

# FACULTAD DE CIENCIAS EXACTAS, INGENIERÍA Y AGRIMENSURA

## U.N.R.

PROGRAMA ANALITICO DE LA ASIGNATURA: Sistemas Operativos I

Código R-322

**PLAN DE ESTUDIOS:** 2010

**CARRERA:** Lic. en Cs. de la Computación

**DEPARTAMENTO:** Cs. de la Computación

**PROFESORES:** Guido Macchi y Esteban Ruiz

2012 HASTA AÑO

TENTATIVO DEFINITIVO DE EXAMEN

**PROGRAMA**

ANUAL SEMESTRAL CUATRIMESTRAL

Táchese lo que no corresponda.

**OBSERVACIONES:**

**PRESUPUESTO HORARIO SEMANAL PROMEDIO**

TEORIA:	3	1
PRACTICA:	2	2
TALLER:	2	3
TOTAL ASIGNADO:	7	4 1+2+3
DEDICACION DEL ALUMNO FUERA DE CLASE:	5	5
PRESUPUESTO TOTAL:	12	6 5+4
PROGRAMA BASADO EN SEMANAS UTILES :	15	7
HORAS TOTALES ASIGNADAS:	105	7x4
HORAS TOTALES PRESUPUESTAS:	180	7x6

**OBJETIVOS:**

Nociones sólidas de los sistemas concurrentes, junto con las técnicas actuales para diseñarlos e implementarlos.

**UBICACION EN LA CARRERA Y CARACTERISTICAS GENERALES:**

Materia del tercer año de la carrera.

**MATERIAS RELACIONADAS:**

**Previas:** R-223 Lógica; R-312 Estructuras de Datos y Algoritmos II; R-313 Análisis de Lenguajes de Programación

**Simultáneas recomendadas:** - -

**Posteriores:** R-412 Sistemas Operativos II

.....  
Firma Profesor

.....  
Fecha

.....  
Aprob. Escuela

.....  
Fecha

Aprobado en reunión de Consejo Académico de fecha: .....

## CONTENIDO TEMATICO

### Ordenar temas utilizando codificación decimal

1. Programación paralela y concurrente. Definiciones. Tipos de paralelismo: cooperativo y preemptivo. Implementaciones. Pascal FC, Ada, POSIX Threads, etc. No determinismo.
2. Recursos compartidos y regiones críticas. Condiciones de concurso. Problema del Jardín Ornamental. Sincronización de procesos. Semáforos mutex: algoritmos de Dekker, Peterson y Lamport. Semáforos de Dijkstra. Problema de los Filósofos Comensales.
3. Variables de condición. Necesidad y uso. Problema de Productores y Consumidores con Buffer Acotado. Problema de la Barbería.
4. Concepto de monitores. Semánticas de Hoare y Lamport. Problemas asociados.
5. Mensajes síncronos y asíncronos. Canales. Solución de los problemas anteriores con mensajes.
6. Rendezvous. La construcción select. Semántica copy-restore. Comparación con call by reference. Alternación, terminación y guardas.
7. Deadlocks. Condiciones necesarias para su aparición. Prevención. Algoritmo del Banquero. Commit y roll-back.
8. Invariantes: cómo encontrarlos y preservarlos. Lógicas temporales y modales. Propiedades de vitalidad y seguridad. Pruebas formales. Verificación de invariantes y exclusión de configuraciones indeseadas.

## TRABAJOS PRACTICOS

### a) Enumeración:

1. Elementos de Pascal FC. Uso de send7recv.
2. POSIX Threads. Uso de mutexes y variables de condición.
3. Implementación de semáforos de Dijkstra y aplicaciones.
4. Rendez Vous.
5. Mensajes asíncronos y patrones. Erlang y OTP.

### b) Guías de trabajos prácticos publicadas: (con su código de publicación)

## BIBLIOGRAFIA

**a) Adecuada al programa. Ordenada por temas y con su codificación de biblioteca, incluidas las publicaciones de la Cátedra con su código de publicación.**

Amstrong, Virding, Wikstrom, Claes and Williams,  
Concurrent Programming in Erlang,  
Prentice-Hall, 2.996 (2da ed).  
Burns, A. and Davies, G.,  
Concurrent Programming,  
Addison-Wesley, 1.993.  
Schneider, F. B.,  
On Concurrent Programming,  
Springer Verlag, 1.997.  
Roscoe, A. W.,  
The Theory and Practice of Concurrency",  
Prentice-Hall (Pearson), 2.005.

**b) Complementaria para profundización o extensión de temas.**

Lea, Doug,  
Programación Concurrente en Java,  
Addison-Wesley, 2.001.  
Burns and Wellings,  
Sistemas de Tiempo Real y Lenguajes de Programación,  
Addison-Wesley, 2.003.

## **REGIMEN DE PROMOCIONALIDAD**

Por parciales y entrega de practicas.