



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

---

## PROGRAMACIÓN DISTRIBUIDA

Carrera: *Licenciatura en Informática*  
*Plan 2003-2007*  
*Licenciatura en Sistemas*  
*Plan 2003-2007*

**Año 2013**

Año:  
Duración: **Semestral**  
Profesor: **Dr. Fernando G. Tinetti**  
Hs semanales: 6 hs

---

### FUNDAMENTACION:

Se presenta el contexto y los conceptos básicos de las herramientas de programación en los entornos distribuidos. Agrega una visión de mayor nivel de abstracción de sistemas operativos y de redes combinados con conceptos propios de sistemas distribuidos.

### OBJETIVOS GENERALES:

Analizar las arquitecturas de procesamiento distribuido y los mecanismos de comunicación y sincronización entre procesos. Integrar los conocimientos anteriores con el manejo de datos distribuidos. Desarrollar prácticas experimentales sobre redes LAN y WAN con ambientes de desarrollo orientados a la programación distribuida.

### CONTENIDOS MINIMOS:

- Procesamiento distribuido. Modelos y paradigmas.
- Distribución de datos y procesos.
- Bases de datos distribuidas.
- Migración de procesos en ambientes distribuidos.
- Programación de aplicaciones en ambientes distribuidos.

### Programa

**Unidad 1.- Sistemas Distribuidos.** Conceptos introductorios. Motivación. Definiciones. Caracterización y problemas a resolver. Arquitecturas de procesamiento paralelo y su evolución a entornos/arquitecturas distribuidos. Redes de interconexión. Modelos de procesamiento distribuido. Distribución de datos y procesos. Ejemplos de utilización actual.

**Unidad 2.- Patrones Básicos de Interacción de Procesos.** Productores/Consumidores: relaciones con pipelines de procesamiento y filtros de los sistemas operativos. Cliente/servidor: conceptos, ejemplos, sistemas operativos distribuidos, relación con Internet,



relación con threads. Interacción entre pares (peer-to-peer): relación con cómputo paralelo clásico, identificación de áreas de aplicación.

**Unidad 3.- Pasaje de Mensajes.** Pasaje de mensajes asincrónicos. Clientes y servidores. Interacción entre pares. Pasaje de mensajes sincrónicos, implementación de mensajes asincrónicos con mensajes sincrónicos. Redes y sockets. Ejemplos con lenguajes y bibliotecas como C, MPI y Java.

**Unidad 4.- RPC (Remote Procedure Call, Llamada a Procedimiento Remoto) y Rendezvous.** Concepto de RPC, extensión del modelo de ejecución procedural tradicional, relación con módulos de programas. Definición de rendezvous, ejemplos cliente/servidor e interacción entre pares. Ejemplos con lenguajes de programación como Ada y Java.

**Unidad 5.- Problemas/Conceptos Relacionados.** Bases de datos distribuidas: motivación, antecedentes, relación con programación distribuida, consultas y transacciones distribuidas, migración y replicación de datos. Modelo de cómputo/aplicaciones N-tier. Migración de procesos en entornos distribuidos, algunas ideas implementadas en Condor. DSM (Distributed Shared Memory, memoria compartida físicamente distribuida): motivación, implementaciones. Otros modelos y lenguajes: Bag-of-Tasks (bolsa de tareas), Manager/worker (manejador/trabajador), Master/worker (maestro/trabajador), Algoritmos de heartbeat y/o sistólicos, Algoritmos de broadcast, Servidores múltiples, lenguajes Linda y OpenMP, modelo BSP.

### **Metodología de enseñanza**

La modalidad de enseñanza incluye:

- 1) Clases teóricas, normalmente guiadas a partir de diapositivas proyectadas y explicaciones de algunos detalles en pizarrón. Se presentan respuestas a los alumnos utilizando el pizarrón y en algunos casos ejemplos funcionando sobre un sistema distribuido para aclaración de conceptos.
- 2) Clases prácticas, mayormente para: a) presentación de los temas de los trabajos prácticos, y b) consultas de los trabajos prácticos.
- 3) La actividad de los trabajos prácticos se presenta a través de informes que se elaboran de manera grupal o individual y se evalúan de manera individual y oral.

### **Propuesta de evaluación**

La evaluación se lleva a cabo mediante exámenes parciales y final.

Los exámenes parciales consisten en la defensa de los informes correspondientes a los trabajos prácticos. Esta defensa es oral e individual a partir de los informes y de los programas que los alumnos desarrollan para la resolución de los trabajos prácticos. El examen final es oral e incluye todos los temas presentados en la asignatura. Cuando los alumnos deciden llevar a cabo un trabajo final integrador, el examen se suele concentrar en la defensa del trabajo y la justificación de cada una de las decisiones tomadas en el mismo a partir de los conceptos vistos de la asignatura.



