



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

---

**SISTEMAS DISTRIBUIDOS Y  
PARALELOS**

**Año 2023**

**Carrera/ Plan:**

Ingeniería en Computación Plan 2008/2011

**Año:** 5to

**Regimen de Cursada:** *Semestral*

**Carácter:** Obligatoria

**Correlativas:** Concurrencia y Paralelismo

**Profesor:** Dr. Adrian Pousa

**Docentes Auxiliares:** -

**Hs. semanales teoría:** 3 hs.

**Hs. semanales práctica:** 3 hs.

---

**Fundamentación**

La evolución tecnológica de los procesadores ha motivado el procesamiento paralelo. Formar al alumno (que ya tiene conocimientos previos de Concurrencia y sus aplicaciones) en los fundamentos de los sistemas paralelos, los paradigmas de programación paralela y las métricas de rendimiento asociadas resulta un aporte fundamental para el futuro profesional. Esta tarea de formación se combina con trabajo experimental sobre sistemas paralelos concretos, disponibles en la Facultad.

**Objetivos generales**

Caracterizar los problemas de procesamiento paralelo desde dos puntos de vista: la arquitectura física y los lenguajes de programación, poniendo énfasis en la transformación de algoritmos secuenciales en paralelos. Describir los modelos de cómputo paralelo y los paradigmas de programación paralela. Estudiar las métricas de performance asociadas al paralelismo, así como modelos de predicción de performance orientados a diferentes arquitecturas multiprocesador. Plantear casos concretos de procesamiento paralelo, resolubles sobre distintas arquitecturas multiprocesador.



### **Contenidos Mínimos**

- Arquitecturas de procesamiento paralelo.
- Diseño de algoritmos paralelos.
- Programación de algoritmos paralelos: Modelo de memoria compartida. Modelo de memoria distribuida. Modelos híbridos.
- Paradigmas de resolución de sistemas paralelos.
- Métricas.

### **Programa Analítico**

#### **Unidad 1: Conceptos básicos**

Concurrencia y Paralelismo. Objetivos del procesamiento paralelo. Concepto de Sistema Paralelo. Software de un Sistema Paralelo. Hardware de un Sistema Paralelo. Motivación de los Sistemas Paralelos. Áreas de aplicación. Limitaciones actuales de hardware. Ley de Moore. Conceptos de paralelismo implícito y explícito. Paralelismo implícito. Conceptos de Segmentación/Pipelining y división funcional. Transparencia. Procesador Escalar y Superescalar. Paralelismo explícito. Comportamiento de las aplicaciones: CPU-Bound, Memory-Bound, I/O-Bound, Híbridos por fases. Etapas en el diseño de algoritmos paralelos: Descomposición, Comunicación, Aglomeración y Mapeo. Herramientas de desarrollo. Herramientas basadas en el modelo de memoria compartida, Herramientas basadas en el modelo de memoria distribuida, Modelos híbridos. Características del procesamiento paralelo en los lenguajes de programación y los sistemas operativos. Características del desarrollo de aplicaciones paralelas. Concepto de High Performance Computing (HPC).

#### **Unidad 2: Arquitecturas Paralelas**

Clasificación de arquitecturas paralelas. Por Organización lógica: mecanismo de control (Taxonomía de Flynn: SISD, SIMD, MISD y MIMD) y modelos de comunicación, (arquitecturas de memoria compartida UMA-NUMA y de memoria distribuida). Por Organización física: organización del espacio de direcciones, redes de interconexión y granularidad de los procesadores.

Evolución y Tendencias en las arquitecturas paralelas. Multiprocesadores, Clusters, Multiclusters, Grid, Multicores (homogéneos, heterogéneos y asimétricos), Manycores (GPUs y otros coprocesadores), Cloud para HPC. Top 500 y Green 500.

Redes de comunicación en sistemas paralelos y distribuidos orientadas a HPC: Ethernet, Infiniband, Myrinet.



