



## Plano de Ensino

---

### 1) Identificação

<b>Disciplina:</b>	INE5645 - Programação Paralela e Distribuída		
<b>Turma(s):</b>	05238B		
<b>Carga horária:</b>	72 horas-aula	Teóricas: 36	Práticas: 36
<b>Período:</b>	1º semestre de 2014		

### 2) Cursos

- Sistemas de Informação (238)

### 3) Requisitos

- INE5611 - Sistemas Operacionais

### 4) Ementa

Modelos de interação entre processos. Compartilhamento de memória e mecanismos de sincronização. Troca de mensagens e mecanismos de comunicação. Problemas clássicos. Princípios de implementação. Programação em redes de computadores. Programação distribuída. Linguagens paralelas e distribuídas.

### 5) Objetivos

Geral: Apresentar conceitos e técnicas de programação paralela e distribuída.

**Específicos:**

- Apresentar e exercitar a programação paralela com sincronização e troca de mensagens.
- Apresentar exemplos de interfaces de programação e linguagens paralelas e distribuídas

### 6) Conteúdo Programático

- 6.1) Introdução à Programação Paralela e Distribuída [4 horas-aula]
  - Vantagens e Dificuldades
  - Plataformas de Execução
  - Suporte Computacional
- 6.2) Programação Paralela [16 horas-aula]
  - Processos
  - Threads
  - Paralelismo em Java
- 6.3) Controle de Concorrência [18 horas-aula]
  - Monitores
  - Locks
  - Semáforos
  - Concorrência na API Java
- 6.4) Programação Distribuída [8 horas-aula]
  - Modelos de sistemas distribuídos
  - Elementos básicos da comunicação
  - Comunicação por passagem de mensagem
- 6.5) Comunicação entre Processos [26 horas-aula]
  - Pipes
  - Sockets
  - RMI

### 7) Metodologia

AT (Aula Teórica), AP (Aula Prática), TP (Tarefa Prática), TT (Tarefa Teórica)

Unidade I - Introdução à Programação Paralela e Distribuída (AT)

Unidade II - Programação Paralela (AT, AP, TP)

Unidade III - Monitor, Locks e Semáforos (AT, AP, TP)

Unidade IV - Programação Distribuída com Sockets (AT, AP, TP)

Unidade V - Programação Distribuída com RMI (AT, AP, TP)

Unidade VI - Eventos e Notificações - Sockets ou RMI (AT, AP, TP)

## 8) Avaliação

Os alunos serão avaliados com base em seu desempenho nas seguintes atividades: 2 Provas teóricas (P1 e P2) realizadas em sala de aula; 4 Atividades de laboratório (LAB1 a LAB5); 1 Projeto final (PROJ); e mais pequenas tarefas práticas realizadas em laboratório, preliminarmente às tarefas LAB1, LAB2, LAB3, LAB4 e LAB5.

A média teórica será calculada aplicando a fórmula:  $MT = (P1 + P2)/2$ .

Caso o aluno não alcance média teórica 6,0 (seis), uma prova de recuperação (PR) será aplicada. A média MT, para esse caso, será  $((P1+P2)/2)+PR)/2$ .

A média prática será atribuída aplicando a fórmula  $MP = (0,20 (LAB1) + 0,20 (LAB2) + 0,20 (LAB3) + 0,10 (LAB4) + 0,10 (LAB5) + 0,20*PROJ)/6$ .

Para ser aprovado, o aluno deve possuir médias teórica e prática iguais ou superiores a 6,0 (seis). Nesse caso, a média final será calculada aplicando a fórmula  $MF = (MT + MP)/2$ . Caso contrário, a média final será igual ao valor mais baixo dentre as médias teórica e prática.

Observações:

Será atribuída uma nota para a equipe (dupla de alunos) que para cada trabalho de laboratório. Se um aluno não responder a uma pergunta do trabalho, a equipe terá desconto na nota do trabalho. Em caso de cópia de trabalhos de qualquer fonte, todos os alunos envolvidos terão nota igual a zero. Será utilizada uma ferramenta de detecção de plágio (ex. SIM - Software and text similarity tester) para comparação dos trabalhos.

Trabalhos entregues com atraso terão desconto automático de pontos. Após a segunda semana (14 dias após o fim do prazo original), o trabalho não mais será aceito, ou seja, terá nota zero.