### **Bibliografía Obligatoria**

- ◆ *Sistemas Distribuidos. Conceptos y Diseño*, George Coulouris, Jean Dollimore, Tim Kindberg. 3ra Edición, Pearson Educación, S. A., Madrid, 2001. ISBN 84-7829-049-4.
- ◆ *Distributed Systems: Principles and Paradigms*, Andrew S. Tanenbaum, Prentice Hall, 1st edition, ISBN: 0130888931, 2002.
- ◆ *Distributed Computing: Principles and Applications*, M. L. Liu, Addison-Wesley, 2004.

### **Bibliografía Complementaria**

- ◆ *Foundations of Multithreaded, Parallel, and Distributed Programming*, Gregory R. Andrews. Addison-Wesley, 2000. ISBN 0-201-35752-6.
- ◆ *Distributed Operating Systems*, Andrew S. Tanenbaum, Prentice Hall; 1st edition, ISBN: 0132199084, 1994.

### **Cronograma de Clases y Evaluaciones**

Clase 1: Introducción a los sistemas distribuidos, desde la perspectiva de la programación en ese entorno. Características comunes. Relación de los temas de programación a desarrollar con Redes y Sistemas Operativos.

Clase 2: RPC (Remote Procedure Call) como primer mecanismo de programación distribuida. Evolución hacia web services. Características de lenguajes utilizados (implementación) y flexibilidad. Impacto de la concurrencia. Presentación del primer trabajo práctico.

Clase 3: Ventajas y desventajas de los sistemas distribuidos respecto de los centralizados. Tipos de sistemas distribuidos. Caracterización del modelo de procesamiento y programación de los sistemas cliente/servidor. Consultas para la resolución del primer trabajo práctico.

Clase 4: Comparación entre Sistema Operativos Distribuidos, Sistemas Operativos de Red y definición de *Middleware* (capas intermedias de software). Arquitecturas/modelos de sistemas distribuidos y su impacto en la programación: 2, 3,  $n$  capas, desacoplados en espacio, desacoplados en espacio y tiempo. Consultas para la resolución del primer trabajo práctico.

Clase 5: RMI (Remote Method Invocation) como mecanismo de programación en Java y utilizado por sistemas programados en Java. Comparación con RMI. Impacto de la concurrencia. Análisis de transparencia. Presentación del segundo trabajo práctico.



Clase 6: Comunicaciones en sistemas distribuidos y su impacto en la programación. Relación con el modelo de capas ISO/OSI. Aspectos de abstracción, persistencia y sincronismo que impactan en la programación de los sistemas distribuidos. Consultas para la resolución del segundo trabajo práctico.

Clase 7: Movilidad de código: motivación/diferencia respecto de la movilidad de datos. Tipos de movilidad. Implementaciones actuales y la relación con el manejo de procesos y threads de los sistemas operativos. Relación de la movilidad de código con la virtualización. Manejo de recursos en general, relacionados con la movilidad de código. Consultas para la resolución del segundo trabajo práctico.

Clase 8: Sincronización en sistemas distribuidos y su impacto en la programación o en el entorno de programación. Definición y motivaciones. Diferencia entre ordenamiento de eventos y sincronización de los relojes. Análisis desde la perspectiva de las relaciones de orden. Presentación del tercer trabajo práctico.

Clase 9: Algoritmos específicos de sincronización. Análisis de la implementación de algoritmos de sincronización lógica y física. Relación de la implementación de los algoritmos con las cotas de error o con la definición de eventos concurrentes. Análisis de los estados globales, y la evaluación formal de eventos o predicados sobre eventos. Consultas para la resolución del tercer trabajo práctico.

Clase 10: Aspectos de la programación de dispositivos móviles. Evolución asociada a la evolución de la comercialización y capacidad de dispositivos. Análisis de los entornos de programación e impacto en la implementación. Consultas para la resolución del tercer trabajo práctico.

Clase 11: Nombres en sistemas distribuidos: definiciones, evolución, motivación y análisis de la implementación de algunos sistemas de nombres. Análisis del sistema DNS (Domain Name System) como sistema de nombres en sistemas distribuidos. Aspectos de transparencia relacionados con la programación de los sistemas distribuidos. Presentación del cuarto trabajo práctico.

Clase 12: Comentarios y análisis extendido de los aspectos de concurrencia, paralelismo y fallas relacionados con los sistemas vistos en la práctica. Consultas para la resolución del cuarto trabajo práctico.

Examen Parcial. Incluye la evaluación de los trabajos prácticos.

1er. Recuperatorio del Examen Parcial. Incluye la evaluación de los trabajos prácticos.

2do. Recuperatorio del Examen Parcial. Incluye la evaluación de los trabajos prácticos.



UNIVERSIDAD NACIONAL DE LA PLATA  
FACULTAD DE INFORMÁTICA

---

**Contacto de la Cátedra:** [pd@ada.info.unlp.edu.ar](mailto:pd@ada.info.unlp.edu.ar)

**Firma de los profesores responsables:**