

## Plano de Ensino

---

### 1) Identificação

**Disciplina:** INE5605 - Desenvolvimento de Sistemas Orientados a Objetos I  
**Turma(s):** 02238A  
**Carga horária:** 108 horas-aula      Teóricas: 48      Práticas: 60  
**Período:** 2º semestre de 2012

### 2) Cursos

- Sistemas de Informação (238)

### 3) Requisitos

- Sistemas de Informação (238)
  - INE5603 - Introdução à Programação Orientada a Objetos
  - INE5603 - Introdução à Programação Orientada a Objetos

### 4) Professores

- Luciana de Oliveira Rech (luciana.rech@ufsc.br)

### 5) Ementa

Implementação de pequenos projetos com programação orientada a objetos. Sistemas de Tipo; Sistemas de Tratamento de Exceções.

### 6) Objetivos

**Geral:** Capacitar os estudantes a desenvolver sistemas utilizando técnicas da programação orientada a objetos e arcabouços básicos de software

**Específicos:**

- Compreender os principais conceitos sobre orientação a objetos em sistemas de informação;
- Aprender técnicas de reuso de software;
- Dominar a utilização de arcabouços básicos de software; e
- Saber implementar sistemas empregando os conceitos da orientação a objetos

### 7) Conteúdo Programático

- 7.1) Introdução ao desenvolvimento de sistemas reusáveis de software [6 horas-aula]
- 7.2) Conceitos e mecanismos da programação orientada a objetos [24 horas-aula]
  - Objetos e classes
  - Herança e polimorfismo
  - Classes abstratas e interfaces
  - Diagramas de classes
  - Bibliotecas de classes
  - Tipos genéricos
- 7.3) Técnicas de uso comum em sistemas orientados a objetos [42 horas-aula]
  - Interface gráfica com o usuário
  - Tratamento de exceções
  - Coleções
  - Persistência de dados e objetos (serialização)
- 7.4) Práticas de Desenvolvimento de Software [36 horas-aula]
  - Introdução a práticas/técnicas de desenvolvimento orientado a objetos.
  - Construção de sistemas de software que demonstrem as características básicas da orientação a objetos.

### 8) Metodologia

As aulas serão ministradas utilizando-se as seguintes técnicas:

- \* exposição oral do conteúdo teórico em sala de aula;

- \* exposição oral do conteúdo prático (programas prontos) em laboratório;
- \* acompanhamento e discussão com os alunos envolvendo prática de programação em laboratório;

## 9) Avaliação

A avaliação consiste de N Testes (T) e um Trabalho Final (TF).

Sobre os Testes:

Cada Teste é realizado em sala de aula, individualmente, e trata de alguma questão teórica e/ou prática (programação em Java). A data e o conteúdo dos testes sempre são informados com antecedência de no mínimo 48 horas.

Sobre o Trabalho Final :

O Trabalho Final (TF) é realizado predominantemente em horários fora dos horários de aula, em grupos de no mínimo 3 e no máximo 4 alunos. O trabalho consiste no desenvolvimento de um sistema que envolva ao menos 80% do conteúdo da disciplina.

O tema do TF bem como os seus requisitos são definidos pelo professor nas primeiras semanas de aula.

Cada grupo apresentará seu TF em três momentos: TF1, TF2 e TF3. No primeiro momento (TF1) o sistema deverá contemplar aproximadamente 30% dos requisitos. No segundo momento (TF2) o sistema deverá contemplar aproximadamente 60% dos requisitos. No terceiro momento (TF3) o sistema deverá contemplar 100% dos requisitos. Em todos os momentos, o grupo deverá empregar os conceitos e técnicas apresentados na disciplina.

Cada grupo informará o professor, via atividade Moodle, quais requisitos serão desenvolvidos em cada momento do TF.

Nos dois primeiros momentos, a apresentação do sistema será em horário de aula (aproximadamente 10 minutos por grupo) na presença de todos os demais alunos. No terceiro momento, cada grupo apresentará seu sistema ao professor em horário de aula sem a presença dos demais alunos. Neste terceiro momento, cada membro do grupo poderá receber uma nota própria em função do domínio sobre o trabalho.

As três versões do TF serão entregues via Moodle no prazo estipulado pelo professor. É permitida a entrega atrasada mas, porém, com a seguinte penalidade: desconto de 0,5 pontos na nota para cada 30 minutos de atraso. O horário de entrega, para fins de desconto, é o que fica registrado no Moodle. Exemplos: atrasos de 1 a 30 minutos => desconto de 0,5 pontos; atrasos de 31 a 60 minutos => desconto de 1,0 ponto; e assim por diante.

Alunos sem grupo definido receberão nota zero no TF.

Sobre a Média Final:

A Média Final (MF) na disciplina é calculada da seguinte forma:

$$TF = TF1 * 0,20 + TF2 * 0,30 + TF3 * 0,50$$

$$MF = MT * 0,75 + TF * 0,25$$

$$\text{onde } MT = (T1 + \dots + TN) / N$$

Não é prevista atividade de recuperação para esta turma, nos termos previstos no art. 70, parágrafo 2o, da Resolução 17/CUn/97, uma vez que cumpre pelo menos um dos seguintes requisitos:

- ter pelo menos 50% de carga prática;
- ter pelo menos 50% do peso da média final originado de trabalho prático;
- ter a inadequação da aplicação de avaliação de recuperação reconhecida pelo colegiado do curso, a partir da avaliação de solicitação fundamentada de dispensa de avaliação de recuperação, encaminhada pelo(s) professor(es) autor(es) do respectivo plano de ensino, para disciplinas com carga prática prevista no programa da disciplina, com nota de trabalho prático considerada no cálculo da média final e que não tenham cumprido um dos requisitos anteriores.

## 10) Cronograma

Testes : Os testes ocorrem aproximadamente a cada duas semanas.

Trabalho Final : TF1 : ao final do segundo mês de aula; TF2 : ao final do terceiro mês de aula; TF3 : ao final do quarto mês de aula. As datas específicas serão anunciadas na página Moodle da disciplina em função da quantidade de grupos.

### **11) Bibliografia Básica**

- DEITEL, Harvey M.; DEITEL, Paul J. Java como programar. Porto Alegre: Bookman, 2001. 1202p.
- BOOCH, G. Object-Oriented Analysis and Design with Applications, 2a Edição. Addison-Wesley, 1994.

### **12) Bibliografia Complementar**

- SANTOS, Rafael. Introdução à Programação Orientada a Objetos Usando Java. Editora Campus Ltda, 2003.
- BORATTI, Isaías C. Programação Orientada a Objetos em Java. Editora VisualBooks, 2007.
- ECKEL, Bruce. Thinking in Java. 3. ed. Prentice-Hall, Dezembro 2002.
- LIANG, Y. Daniel. Introduction to java programming. 2nd ed. Indianapolis: Que E&T, c1999. 610p. ISBN 1-58076-255-7
- SILVA, Ricardo Pereira e. UML 2 – Modelagem Orientada a Objetos. Editora VisualBooks, 2007.
- WAZLAWICK, R. Análise e Projeto de Sistemas de Informação Orientados a Objetos. Editora Campus, 2004.