Em caso de "cola" nas avaliações, os alunos envolvidos estarão automaticamente reprovados com média zero, e o caso poderá ser reportado ao colegiado do curso para que sejam tomadas medidas disciplinares.

Para efeito de avaliação, o prazo para entrega de tarefas deverá ser respeitado, não aceitando tarefas fora do prazo.

Conforme parágrafo 2º do artigo 70 da Resolução 17/CUn/97, o aluno com frequência suficiente (FS) e média final no período (**MF**) entre 3,0 e 5,5 terá direito a uma nova avaliação ao final do semestre (**REC**), sendo a nota final (**NF**) calculada conforme parágrafo 3º do artigo 71 desta resolução, ou seja:  $NF = (MF + REC) / 2$ .

## 9) Cronograma

17/3 Unidade I - Plano de Ensino, Histórico da Programação concorrente (AT)

19/3 Unidade II - Multithreading: Threads, Estados de threads, Prioridades de threads, Implementação de threads (AT, AP)

24/3 Ambiente de execução Multithreading (AT)

26/3 Sincronização de threads: Produtor-Consumidor sem sincronização (AP)

31/3 Unidade III - Monitores (AT)

02/4 Produtor-Consumidor com sincronização de Monitor (AP)

07/4 Atividade de Laboratório 1 (Lab 1) (TP)

09/4 Atividade de Laboratório 1 (Lab 1) (TP)

14/4 Unidade III - Locks (AT)

16/4 Produtor-Consumidor com sincronização de Looks (AP)

21/4 Atividade de Laboratório 2 (Lab 2) (TP)

23/4 Atividade de Laboratório 2 (Lab 2) (TP)

28/4 Unidade III - Semáforos (AT)

30/4 Atividade de Laboratório 3 (Lab 3) (TP)

05/5 Atividade de Laboratório 3 (Lab 3) (TP)

07/5 Questionário 1 (TT)

12/5 Prova 1 - Assunto: Unidades I, II e III

14/5 Correção da Prova 1

19/5 Unidade IV - Sockets - Aplicações Distribuídas (AT)

21/5 Datagram Sockets (AP)

26/5 TCP Stream Sockets (AP)

28/5 Multicast Socket (AP)

02/6 Atividade Laboratório 4 (Lab 4) (TP)

04/6 Atividade Laboratório 4 (Lab 4) (TP)

09/6 Unidade V - Java RMI - Aplicações Distribuídas (AP)

11/6 Exemplo básico RMI (AP)  
16/6 Atividades do Laboratório 5 (Lab 5) (TP)  
18/6 Atividades do Laboratório 5 (Lab 5) (TP)  
23/6 Unidade VI - Eventos e Notificações (AT)  
25/6 Atividades do Laboratório 6 (AP)  
30/6 Questionário 2 (TT)  
02/7 Prova 2 - Unidades IV, V e VI  
07/7 Correção da Prova 2  
09/7 Atividades do Laboratório 6 (TP)  
14/7 Atividades do Laboratório 6 (TP)  
16/7 Prova de Recuperação

#### **10) Bibliografia Básica**

- ANDREWS, G. R., Concurrent Programming, Benjamin-Cummings, 1991.
- COULOURIS, George; DOLLIMORE, Jean; KINDBERG, Tim. Distributed Systems: -Concepts and Design. 3rd Edition. Addison-Wesley, 2001.
- DEA, Doug &#8220;Concurrent Programing in Java&#8221;, 2nd Ed., Addison-Wesley, 2000.
- DEITEL, Harvey M.; DEITEL, Paul J. Java: Como Programar. 4a Edição. Bookman, 2002.
- HORSTMANN, Cay S.; CORNELL, Gary. Core Java 2. Vol I e II. Makron Books, 1999.

#### **11) Bibliografia Complementar**

- ORFALI, Robert; HARVEY, Dan. Client/Server Programming with Java and CORBA. 2nd Edition. John Wiley, 1998.
- STEVENS, W. R., Unix Network Programming, vols. 1 e 2, Prentice-Hall, 1998.
- TANENBAUM, Andrew. Sistemas Operacionais Modernos. Prentice-Hall, 2003.