como uma metodologia e</p>

desenvolver a capacidade de o aluno realizar de forma correta e satisfatória a elicitação de requisitos de um sistema computacional.

### **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 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 – conceitos básicos
  - 1.1. Definição de cliente, usuário, projeto, marco
  - 1.2. Tipos de software
  - 1.3. Engenharia de requisitos
2. Requisitos de Software
  - 2.1. Requisitos do usuário

- 2.2. Requisitos do sistema
- 2.3. Requisitos Funcionais
- 2.4. Requisitos Não funcionais
- 2.5. Qualidade dos requisitos
- 3. Processos de Engenharia de Requisitos
  - 3.1. Estudo de viabilidade
  - 3.2. Técnicas de Elicitação e análise de requisitos
    - 3.2.1. Brainstorming
    - 3.2.2. Workshop
    - 3.2.3. Entrevista
    - 3.2.4. Questionário
    - 3.2.5. Observação Direta
    - 3.2.6. JAD
  - 3.3. Validação de requisitos
  - 3.4. Gerenciamento de Requisitos
- 4. Regras de negócio
- 5. Modelo de Casos de Uso
  - 5.1. Motivação e objetivos
  - 5.2. Definições preliminares: ator, caso de uso
  - 5.3. Relacionamentos entre casos de uso: extensão, inclusão e generalização/especialização
  - 5.4. Mapeamento de requisitos funcionais em casos de uso, vinculado às regras de negócio
  - 5.5. Diagramas de casos de uso: definição, componentes, notação, especificação
  - 5.6. Descrição de casos de uso
- 6. Diagrama de Atividades
  - 6.1. Componentes: estados inicial e final, de ação e de atividade, fluxos de controle sequencial e paralelo, ramificação, notas/restrições, raias de natação (partition)
  - 6.2. Diagrama de atividade no processo de desenvolvimento iterativo
- 7. Especificação de Requisitos de Software (ERSw) - IEEE/ANSI830-1993