

CONCURRENCIA Y PARALELISMO**Carrera/ Plan:**

Ingeniería en Computación Plan 2008/2011

Año: 4**Régimen de Cursada:** Semestral**Carácter (Obligatoria/Optativa):** Obligatoria**Correlativas:**

Plan 2008: Taller de lenguajes II-Redes de Datos 1.

Plan 2011: Taller de lenguajes II-Conceptos de Sistemas operativos.

Año 2026

Profesor/es: Franco Chichizola**Hs. semanales:** 6**FUNDAMENTACIÓN**

La temática de la Concurrencia es central en el desarrollo actual de la Ciencia Informática, en particular por el creciente desarrollo de arquitecturas multiprocesador que permiten implementar físicamente los conceptos teóricos de concurrencia "real".

El impacto de la concurrencia se refleja en diferentes ámbitos de la disciplina tales como las arquitecturas, los sistemas operativos, los lenguajes y el diseño y desarrollo de aplicaciones. En este sentido, se impone que los futuros profesionales sean capaces de desarrollar soluciones que utilicen adecuadamente la tecnología disponible con fundamentos teóricos firmes.

OBJETIVOS GENERALES

Dar los conceptos fundamentales de Concurrencia en software. Analizar la semántica y sintaxis para especificar concurrencia. Estudiar la sincronización de procesos concurrentes por memoria compartida y mensajes. Desarrollar estudios de casos con diferentes lenguajes/ herramientas para concurrencia.

EJES TRANSVERSALES

01. Diseño e implementación de diversas Arquitecturas de Computadoras y todos los subsistemas relacionados. (1-Bajo)
03. Especificación, proyecto y Desarrollo de Software y Sistemas Conjuntos de Hardware y Software haciendo uso de conceptos, métodos y herramientas de gestión de proyectos, ingeniería de software, base de datos, experiencia del usuario, elicitación, análisis, especificación y validación de requerimientos. (2-Medio)
04. Desarrollo de Redes de Computadoras y de Redes de Computadoras de área amplia, locales, inalámbricas y móviles. (1-Bajo)
05. Sistemas de Gestión de Recursos de Hardware y Software a sistemas generales, de tiempo real, distribuidos, para dispositivos fijos y móviles. (2-Medio)
06. Proyecto, desarrollo, dirección, control, construcción, operación y mantenimiento de Sistemas de Procesamiento de Señales, Sistemas Embebidos y sus periféricos incluido en software de soporte, Sistemas Computarizados de automatización y control y Sistemas Conjuntos de Hardware y Software. (2-Medio)
07. Certificación del funcionamiento, condición de uso o estados de Sistemas de Procesamiento de Señales, Sistemas Embebidos, Sistemas Computarizados de automatización y control, Sistemas Conjuntos de Hardware y Software. (1-Bajo)
08. Proyecto, Dirección y Aseguramiento de la calidad en lo referido a Seguridad Informática. (1-Bajo)
09. Identificación, formulación y resolución de problemas de ingeniería en computación. (2-Medio)
10. Concepción, diseño y desarrollo de proyectos de ingeniería en computación. (1-Bajo)
12. Utilización de técnicas y herramientas de aplicación en la ingeniería en computación. (2-Medio)
13. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas. (1-Bajo)
14. Fundamentos para el desempeño en equipos de trabajo. (1-Bajo)

CONTENIDOS MINIMOS (de acuerdo con el Plan de Estudios)

- *Especificación de la ejecución concurrente.*
- *Comunicación y sincronización.*
- *Concurrencia con variables compartidas.*
- *Concurrencia con pasajes de mensajes.*
- *Lenguajes de programación concurrente.*
- *Introducción a los conceptos de procesamiento paralelo.*

PROGRAMA ANALÍTICO

1. Conceptos básicos

- *Objetivos de los sistemas concurrentes.*
- *Procesamiento secuencial, concurrente y paralelo. Características.*
- *Concurrencia y paralelismo. Relación con la arquitectura.*
- *Sincronización (por exclusión mutua y por condición) y comunicación (por memoria compartida y por mensajes).*

