



# Programación concurrente

**Carrera:** Licenciatura en Informática

**Actividad curricular:** Programación Concurrente

**Prerrequisitos:** Estructuras de Datos

**Área:** Algoritmos y lenguajes

## Carga Horaria:

- Carga horaria total 64 horas
- Carga horaria práctica: 48 hs
  - Formación Experimental: 8hs
  - Resolución de problemas: 32hs
  - Proyecto y diseño: 8hs
- Carga horaria semanal: 4 horas por semana

## Objetivos:

Que el estudiante:

- Tenga conocimiento de los modelos más utilizados, las técnicas y conceptos básicos asociados a la programación concurrente.
- Comprenda las dificultades asociadas a la interacción de componentes en un sistema concurrente y conozca los recursos del que dispone el programador para mitigarlos.
- Pueda analizar y desarrollar programas concurrentes.
- Diseñe y Implemente un pequeño modelo de programa concurrente con un objetivo específico.

## Contenidos mínimos:

Concurrencia. Los porqués de la concurrencia. Concurrencia y paralelismo. Modelo de memoria compartida, atomicidad e independencia. Manejo de memoria en ejecución. Secciones críticas, locks y barriers, semáforos, monitores y condition variables, Rendezvous. Problemas de la concurrencia: Starvation, Deadlocks, Liveness y Progress, Safety, Raceconditions, Fairness. Modelo de pasaje de mensajes: Comunicación sincrónica vs comunicación asincrónica, Modelo de transacciones. Modelos de interacción: Cliente/Servidor, Productor/Consumidor. Aplicación de los conceptos estudiados en lenguajes de programación concretos, mecanismos de sincronización.

**Programa analítico:**

1. Conceptos básicos Objetivos de los sistemas concurrentes. Procesamiento secuencial, concurrente y paralelo. Características. Evolución histórica. El modelo de CSP (Communicating Sequential Processes). Programa concurrente. No determinismo. Multithreading. Concurrencia y paralelismo. Algoritmos concurrentes, distribuidos y paralelos. Relación con la arquitectura. Tendencias actuales en procesadores. Conceptos de arquitecturas Grid y Cloud. Sincronización y comunicación. Sincronización por exclusión mutua y por condición. Prioridad, granularidad, deadlock, manejo de recursos.

2. Concurrencia y sincronización Aspectos de programación secuencial Especificación y semántica de la ejecución concurrente. Acciones atómicas y sincronización. El problema de interferencia. Atomicidad de grano fino y de grano grueso. La propiedad de "A lo sumo una vez". Semántica. Técnicas para evitar interferencia. Propiedades de seguridad y vida. Políticas de scheduling y Fairness. Requerimientos para los lenguajes de programación. Problemas en sistemas distribuidos.

3. Concurrencia con variables compartidas Sincronización por variables compartidas. Secciones críticas (SC). Definición del problema. Algoritmos clásicos de soluciones fair al problema de la SC (tie-breaker, ticket, bakery). Implementación de sentencias Await arbitrarias. Sincronización barrier. Definición. Planteo de soluciones (contador compartido, flags y coordinadores, árboles, barreras simétricas, butterfly). Algoritmos data parallel. Sincronización por semáforos. Sintaxis y semántica. Usos básicos y técnicas de programación. Semáforos binarios divididos (split). La técnica "passing the baton". Sincronización por condición general. Alocación de recursos. Políticas de alocación. Sincronización por monitores. Monitores. Sintaxis y semántica. Disciplinas de señalización: "Signal and wait" y "Signal and continue": diferencias y efecto sobre la programación. La técnica "passing the condition". Implementaciones Conceptos de implementación de procesos en arquitecturas mono y multiprocesador. Kernel monoprocesador y multiprocesador.

4. Programación distribuida. Concurrencia con pasaje de mensajes Programas distribuidos. Definición. Relación entre mecanismos de comunicación. Clases básicas de procesos: productores y consumidores, clientes y servidores, peers. Control de concurrencia en Sistemas Distribuidos. Mensajes asincrónicos Sintaxis y semántica. Peers. Intercambio de valores. Mensajes sincrónicos Sintaxis y semántica. Ejemplos Remote Procedure Calls y Rendezvous. Sintaxis y semántica. Similitudes y diferencias. RPC: Sincronización en módulos. RPC en lenguajes reales. RMI. Ejemplos.