### **Unidad 3: Sistema de memoria**

Sistema de memoria. Limitaciones. La importancia de la memoria caché. Conceptos asociados a la memoria cache (Estados, Niveles, Contención y Coherencia) Principio de localidad: Localidad Temporal y Espacial. Principio de localidad aplicado a la multiplicación de matrices Minimizando el espacio de memoria. Caso de estudio matrices triangulares. Conceptos de contadores hardware para evaluar el comportamiento del sistema de memoria.

### **Unidad 4: Principios de diseño de algoritmos paralelos**

Paralelismo explícito. Etapas de diseño de algoritmos paralelos: Descomposición, Comunicación, Aglomeración y Mapeo. Etapa de Descomposición. Descomposición en tareas: Descomposición de Datos o de Dominio y Descomposición Funcional. Dependencias de tareas, Grafo de dependencias. Embarrassingly Parallel Problems. Características de las tareas: Tareas estáticas y dinámicas, Comportamiento uniforme y no uniforme, Volumen de tareas y Granularidad. Técnicas de descomposición de datos: recursiva, basada en datos (de salida, de entrada - problema de la frontera, de entrada/salida y de datos intermedios), exploratoria y especulativa. Técnica de descomposición híbrida. Etapa de asignación/aglomeración. Reducción del volumen de tareas. Asignación, balance de carga. Aglomeración, localidad. Etapa de Comunicación. Patrones de comunicación: Global/Local, Estructurado/No Estructurado, Estático/Dinámico, Síncrono/Asíncrono. Overhead de las comunicaciones. Etapa de Aglomeración. Reducción de overhead y optimización de la localidad. Conservación de la escalabilidad. Costos asociados a la Ingeniería de Software. Etapa de Mapeo. Mapeo estático. Estrategias de mapeo estático: basado en datos (arrays y grafos), en particiones de tareas y jerárquico. Mapeo dinámico. Estrategias de mapeo dinámico: centralizados y distribuidos. Mapeo dinámico y balance de carga. Diseño sobre arquitecturas modernas. Heterogeneidad e integración: arquitecturas, sistemas operativos, herramientas y compiladores. Afinidad.

### **Unidad 5: Programación de algoritmos paralelos. Modelo de memoria compartida**

Conceptos del Modelo de Programación sobre Memoria Compartida. Concepto de thread. Herramientas de desarrollo Posix Threads (Pthreads) y OpenMP. Pthreads: Gestión de hilos, Sincronización, Afinidad. OpenMP: modelo Fork-Join, Estructura de control paralela, Distribución de trabajo entre hilos, Gestión de entorno de datos, Sincronización, Funciones y variables de ambiente, Afinidad. Resolución de aplicaciones específicas.



### **Unidad 6: Programación de algoritmos paralelos. Modelo de memoria distribuida**

Conceptos del Modelo de Programación sobre Memoria Distribuida. Antecedentes: Herramienta de desarrollo PVM. Herramientas de desarrollo MPI: Funcionamiento y Estructura de programa, Comunicación Punto a Punto y Colectiva. Ocultamiento de la latencia.

Resolución de aplicaciones específicas.

### **Unidad 7: Programación de algoritmos paralelos. Modelo híbrido**

Arquitecturas Paralelas Híbridas y desafíos en la programación paralela. Modelos de Programación Híbridos: MPI-Pthreads y MPI-OpenMP. Compilación y Modularidad.

Resolución de aplicaciones específicas.

### **Unidad 8: Métricas**

Importancia de las métricas en los sistemas paralelos. Métricas de rendimiento, Speedup, Speedup Absoluto y Relativo. Speedup lineal, Speedup perfecto, Speedup superlineal. Eficiencia. Overhead. Balance de carga. Escalabilidad en los sistemas paralelos: Ley de Amdahl y Ley de Gustafson, Definición de Escalabilidad, Modelo de Isoeficiencia.

Métricas en arquitecturas heterogéneas. Concepto de potencia de cómputo relativa (PCR). Limitaciones del speedup.

Ejecución Paramétrica.

