

**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
<b>DEPIN - Departamento Acadêmico de Informática</b>

PLANO DE CURSO DA DISCIPLINA
<b>ANÁLISE E PROJETO DE SISTEMAS</b>

CÓDIGO
<b>GTSI 1446</b>

PERÍODO
4º

ANO
2014

SEMESTRE
1

PRÉ-REQUISITOS
GTSI 1431 Engenharia de Requisitos GTSI 1433 Programação Orientada a Objetos

CRÉDITOS
4

AULAS/SEMANA		
TEÓRICA	PRÁTICA	ESTÁGIO
4	0	0

TOTAL DE AULAS NO SEMESTRE
72

EMENTA
Modelagem de Classes. Cenários e interações entre objetos. Modelagem de Estados. Modelagem de Atividades. Modelo de implementação.

BIBLIOGRAFIA
<p>Bibliografia básica</p> <ol style="list-style-type: none"> <li>1. BEZERRA, Eduardo, Princípios de Análise e Projeto de Sistemas com UML, Rio de Janeiro: Campus. ISBN: 9788535216967.</li> <li>2. BOOCH, RUMBAUGH e JACOBSON, UML - Guia do Usuário, Rio de Janeiro: Campus. ISBN: 8535217843.</li> <li>3. LARMAN, Craig. Utilizando UML e padrões: uma introdução à análise e ao projeto orientados a objetos e ao desenvolvimento iterativo. Porto Alegre, RS: Bookman. xiv, 695p., il. ISBN 9788560031528.</li> </ol> <p>Bibliografia complementar</p> <ol style="list-style-type: none"> <li>1. FOWLER, Martin. UML essencial: um breve guia para a linguagem-padrão de modelagem de objetos. Porto Alegre: Bookman. xv, 160 p., il. ISBN 8536304545.</li> <li>2. BLAHA, Michael; RUMBAUGH, James. Modelagem e projetos baseados em objetos com UML 2. Rio de Janeiro: Elsevier, xvii, 496 p., il. ISBN 9788535217537.</li> <li>3. COCKBURN, Alistar, Escrevendo Casos de Uso Eficazes: Um Guia Prático para Desenvolvedores de Software, Porto Alegre: Bookman.</li> <li>4. EVANS, Eric, Domain-Driven Design Atacando As Complexidades na Criação do Software, Rio de Janeiro: Alta Books.</li> <li>5. MELO, Ana Cristina. Desenvolvendo aplicações com UML 2.0: do conceitual à implementação. Rio de Janeiro: Brasport, 284 p., il. ISBN 8574521752.</li> </ol>

## OBJETIVO GERAL

Instruir no uso de metodologias e técnicas de análise orientadas a objetos para modelagem de sistemas aplicativos através do uso da Linguagem de Modelagem Unificada (UML).

## 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. Modelagem de Classes
  - 1.1. Motivação e objetivos (modelos de classes como uma ferramenta de especificação do comportamento estático sistema)
  - 1.2. Definições preliminares: classes, atributo, operação, relacionamento
  - 1.3. Diagramas de classes: definição, componentes, notação, identificação e especificação
  - 1.4. Casos de uso como um ponto inicial na identificação de classes

- 1.5. Identificando atributos e operações de uma classe
- 1.6. Identificando relacionamentos entre objetos: associação e multiplicidades
- 1.7. Casos particulares de associação: agregação, composição, herança
2. Cenários e interações entre objetos
  - 2.1. Cenários: definição, objetivos, metodologia de especificação
  - 2.2. Diagramas de sequencias (DS): definição, objetivos, metodologia de especificação
  - 2.3. Identificando objetos em um DS
  - 2.4. Eventos em um DS
  - 2.5. Identificando operações e relacionamentos entre classes em um DS
  - 2.6. Visão Geral de Padrões GRASP
    - 2.6.1. Expert, Creator, High Coesion, Low Coupling
    - 2.6.2. Polymorphism, Indirection, Pure Fabrication, Inversion of Control
3. Modelagem de Estados
  - 3.1. Motivação e objetivos (modelos de estados como uma ferramenta de especificação do comportamento dinâmico do sistema)
  - 3.2. Definições preliminares: estados, transições, estados inicial e final
  - 3.3. Identificando estados e transições
4. Modelagem de Atividades
  - 4.1. Motivação e objetivos (modelos de atividades como uma ferramenta de especificação do comportamento dinâmico do sistema)
5. Modelo de Implementação
  - 5.1. Modelo de componentes
  - 5.2. Modelo de implantação