

## Plano Analítico: Computação Paralela e Distribuída

### 1. Identificação da Unidade Curricular

- **Curso:** Engenharia de Redes e Telecomunicações (ERT)
- **Ano:** 5º | **Semestre:** 1º
- **Créditos:** 6.0 UC
- **Carga Horária Total:** 90 Horas
- **Distribuição:**
  - **Teóricas (T):** 20h
  - **Teórico-Práticas (TP):** 20h
  - **Práticas (P):** 20h
  - **Trabalho Autónomo (TA):** 22h
  - **Orientação e Tutoria (OT):** 4h
  - **Avaliação (AV):** 4h

---

### 1. Fundamentação

A capacidade de processamento de um único núcleo atingiu limites físicos significativos. A solução moderna é o paralelismo. Para um engenheiro de ERT, compreender como dividir uma tarefa complexa em subtarefas que executam simultaneamente (paralelismo) e como gerir a comunicação entre máquinas fisicamente separadas (distribuição) é crucial para otimizar servidores de rede e plataformas de computação em nuvem.

### 2. Objectivos Instrutivos e Educativos

- **Instrutivos:** Diferenciar arquiteturas de memória partilhada e memória distribuída; aplicar leis de desempenho (**Lei de Amdahl** e **Lei de Gustafson**); dominar bibliotecas de passagem de mensagens (**MPI**) e multithreading (**OpenMP**); compreender algoritmos de consenso e sincronização em sistemas distribuídos.
- **Educativos:** Fomentar a eficiência algorítmica e o uso racional de recursos computacionais; desenvolver a capacidade de resolver problemas de grande escala; promover o pensamento lógico para lidar com concorrência e estados globais.

### 3. Resultado de Aprendizagem

O estudante será capaz de:

- Identificar secções de código passíveis de serem paralelizadas.
- Desenvolver aplicações que utilizam múltiplos núcleos e múltiplos nós de processamento.
- Resolver problemas de condições de corrida (*race conditions*) e impasses (*deadlocks*).

- Avaliar a escalabilidade e o *speedup* de um sistema computacional.
- Implementar mecanismos de tolerância a falhas em clusters.

#### 4. Planejamento Temático (6 UC)

Tema	Horas (T+TP+P)	Conteúdo Programático
I. Fundamentos de Paralelismo	8h	Taxonomia de Flynn (SISD, SIMD, MISD, MIMD); Lei de Amdahl; Métricas de desempenho.
II. Programação Multicore	12h	Threads e processos; Memória partilhada; OpenMP: Diretivas, sincronização e partilha de dados.
III. Sistemas de Memória Distribuída	14h	Passagem de mensagens; Biblioteca MPI (Message Passing Interface); Comunicação ponto-a-ponto e coletiva.
IV. Algoritmos Distribuídos	14h	Relógios lógicos (Lamport); Eleição de líder; Consenso e exclusão mútua distribuída.
V. Computação em Cluster e Grid	12h	Arquitetura de Clusters; Introdução ao MapReduce/Hadoop; Virtualização e containers para HPC.

#### 5. Recomendações Metodológicas

- **Laboratório de Programação (20h):** Desenvolvimento de exercícios em **C/C++** ou **Python** utilizando **OpenMP** para paralelismo local e **MPI** (via MPICH ou OpenMPI) para paralelismo em rede.
- **Hands-on Cluster:** Configuração de um pequeno cluster de Raspberry Pi ou máquinas virtuais para executar uma tarefa de processamento distribuído.
- **Análise de Performance:** Uso de ferramentas de *profiling* para medir o tempo de execução e identificar gargalos de comunicação.

#### 6. Sistema de Avaliação

Conforme a alocação de **4h para AV**:

- **Avaliação Contínua (60%):** Projeto prático de paralelização de um algoritmo (ex: multiplicação de matrizes de larga escala ou processamento de imagem) (40%) e relatórios de laboratório (20%).
- **Avaliação Formal (40%):** Exame final focado na teoria de arquiteturas paralelas, leis de desempenho e algoritmos distribuídos.

#### 7. Bibliografia Principal Indicada

1. **COULOURIS, G., DOLLIMORE, J. & KINDBERG, T.** *Sistemas Distribuídos: Conceitos e Projeto*. Bookman.
2. **PACHECO, Peter.** *An Introduction to Parallel Programming*. Morgan Kaufmann.

3. **QUINN, Michael J.** *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill.