



## Plano de Ensino

---

### 1) Identificação

<b>Disciplina:</b>	INE5645 - Programação Paralela e Distribuída		
<b>Turma(s):</b>	05238A		
<b>Carga horária:</b>	72 horas-aula	Teóricas: 36	Práticas: 36
<b>Período:</b>	1º semestre de 2016		

### 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

Pela característica prática da disciplina, todas as aulas serão realizadas em laboratório de ensino, enquanto que as provas escritas serão realizadas na sala de aula destinada à disciplina.

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

Unidade I - Introdução à Programação Concorrente, Paralela e Distribuída (AT). Processos. Threads, Ciclo de Vida de Threads. Prioridades de threads. Métodos de mudança de estado de threads.  
Unidade II - Programação Concorrente sem sincronização. Com sincronização de threads (AT, TP).  
Unidade III - Programação Concorrente, Mecanismos de sincronização de threads: Monitor (Lab 1), Locks (Lab 2) e Semáforos (Lab 3) (AT, TP).  
Unidade IV - Programação Paralela Lab 4(AT, TT, TP).  
Unidade V - Programação Distribuída em Redes, com Sockets (AT, TP), Lab (a,b,c,d).  
Unidade VI - Programação Distribuída em Redes, com objetos distribuídos Java RMI (AT, TP) (Lab 6).

## 8) Avaliação

Os alunos serão avaliados com base em seu desempenho nas seguintes atividades:

2 Provas Teóricas (P1 e P2);

09 Atividades de laboratório (Lab 1, Lab 2, Lab 3, Lab 4, Lab 5(a,b,c,d), Lab 6);

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

$$MT = [ 0,15*(Lab 1) + 0,15*(Lab 2) + 0,15*(Lab 3) + 0,25*Lab4 + 0,05*Lab 5a + 0,05*Lab 5b + 0,05*Lab 5c + 0,10*Lab 5d + 0,05*Lab 6 ]$$

Por se tratar de uma disciplina essencialmente prática, para que o aluno tenha a sua nota de cada tarefa prática na sua totalidade, o aluno deve realizar cada tarefa no horário da aula (por si só ou em dupla) e termine a mesma, mostrando sua execução correta, na data prevista, ao professor.

O professor procurará realizar recuperações de notas das provas escritas, logo após os resultados das mesmas, durante o semestre, procurando atender aos alunos diante de suas necessidades de aprendizado.

Nota final do semestre (Moodle) = Média do semestre (Moodle)=  
 $MF = (MP + MT)/2$

Casos de recuperação (regras da UFSC): Vide no que segue.

Para realização de avaliações em atraso, de acordo com a RESOLUÇÃO N° 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

O cronograma aproximado, pelas experiências em períodos passados:

14/03 Unidade I - Plano de Ensino, Histórico da Programação Concorrente, Paralela e Distribuída (AT)

16/03 Unidade I – Processos e Threads. Ciclo de Vida das Threads. Métodos de mudança de estado de threads. Prioridades de threads. Implementação de threads (AT)

21/03 Unidade II – Multithreading. Exemplo sem sincronização; Exemplo com sincronização de threads. Modelo Produtor-Consumidor. (AP)

25/03 Unidade III - Atividade de Laboratório 1 (Lab 1) Monitor (TP)

30/03 Unidade III - Atividade de Laboratório 1 (Lab 1) Monitor (TP)

04/04 Unidade III - Atividade de Laboratório 1 (Lab 1) Monitor (TP)

06/04 Unidade III - Atividade de Laboratório 2 (Lab 2) Locks (TP)

11/04 Unidade III - Atividade de Laboratório 2 (Lab 2) Locks (TP)  
13/04 Unidade III - Atividade de Laboratório 2 (Lab 2) Locks (TP)  
18/04 Unidade III - Atividade de Laboratório 2 (Lab 2) Locks (TP)  
20/04 Unidade III - Atividade de Laboratório 3 (Lab 3) Semáforos (TP)  
25/04 Unidade III - Atividade de Laboratório 3 (Lab 3) Semáforos (TP)  
27/04 Unidade III - Atividade de Laboratório 3 (Lab 3) Semáforos (TP)  
02/05 Unidade III - Atividade de Laboratório 3 (Lab 3) Semáforos (TP)  
04/05 Prova 1 – Unidades I, II e III. (escrita)  
09/05 Unidade IV - Atividade de Laboratorio IV (Lab 4) Programação paralela (AT)  
11/05 Unidade IV - Atividade de Laboratorio IV (Lab 4) Programação paralela (TP)  
16/05 Unidade IV - Atividade de Laboratorio IV (Lab 4) Programação Paralela (TP)  
18/05 Unidade IV - Atividade de Laboratorio IV (Lab 4) Programação Paralela (TP)  
23/05 Unidade IV - Atividade de Laboratorio IV (Lab 4) Programação Paralela (TP)  
25/05 Unidade IV - Atividade de Laboratorio IV (Lab 4) Programação Paralela (TP)  
30/05 Unidade V - Atividade de Laboratorio V (Lab 5a) Datagrama Sockets (TP)  
01/06 Unidade V - Atividade de Laboratorio V (Lab 5b) TCP Sockets (TP)  
06/06 Unidade V - Atividade de Laboratorio V (Lab 5c) IP Multicast (TP)  
08/06 Unidade V - Atividade de Laboratorio V (Lab 5d) Transferência de Arquivo (TP)  
13/06 Unidade V - Atividade de Laboratorio V (Lab 5d) Transferência de Arquivo (TP)  
15/06 Unidade V - Atividade de Laboratorio V (Lab 5d) Transferência de Arquivo (TP)  
20/06 Unidade VI - Atividade de Laboratório VI (Lab 6) Callback (TP)  
22/06 Unidade VI - Atividade de Laboratório VI (Lab 6) Callback (TP)  
27/06 Unidade VI - Atividade de Laboratório VI (Lab 6) Callback (TP)  
29/06 Unidade VI - Atividade de Laboratório VI (Lab 6) Objetos Distribuídos com RMI (TP)  
04/07 Prova 2 – Unidades IV, V e VI (escrita)  
06/07 Entrega de Tarefas Atrasadas  
11/07 Prova de Recuperação (escrita)  
13/07 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.