

## 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 2019		

### 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

A metodologia empregada nas aulas será em termos da divisão do conteúdo em unidades (I a IV), correspondendo, inicialmente, em aulas-teóricas, seguidas de tarefas práticas.

Em cada Unidade, o professor poderá realizar no horário de aula, tarefas teóricas relativas ao ensino da Unidade correspondente, que valerá pontos para avaliar as notas P1 e P2. Para essas atividades serão disponibilizadas via

página da disciplina, o arquivo a ser usado na elaboração das questões, o qual deverá ser impresso e levado à sala de aula pelo aluno. O aluno deverá entregar ao professor ao findar sua atividade em sala de aula.

As tarefas práticas poderão ser realizadas em laboratório de ensino, em sites especializados em programação paralela ou através dos recursos computacionais dos alunos. Tarefas práticas serão submetidas ao Moodle.

## 8) Avaliação

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

- 2 Provas Teóricas (P1 e P2) que serão compostas de tarefas teóricas em aula, mais a presença nas palestras, e de uma parte escrita, sem consulta, e em sala de aula.

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

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

- A presença nas palestras valerão pontuação para cada uma das provas. Devendo a Palestra 1 valer 10% da P1. Enquanto as Palestras 2 e 3 valem 20%, cada, respectivamente, em relação à P2.

A média das tarefas práticas (MT) será atribuída aplicando a fórmula seguinte, às atividades práticas (Lab I, Lab II, Lab III, Lab IV):

$$MT = [0,10*Lab I (OpenMP) + 0,10*Lab II (CUDA) + 0,25*Lab III (Hadoop) + 0,15*Lab IV (Spark)]$$

Cada Lab poderá ser subdividido em pequenas tarefas práticas, que somadas darão a nota do Lab

O aluno deve realizar cada tarefa prática, por si só ou em dupla.

$$\text{Media do semestre} = MF = (0.70*MP + 0.30*MT)/2.0$$

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 comunicar ao professor ou 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.

Caso o tamanho da turma torne o ambiente de uso do laboratório de ensino, inadequado para a locomoção do professor e alunos, as tarefas práticas serão explicadas em aula anterior e, poderão ser realizadas em outro ambiente, a ser escolhido pelos alunos. Entretanto, neste caso, dado as dificuldades de acompanhamento remoto do professor às várias duplas de alunos, as notas dessas tarefas no Moodle, serão dependentes das respostas corretas sobre questões colocadas nas provas escritas, relativas as referidas tarefas. Não sendo reconhecidas no Moodle, tarefas postadas em contradição às respostas dadas nas provas escritas.

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

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

Março - 11/03 a 27/03

Plano de Ensino,  
Introdução à Programação Paralela e Distribuída,  
Unidade I (Lab I-OpenMP),

Abril - 01/04 a 29/04

Unidade II (Lab II - CUDA), Palestra 1 "Programação Paralela com CUDA"

Prova I - 29/04 (Segunda-feira)

Maio - 01/05 a 29/05

Unidade III (Lab III), Programação Paralela e Distribuída com Hadoop.  
Palestra 2 "Processamento paralelo e distribuído com Hadoop"

Junho - 03/06 à 26/06

Unidade IV (Lab IV), Palestra 3 "Processamento Paralelo e Distribuído com Spark"

Prova II - 26/06 (Quarta-feira)

Rec - 01/06

Divulgação das notas finais - até 03/07

#### **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.