2. Concurrencia y sincronización

- *Especificación y semántica de la ejecución concurrente. La sentencia “co” y “process”*
- *Acciones atómicas y sincronización.*
- *El problema de interferencia. Historias válidas e inválidas.*
- *Atomicidad de grano fino y de grano grueso.*
- *La propiedad de “A lo sumo una vez”.*
- *La sentencia “await”. Semántica. Especificación de la sincronización.*
- *Técnicas para evitar interferencia.*
- *Propiedades de seguridad y vida.*
- *Políticas de scheduling y fairness.*

3. Concurrencia con variables compartidas

• Sincronización por variables compartidas

- *Sincronización de grano fino.*
- *Secciones críticas (SC). Definición del problema. Propiedades necesarias de las soluciones. Planteo de soluciones clásicas.*
- *Sincronización Barrier. Definición del problema. Planteo de soluciones.*

• Sincronización por semáforos

- *Defectos de la sincronización por variables compartidas.*
- *Semáforos. Sintaxis y semántica.*
- *Usos básicos y técnicas de programación con semáforos.*

• Sincronización por monitores

- *Evolución histórica a partir de semáforos.*
- *Monitores. Sintaxis y semántica.*
- *Sincronización en monitores. Disciplinas de señalización: “Signal and wait” y “Signal and continue”.*
- *Usos básicos y técnicas de programación con monitores.*

• Lenguajes para programación con variables compartidas

- *Pthreads.*
- *OpenMP..*

4. Programación distribuida. Concurrencia con pasaje de mensajes

- *Programas distribuidos. Relación entre mecanismos de comunicación.*
- Pasaje de mensajes asincrónicos (PMA)**
 - *Sintaxis y semántica. Canales. Operaciones.*
 - *Usos básicos y técnicas de programación con Pasaje de Mensajes Asincrónicos.*
- Pasaje de mensajes sincrónicos (PMS)**
 - *Sintaxis y semántica.*

- Conceptos de CSP. Comunicación guardada. Sintaxis y semántica.
- Usos básicos y técnicas de programación con Pasaje de Mensajes Sincrónicos.
- **Remote Procedure Calls (RPC) y Rendezvous.**
 - Sintaxis y semántica.
 - Similitudes y diferencias.
 - Usos básicos y técnicas de programación con RPC y Rendezvous.
- **Lenguaje para Programación distribuida**
 - Librería para manejo de mensajes – MPI.
 - Lenguaje para Rendezvous - ADA.

5. Introducción a la Programación Paralela

- Objetivos del procesamiento paralelo.
- Necesidad del paralelismo.
- Concepto de Sistema Paralelo.
- Diseño de algoritmos paralelos.
- Métricas de rendimiento de los sistemas paralelos (speedup y eficiencia).
- Concepto de asignación de tareas y balance de carga.
- Balance de carga estático y dinámico.
- Arquitecturas orientadas a Procesamiento Paralelo.

6. Paradigmas de cómputo paralelo

- Objetivo de los paradigmas de resolución de programas paralelos.
- Paradigma Master/Slave.
- Paradigma Divide/Conquer.
- Paradigma Pipelining.
- Otros paradigmas.

En la práctica se realiza trabajo experimental sobre arquitecturas paralelas utilizando los lenguajes y librerías enseñados (Pthreads, OpenMP, MPI, ADA).

BIBLIOGRAFÍA

• Principal

- Andrews G. "Foundations of Multithreaded, Parallel and Distributed Programming", Addison Wesley, 2000.
- Grama A., Gupta A., Karypis G., Kumar V. "An Introduction to Parallel Computing. Design and Analysis of Algorithms", Pearson Addison Wesley, 2nd Edition, 2003.