### **Unidad 9: Paradigmas**

Paradigmas de programación paralela: Paralelismo de Datos SIMD/SPMD, Divide y Vencerás, Pipelines, Algoritmos sistólicos, Master-Worker, Task pools. Modelos híbridos.

Ejemplos de resolución de aplicaciones específicas.

### **Unidad 10: Introducción a la programación sobre GPUs**

Era Manycore - Introducción al concepto de GPGPU. Características paralelas de las GPUs.

Arquitectura Nvidia: Sistema de procesamiento y Sistema de memoria. Modelo de programación Nvidia CUDA: Declaración de variables, Gestión de la memoria global, Gestión de hilos, Kernel, Modularidad, Variables Built-in y Thread ID, Planificación. Optimizaciones Nvidia CUDA.

Ejemplos de resolución de aplicaciones específicas.



### **Bibliografía**

- “An Introduction to Parallel Computing. Design and Analysis of Algorithms” Grama, Gupta, Karypis, Kumar, 2003.
- “Foundations of Multithreaded, Parallel and Distributed Programming” Andrews, 2000.
- “Introduction to HPC for Scientists and Engineers” Hager, Wellein, 2011.
- “An introduction to parallel programming” Peter Pacheco, 2011.
- “Parallel Programming for Multicore and Cluster Systems” Rauber, 2010.
- “Parallel programming in C with MPI and OpenMP” Quinn, 2003.
- “Sourcebook of Parallel Computing” Dongarra, Foster, Fox, 2003.



## **Metodología de Enseñanza**

### **Modalidad Presencial**

La asignatura se estructura con clases teóricas y clases prácticas experimentales.

- Las clases teóricas introducen los conceptos teóricos de la asignatura aplicables en las clases prácticas.
- Las clases prácticas experimentales se desarrollan en una de las Salas de Cómputo de la Facultad y equipamiento especial del III-LIDI accedido de forma remota. Estas clases consisten de un conjunto de problemas que los alumnos deben resolver sobre las arquitecturas disponibles. Durante los días de práctica pueden incluirse explicaciones introductorias al trabajo en la sala y relacionadas a la utilización del equipamiento.
- Las consultas se realizarán durante los días de clases prácticas. Eventualmente, se responderán consultas de forma asincrónica mediante el entorno IDEAS cuando involucren dudas puntuales de menor complejidad.

### **Redictado**

La cátedra no ve necesidad alguna de realizar un redictado.

### ***Modalidad virtual de situación excepcional***

*En caso de extrema necesidad se seguirá una modalidad virtual. Las clases y las explicaciones relacionadas a las prácticas experimentales se dictarán mediante alguna plataforma virtual (Webex o similar). Las consultas podrán hacerse sincrónicamente a través de la plataforma virtual o asincrónicamente a través de la mensajería de IDEAS. Las prácticas experimentales se llevarán a cabo sobre equipamiento con acceso remoto provisto por la cátedra.*

### **Evaluación**

La cursada se aprueba a partir de un trabajo experimental integrador y una defensa del mismo.

El trabajo es individual y consiste en resolver uno o más problemas específicos sobre arquitecturas paralelas, utilizando las herramientas vistas en clase y realizando el análisis de escalabilidad de las soluciones propuestas.

Para los trabajos entregados que no cumplan con los requerimientos mínimos, establecidos oportunamente por la cátedra, habrá una única re-entrega. Una vez que la cátedra considere satisfactorio el trabajo se pasará a la instancia de defensa.



La defensa del trabajo experimental consiste en contestar preguntas relacionadas al trabajo y soluciones relacionadas a las prácticas experimentales. En cada caso, la cátedra determinará si se realiza mediante un coloquio oral o de manera escrita. Cabe aclarar que la defensa no tiene instancia recuperatoria.

La materia se aprueba mediante un examen final que incluye los temas del programa analítico vistos durante la cursada. Serán eximidos de este examen quienes aprueben la promoción.

