

<b>Estabelecimento:</b> Universidade Lusófona de Humanidades e Tecnologias		
<b>Unidade Orgânica:</b> Escola de Comunicação, Artes e Tecnologias de Informação		
<b>Curso :</b> Engenharia Informática e Sistemas de Informação		
<b>Grau ou Diploma:</b> 2º Ciclo 1º Ciclo   2º ciclo   3º Ciclo		
<b>Unidade curricular:</b> Técnicas de Computação Paralela		
<b>Nível:</b> 2º Ciclo 1º Ciclo   2º Ciclo   3º Ciclo		<b>ECTS:</b> 8
<b>Área Científica:</b> Ciências Informáticas De acordo com a Portaria 256/2005, de 16 de Março		<b>Código:</b>
<b>Total Horas Trabalho:</b> 213	<b>Contacto:</b> 2TP (TP - P - T - PL - TC - S - E - OT - O) – Exemplo: 2TP + 2P	
	<b>Tempo de Contacto:</b> 30 Horas Total de horas Semestral	
<b>Tipo:</b> Semestral Anual   Semestral   Trimestral   Modular   Outra	<b>Natureza:</b> Obrigatória Opcional   Obrigatória	<b>Língua:</b> Português Língua de Instrução
<b>Ano/Semestre:</b> 1º Ano / 2º Semestre Ex: 1º Ano / 2º Semestre	<b>Precedências:</b>  Se aplicável	
<b>Disciplinas complementares recomendadas:</b>  Arquitectura de Computadores (1º ciclo) Linguagens de Programação II (1º ciclo) Sistemas Operativos (1º Ciclo) Computação Distribuída (1º ciclo)  Código:	<b>Estágio Profissional:</b> Não Sim / Não	
<b>Docente:</b> José Luís de Azevedo Quintino Rogado Nome do Docente		
<b>Conteúdos programáticos</b> (1000 caracteres disponíveis): Course contents		
<ol style="list-style-type: none"> <li>1. Introdução <ul style="list-style-type: none"> <li>Motivações para o Paralelismo</li> <li>Aplicações: Engenharia, Científicas e Comerciais</li> </ul> </li> <li>2. Arquitecturas para Computação Paralela <ul style="list-style-type: none"> <li>Evolução e Limitações das Arquitecturas</li> <li>Modelos de Computadores Paralelos</li> <li>Modelo de Programação Paralela</li> <li>Computação, Controle e Comunicação</li> <li>Estrutura Física das plataformas</li> </ul> </li> </ol>		

Processamento vs. Comunicação

### 3. Desenho de Algoritmos Paralelos

Introdução

Técnicas de Decomposição e Aglomeração

Mapeamento e Balanceamento de Carga

Comunicação Inter Tarefas

Minimização de interacções

Modelos de Algoritmos Paralelos

### 4. Quantificação do Desempenho

Definição de Desempenho

Aproximações à Modelização do Desempenho

Modelos e Análise de escalabilidade

Desenho e Experimentação

Avaliação de Implementações

Problemas de Input/Output

### 5. Modelos e Plataformas de Programação Paralela

Programação por Mensagens: Message Passing Interface

Programação em Memória Partilhada: Modelo de Threads

Clusters: funcionamento e desempenho

Grid Computing: mecanismos e aplicações

#### **Objectivo da unidade curricular** (1000 Caracteres):

Objectives

Em certas áreas da engenharia, os problemas a resolver são de grande complexidade e implicam transferência, armazenamento e tratamento de quantidades massivas de informação. As capacidades fornecidas pelos meios computacionais clássicos não são suficientes para dar resposta aos requisitos que lhes estão associados.

O recurso a técnicas de computação paralela, baseadas na agregação de recursos de processamento e/ou armazenamento (multicores, clusters e Grids) aparece como uma forma económica e escalável de tratar esses problemas. A engenharia associada à aplicação destas tecnologias constitui tema extremamente fecundo e actual, que abre caminho à implementação de soluções inovadoras em áreas de reconhecido valor científico e tecnológico.

O objectivo desta Unidade Curricular é apresentar os princípios fundamentais da Computação Paralela, tanto a nível das plataformas hardware e software como das tecnologias utilizadas no desenvolvimento e optimização de aplicações paralelas.

#### **Conhecimentos, capacidades e competências a adquirir** (1000 Caracteres):

Learning outcomes of the course unit

No final da cadeira, os alunos deverão saber:

- Identificar a existência de tarefas independentes em algoritmos computacionais;
- Decompô-los em vários fluxos de execução paralela;
- Determinar o grau de paralelismo óptimo e prever os ganhos de desempenho;
- Utilizar alguns dos principais paradigmas de programação paralela e as respectivas plataformas de execução para realizar a sua implementação.

#### **Metodologias de ensino e avaliação** (1000 caracteres):

Planned learning activities, teaching methods and assessment methods and criteria

A metodologia de ensino privilegia a aquisição dos conceitos fundamentais de computação paralela através de uma abordagem em que os conceitos teóricos apresentados são consolidados pela análise de algoritmos conhecidos, nos quais é determinado o grau de paralelismo, realizada a decomposição em fluxos de execução independentes, e a implementação com base nos modelos de programação abordados, com análise crítica da optimização de desempenho.

A implementação é realizada em computadores configurados em rede independente, utilizando de plataformas de computação paralela (MPI) ou de clustering (OpenPBS).

A aprendizagem tem igualmente uma forte componente de trabalho de pesquisa realizado individualmente, complementada pela apresentação de temas relevantes para o programa da cadeira.

A avaliação é constituída pela realização de uma apresentação (40%) e de um projecto (60%). A aprovação na cadeira é obtida com uma nota mínima de 10 valores na média ponderada das duas componentes.

**Bibliografia principal – estilo APA** (1000 caracteres):

Recommended or required reading:

- . “Introduction to Parallel Computing (2<sup>nd</sup> Edition)”, by Ananth Grama, George Karypis, Anshul Gupta, ed. Addison Wesley, 2003, ISBN-10: 0201648652
- . “Designing and Building Parallel Programs”, by Ian Foster, available on-line <http://www-unix.mcs.anl.gov/dbpp>
- . “Foundations of Multithreaded, Parallel and Distributed Programming”, by Gregory Andrews, Ed. Addison Wesley, 2000, ISBN-10: 0201357526
- . “The Grid 2: Blueprint for a New Computing Infrastructure”, by Ian Foster (Editor) and Carl Kesselman (Editor), ed. Morgan Kaufmann, 2003, ISBN-10: 1558609334

As regras do estilo APA podem ser consultadas em <http://www.apastyle.org/index.aspx>

**Data:** 28 / 12 / 2010 (dd/mm/aaaa)