• Complementaria

- Barnes J. "Programming in Ada 2005 with CD", Addison Wesley, 2006.
- Ben-Ari, M. "Principles of Concurrent and Distributed Programming, 2/E". Addison-Wesley. 2006. ISBN 0-321-31283-X.
- Downey, A. "The Little Book of Semaphores, Second Edition". Free book disponible en <http://www.freetechbooks.com/the-little-book-of-semaphores-second-edition-t519.html>, 2007.
- Ghosh, S. "Distributed Systems: An Algorithmic Approach". Chapman & Hall/CRC, 2007. ISBN1584885645, 9781584885641.
- Habermann A., Perry D. "Ada for Experienced Programmers", Addison-Wesley Publishing Company. 1993.
- Herlihy M., Shavit N. "The Art of Multiprocessor Programming". Morgan Kaufmann, 2008.
- Hoare C. "Communicating Sequential Processes", Englewood Cliffs, Prentice Hall, 1985
- Jordan H.F., Alaghand G., Jordan H.E. "Fundamentals of Parallel Computing", Prentice Hall, 2002.
- Leopold C. "Parallel and Distributed Computing. A survey of Models, Paradigms, and Approaches", Wiley Series on Parallel and Distributed Computing. Albert Zomaya Series Editor, 2001.
- Pacheco, P. "An introduction to parallel programming". Morgan Kaufmann, 2011.
- Raynal M. "Concurrent Programming: Algorithms, Principles, and Foundations". Springer, 2012.
- Snir, M., Otto, S., Huss-Lederman, S., Walker, D., Dongarra, J. "MPI: The Complete Reference". Cambridge, MA: MIT Press, 1996. Available in web site: <http://www.netlib.org/utk/papers/mpi-book/mpi-book.html>.
- Taubenfeld, G. "Synchronization Algorithms and Concurrent Programming". Prentice Hall. 2006.

METODOLOGÍA DE ENSEÑANZA

En la cátedra se pone énfasis en el proceso de identificación de problemas del mundo real, la especificación de los mismos como problemas resolubles desde la Ingeniería en Computación y en el desarrollo de soluciones verificables para los mismos.

Se trata de poner al alumno en el contexto de aplicación en el campo de la Ingeniería en Computación de los conceptos y métodos que se encuentran en el programa de la asignatura. Esta contextualización es informativa y se discuten diferentes casos de aplicación para mostrar la utilidad de las teorías y herramientas matemáticas para resolver diferentes problemas “informáticos” conocidos por el alumno. Se pone énfasis en la capacidad del alumno para conocer técnicas y herramientas de aplicación en Ingeniería en Computación (en lo posible siguiendo las tendencias marcadas por el cambio tecnológico) y en la aplicación efectiva de las mismas, tanto en sistemas donde se comparten recursos de hardware o software (de tiempo real, distribuidos, etc.) como en problemas de alta demanda computacional.

La cátedra acompaña el proceso de aprendizaje del alumno brindando materiales para que el alumno estudie casos y valore la selección y empleo eficiente de herramientas y técnicas determinadas para cada problema en diferentes modelos de arquitecturas paralelas (máquinas de memoria compartida y diferentes tipos de redes de computadoras).

Se plantean actividades relacionadas con las tecnologías existentes para diferentes problemas y se los “desafía” a presentar la posible evolución de la solución para ese tipo de problema y en qué aspectos podría mejorarse la solución/soluciones actuales. Para esto el alumno debe buscar bibliografía relacionada con el cambio tecnológico y formarse un criterio sobre las tendencias (por ejemplo, en los procesadores a utilizar, el tipo de topología de red o los lenguajes y librerías para programación concurrente/paralela, etc)

*Se utiliza un entorno virtual de enseñanza-aprendizaje (**ideas**), donde se encuentran disponibles clases, guías de TP, avisos, resultados de exámenes, etc. Se incluye como material de las clases los archivos PDF de las teorías y las explicaciones prácticas.*

La asignatura se estructura con clases teóricas, explicaciones prácticas y consultas prácticas:

- Las **clases teóricas** son dictadas por el profesor de la asignatura. En estas clases se explican los conceptos teóricos de la materia, el funcionamiento de las herramientas para trabajar con memoria compartida y distribuida, y las técnicas de programación para resolver problemas con esas herramientas. Periódicamente se publicarán en IDEAS las clases en PDF para que puedan ser accedidas asincrónicamente por los estudiantes. Se recomienda a los estudiantes haber visto previamente el material para un efectivo aprovechamiento de las clases teóricas.
- Las **explicaciones de práctica** son dictadas por el profesor de la asignatura. En ellas se realiza un repaso del funcionamiento de la herramienta correspondiente además de explicar las soluciones correctas y los errores frecuentes en ejercicios semejantes a los de los trabajos prácticos. Periódicamente se publicarán en IDEAS las clases en PDF para que puedan ser accedidas asincrónicamente por los estudiantes.
- Las **clases de consulta prácticas** son dictadas por auxiliares docentes y sirven para que los alumnos trabajen/consulten sobre los ejercicios propuestos en la guía de trabajos prácticos. En algunas de ellas implica el desarrollo de grupo de trabajos de laboratorio con diferentes arquitecturas paralelas y lenguajes/librerías de programación.