5. Paradigmas de interacción entre procesos distribuidos Resolución de problemas mediante diferentes paradigmas de interacción entre procesos: Servidores replicados, heartbeat, pipeline, prueba-eco, broadcast, token passing, manager/workers. Objetivos del cómputo paralelo. Necesidad del paralelismo. Areas de aplicación. Computación científica. Diseño de algoritmos paralelos.

**Bibliografía obligatoria:**

- 
- Raynal M. "Concurrent Programming: Algorithms, Principles, and Foundations". Springer, 2012
  - Concurrent Programming in Java: Design Principles and Patterns (Doug Lea)
  - Conceptos de Concurrencia y Paralelismo, NAIOUF MARCELO; DE GIUSTI ARMANDO; DE GIUSTI LAURA; CHICHIZOLA FRANCO, EDULP. Editorial de la Universidad de La Plata, 2012
- 



## Instituto de Tecnología e Ingeniería

### Bibliografía de consulta:

- Taubenfeld, Gadi. "Synchronization Algorithms and Concurrent Programming". Prentice Hall. 2006.
- Downey, Allen. "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.

### Organización de las clases:

Las clases se organizan como teórico-prácticas. Durante las clases se presenta el contenido formal y se ahonda en las características de la programación concurrente. Las clases son acompañadas por un apunte en forma de diapositivas que sirve a los alumnos como material de referencia más allá de la bibliografía sugerida.

Después de completado cada tema se resuelven ejercicios de programación haciendo hincapié en las características de la programación concurrente mediante las computadoras del laboratorio.

En la práctica se plantean ejercicios que pueden ser realizados como trabajo experimental sobre arquitecturas multiprocesador distribuidas (clusters), multiprocesadores con memoria compartida e híbridos.

El objetivo de las prácticas es completar la cursada con la implementación de un pequeño proyecto cuyo diseño se discute sobre la base de la teoría.

### Uso del campus virtual.

El Campus Virtual es un espacio fundamental para el desarrollo de la asignatura. En el aula virtual se propondrá material educativo, apuntes de clase, bibliografía así como también el programa y cronograma de la asignatura y las guías de Trabajos Prácticos y ejercicios.

### Modalidad de evaluación:

Consistirá en dos exámenes parciales con recuperatorios, según el cronograma previsto, de la totalidad de la materia descrita en el programa. Los mismos se realizarán en las fechas que establezcan en el cronograma correspondiente. Se contemplará también como nota de evaluación el trabajo práctico final de implementación.

- La calificación de cada evaluación se determinará en la escala 0 a 10, con los siguientes valores: 0, 1, 2 y 3: insuficientes; 4 y 5 regular; 6 y 7 bueno; 8 y 9 distinguido; 10 sobresaliente. La materia podrá aprobarse mediante: régimen de promoción directa, exámenes finales regulares y exámenes libres.
- Régimen de promoción directa (sin examen final): los/las estudiantes deberán aprobar las materias con siete (7) o más puntos de promedio entre todas las instancias evaluativas, sean éstas parciales o sus recuperatorios, debiendo tener una nota igual o mayor a seis (6) puntos en cada una de éstas. Todas las instancias evaluativas tendrán



**Instituto de  
Tecnología e Ingeniería**

una posibilidad de recuperación. En el caso de los ausentes en la fecha original, el recuperatorio operará como única fecha de examen. El examen recuperatorio permite mantener la chance de la promoción siempre y cuando respete las condiciones de calificación respectiva.

- Exámenes finales regulares: para aquellos/as estudiantes que hayan obtenido una calificación de al menos de 4 (cuatro) y no se encuentren en las condiciones de promoción, deberán rendir un examen final que se aprobará con una nota no inferior a 4 (cuatro) puntos.

La asistencia no debe ser inferior al 75% en las clases presenciales.