### **Régimen de Promoción**

Para acceder a la promoción se deben cumplir los siguientes requisitos:

- Estar inscripto en modalidad Promoción.
- 80% de asistencia a las clases teóricas.
- 80% de asistencia a las clases prácticas.

*Quienes cumplan con lo anterior podrán realizar el trabajo experimental integrador en grupo de a lo sumo dos personas.*

- Aprobar la cursada sin haber re-entregado el trabajo experimental integrador.
- Aprobar con nota 6 o más un examen de promoción. El mismo incluye los temas del programa analítico vistos durante la cursada y no tiene instancia recuperatoria.
- Tener aprobadas las materias correlativas antes de diciembre del corriente año.

La nota final para la promoción se computa como promedio entre la nota del trabajo, la defensa del mismo y la nota del examen de promoción.

### **Cronograma de clases y evaluaciones**

Cronograma de clases teóricas:

Clase	Fecha	Contenidos/Actividades
1	Semana del 6/3	Unidad 1 (Conceptos básicos) – Unidad 2 (Arquitecturas Paralelas - Clasificación)
2	Semana del 13/3	Unidad 2 (Arquitecturas Paralelas – Evolución y Tendencias)
3	Semana del 20/3	Unidad 3 (Sistema de memoria)
4	Semana del 27/3	Unidad 4 (Principios de Diseño de algoritmos paralelos)
5	Semana del 3/4	Unidad 5 (Programación de algoritmos paralelos. Modelo de memoria compartida) Pthreads
6	Semana del 10/4	Unidad 5 (Programación de algoritmos paralelos. Modelo de memoria compartida) OpenMP



**UNIVERSIDAD NACIONAL DE LA PLATA**  
**FACULTAD DE INFORMÁTICA**

7	Semana del 17/4	Unidad 6 (Programación de algoritmos paralelos. Modelo de memoria distribuida) MPI Unidad 7 (Programación de algoritmos paralelos. Modelo híbrido)
8	Semana del 24/4	Unidad 8 (Métricas)
9	Semana del 1/5	Unidad 9 (Paradigmas)
10	Semana del 8/5	Unidad 10 (Introducción a la programación sobre GPUs)

Cronograma de prácticas experimentales:

Clase	Fecha	Contenidos/Actividades
1	Semana del 27/3	Práctica 1 – Optimización de algoritmos secuenciales
2	Semana del 3/4	Práctica 1 – Optimización de algoritmos secuenciales
3	Semana del 10/4	Práctica 2 – Modelo de memoria compartida - Pthreads
5	Semana del 17/4	Práctica 2 y 3 – Modelo de memoria compartida – Pthreads y OpenMP
6	Semana del 24/4	Práctica 4 – Modelo de memoria distribuida - MPI
7	Semana del 1/5	Práctica 4 y Práctica 5 – Modelo de memoria distribuida – MPI e Híbridos
8	Semana del 8/5	Práctica 5 –Introducción a la programación sobre GPUs

Cronograma de entregas y defensas:

Fecha tentativa	Contenidos/Actividades
Semana del 8/5	Presentación del trabajo a entregar
Semana del 29/5	Entrega
Semana del 12/6	Resultados de Entrega
Semanas del 12/6 y 19/6	Defensa del trabajo a quienes cumplieron satisfactoriamente con la Entrega
Semana del 19/6	Re-entrega
Semana del 26/6	Resultados de Re-entrega
Semanas del 26/6 y 3/7	Defensa del trabajo a quienes cumplieron satisfactoriamente con la Re-Entrega

Evaluaciones:

Fecha tentativa	Evaluación previstas
12/07/2023	Examen de Promoción

**Contactos de la cátedra:**

Mail: [apousa@lidi.info.unlp.edu.ar](mailto:apousa@lidi.info.unlp.edu.ar)

Sitio WEB: -

Plataforma virtual: IDEAS (<https://ideas.info.unlp.edu.ar>), Webex (<https://unlpeducar.webex.com>)

Otros: -

Firma del profesor responsable