El reglamento y cronograma tentativo son conocidos por los alumnos desde el inicio de la cursada.

Para las clases teóricas se utiliza PC, cañón y pizarrón.

Los trabajos de Laboratorio son realizados sobre diferentes arquitecturas paralelas y lenguajes/librerías de programación paralela.

EVALUACIÓN

La cátedra acompaña el proceso de aprendizaje del alumno, para contrastar sus conclusiones y validar su habilidad para interpretar la evolución tecnológica y su visión de las tendencias futuras.

En la evaluación de las competencias en las pruebas parciales y finales se tienen en cuenta los siguientes aspectos:

- la capacidad para identificar, formular y resolver los problemas, reflejándolo en la corrección de las pruebas escritas
- la capacidad para conocer e interpretar los conceptos, teorías y métodos, aplicándolos a problemas concretos. Esto se evalúa a través de preguntas del tipo “donde cree Ud. que es aplicable este conocimiento o método”, o la interpretación de código, o interpretación de resultados

-
- en qué medida el alumno es capaz de utilizar de manera efectiva las técnicas y herramientas que son parte de la asignatura
 - los resultados del estudio bibliográfico y capacidad para formular las ventajas potenciales del cambio tecnológico en los problemas planteados, plasmando esto en una planilla.

Para aprobar la cursada de la materia, los alumnos deben aprobar un **examen práctico parcial que tiene dos temas (Memoria Compartida y Memoria Distribuida)** para el que dispone de una primera fecha dividida (un día diferente para cada tema) y dos recuperatorios globales (se recuperan el/los temas pendientes).

Además, se cuenta con un **examen teórico optativo**.

Para promocionar la materia la suma del puntaje obtenido en cada uno de los dos temas aprobados (máximo de 4 puntos para cada uno de ellos) y el puntaje del **examen teórico optativo** (máximo de 2 puntos) debe ser mayor o igual a 6. El no rendir o no aprobar el **examen teórico optativo** no impide obtener la promoción, sólo se considera 0 el puntaje de ese ítem.

En caso de sólo obtener la cursada, se debe rendir un examen teórico/práctico para aprobar la materia en alguna de las mesas de finales.

CRONOGRAMA DE CLASES Y EVALUACIONES

Clase	Fecha	Contenidos/Actividades
1 - 3	12/3/2026 19-3/2026	Conceptos básicos y variables compartidas.
4 - 5	20/3/2026 26/3/2026	Semáforos
6 - 8	27/3/2026 10/4/2026	Monitores
9 - 13	17/4/2026 7/5/2026	Pasaje de Mensajes Asíncronos y Sincrónicos
14 - 15	15/5/2026 21/5/2026	RPC y Rendezvous
16 - 17	22/5/2026 29/5/2026	Introducción al Paralelismo

Intercaladas a estas clases son las consultas prácticas.

Evaluaciones previstas	Fecha
Parcial práctico (Memoria Compartida)	08/05/2026
Parcial práctico (Memoria Distribuida)	12/06/2026
Primer recuperatorio práctico (ambos temas)	26/06/2026
Segundo recuperatorio práctico (ambos temas)	17/07/2026
Examen teórico optativo de Promoción	06/08/2026

Contacto de la cátedra (mail, sitio WEB, plataforma virtual de gestión de cursos):

- **Mail:** francoch@lidi.info.unlp.edu.ar
- **Sitio WEB:** <http://weblidi.info.unlp.edu.ar/catedras/concurrenciayparalelismo/index.html>
- **Plataforma virtual:** ideas.info.unlp.edu.ar
- **Otros:** --

Firma del profesor

