



## Plano de Ensino

---

### 1) Identificação

<b>Disciplina:</b>	INE5645 - Programação Paralela e Distribuída		
<b>Turma(s):</b>	05238		
<b>Carga horária:</b>	72 horas-aula	Teóricas: 36	Práticas: 36
<b>Período:</b>	1º semestre de 2015		

### 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)

Plano de Ensino - INE5645 - Programação Paralela e Distribuída - 17/03/2014 - Página: 1/3 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);

8 Atividades de laboratório (LAB1, LAB2, LAB3, LAB4(a,b,c,d), LAB5);

O aluno, após terminar a tarefa prática, deverá mostrar a execução ao professor.

A média das provas (MP) será calculada aplicando a fórmula:

$$MP = (P1 + P2)/2.$$

A média das tarefas práticas (MT) será atribuída aplicando a fórmula

$$MT = [ 0,20*(LAB1) + 0,20*(LAB2) + 0,10*(LAB3) + 0,05*LAB4a + 0,05*LAB4b + 0,05*LAB4c + 0,15*LAB4d + 0,20*LAB5$$

Nota final do semestre (Moodle) = Média do semestre (Moodle)=

$$MF = (MP + MT)/2$$

Casos de recuperação: Vide no que segue.

Para realização de avaliações em atraso, de acordo com a RESOLUÇÃO Nº 17/CUn/97, de 30 de setembro de 1997:

Art. 70 § 4o - Ao aluno que não comparecer às avaliações ou não apresentar trabalhos no prazo estabelecido será atribuída nota 0 (zero).

Art. 74 - O aluno, que por motivo de força maior e plenamente justificado, deixar de realizar avaliações previstas no plano de ensino, deverá formalizar pedido de avaliação à Chefia do Departamento de Ensino ao qual a disciplina pertence, dentro do prazo de 3 (três) dias úteis, recebendo provisoriamente a menção I.

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

09/03 Unidade I - Plano de Ensino, Histórico da Programação concorrente(AT)  
11/03 Unidade II - Multithreading: Threads, Estados de threads, Prioridades de threads, Implementação de threads (AT, AP)  
16/03 Ambiente de execução Multithreading (AT)  
18/03 Sincronização de threads: Produtor-Consumidor sem sincronização (AP)  
25/03 Unidade III - Monitores (AT)  
30/03 Produtor-Consumidor com sincronização de Monitor (AP)  
06/04 Atividade de Laboratório 1 (Lab 1) (TP)  
08/04 Atividade de Laboratório 1 (Lab 1) (TP)  
15/04 Unidade III - Locks (AT)  
22/04 Produtor-Consumidor com sincronização de Looks (AP)  
27/04 Atividade de Laboratório 2 (Lab 2) (TP)  
29/04 Atividade de Laboratório 2 (Lab 2) (TP)  
04/05 Unidade III - Semáforos (AT)  
06/05 Atividade de Laboratório 3 (Lab 3) (TP)  
11/05 Atividade de Laboratório 3 (Lab 3) (TP)  
13/05 Prova 1 - Assunto: Unidades I, II e III  
18/05 Unidade IV - Sockets - Aplicações Distribuídas (AT)  
20/05 Datagram Sockets (AP)  
25/05 TCP Stream Sockets (AP)  
27/05 Multicast Socket (AP)  
01/06 Atividade Laboratório 4 (Lab 4) (TP)

03/06 Atividade Laboratório 4 (Lab 4) (TP)  
08/06 Unidade V - Java RMI - Aplicações Distribuídas (AP)  
10/06 Exemplo básico RMI (AP)  
15/06 Atividades do Laboratório 5 (Lab 5) (TP)  
17/06 Atividades do Laboratório 5 (Lab 5) (TP)  
22/06 Prova 2 - Unidades IV, V e VI  
29/06 Prova de Recuperação  
06/07 Entrega das Notas Finais

#### **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.