

 UNIVERSIDAD NACIONAL DE CÓRDOBA Facultad de Ciencias Exactas, Físicas y Naturales República Argentina	Programa de: <h2 style="text-align: center;">Programación Concurrente</h2> Código: 2628																
Carrera: <i>Ingeniería en Computación</i> Escuela: <i>Ingeniería Electrónica y Computación.</i> Departamento: <i>Computación.</i>	<table border="0" style="width: 100%;"> <tr> <td>Plan:</td> <td style="text-align: right;">285-05</td> <td>Puntos:</td> <td style="text-align: right;">4</td> </tr> <tr> <td>Carga Horaria:</td> <td style="text-align: right;">96</td> <td>Hs. Semanales:</td> <td style="text-align: right;">6</td> </tr> <tr> <td>Semestre:</td> <td style="text-align: right;"><i>Séptimo</i></td> <td>Año:</td> <td style="text-align: right;"><i>Cuarto</i></td> </tr> <tr> <td>Carácter:</td> <td colspan="3" style="text-align: right;"><i>Obligatoria</i></td> </tr> </table>	Plan:	285-05	Puntos:	4	Carga Horaria:	96	Hs. Semanales:	6	Semestre:	<i>Séptimo</i>	Año:	<i>Cuarto</i>	Carácter:	<i>Obligatoria</i>		
Plan:	285-05	Puntos:	4														
Carga Horaria:	96	Hs. Semanales:	6														
Semestre:	<i>Séptimo</i>	Año:	<i>Cuarto</i>														
Carácter:	<i>Obligatoria</i>																
Objetivos: <i>Que el alumno</i> <ul style="list-style-type: none"> • <i>Saber identificar la concurrencia necesaria en una aplicación (análisis de la concurrencia). Para ello es necesario conocer la interacción de la aplicación con su entorno y se pondrá especial atención en la detección de y prevención de situaciones problemáticas— interbloqueos, falta de concurrencia, etc.</i> • <i>Proporcionar un lenguaje formal de diseño de aplicaciones concurrentes, basado en la definición de procesos, e interacciones. Las interacciones se definen a partir de pre- y pos condiciones de uso y son la base de la codificación del programa concurrente.</i> • <i>Proporcionar una sistemática para construir un programa concurrente correcto a partir de un diseño.</i> 																	
Programa Sintético: <ol style="list-style-type: none"> 1. <i>Introducción a los sistemas informáticos.</i> 2. <i>Introducción a los sistemas operativos.</i> 3. <i>Descripción y control de procesos.</i> 4. <i>Hilos, SMP y micro núcleos.</i> 5. <i>Concurrencia: exclusión mutua y sincronización.</i> 6. <i>Concurrencia: interbloqueo e inanición.</i> 7. <i>Proceso distribuido, cliente/servidor y agrupaciones.</i> 8. <i>Autómatas finitos y redes de Petri.</i> 																	
Programa Analítico: de foja 2 a foja 2.																	
Programa Combinado de Examen (si corresponde): de foja a foja .																	
Bibliografía: de foja 3 a foja 3.																	
Correlativas Obligatorias: <i>Modelos y Simulación</i>																	
Correlativas Aconsejadas:																	
Rige: 2005																	
<table border="0" style="width: 100%;"> <tr> <td style="width: 50%;">Aprobado HCD, Res.:</td> <td style="width: 50%;">Modificado / Anulado /Sust. HCD Res.:</td> </tr> <tr> <td>Fecha:</td> <td>Fecha:</td> </tr> </table> <p>El Secretario Académico de la Facultad de Ciencias Exactas, Físicas y Naturales (UNC) certifica que el programa está aprobado por el (los) número(s) y fecha(s) que anteceden. Córdoba, / / .</p>		Aprobado HCD, Res.:	Modificado / Anulado /Sust. HCD Res.:	Fecha:	Fecha:												
Aprobado HCD, Res.:	Modificado / Anulado /Sust. HCD Res.:																
Fecha:	Fecha:																
Carece de validez sin la certificación de la Decretaría Académica:																	

PROGRAMA ANALITICO

LINEAMIENTOS GENERALES

La presente asignatura es una actividad curricular que pertenece al octavo semestre de la carrera de Ingeniería en Computación.

El uso de la programación experimentó una gran expansión, debido a la confluencia de diversos factores entre los que debemos destacar.

- Lenguajes que implementan distintos paradigmas
- Arquitecturas computacionales que facilitan y promueven la programación en paralelo
- Herramientas de depuración que faciliten las pruebas.

Con estos factores como foco, en esta materia se desarrollan y se aplican conceptos dentro de las dos líneas de trabajo:

- **Diseño de programas:** Ser capaz, a partir de los requerimientos, de diseñar e implementar un sistema con la problemática de la concurrencia
 - Aprender como se representa un diseño de software
 - Aprender las actividades mas importante del proceso de diseño e implementación
 - Hacer uso de los distintos modelos y su representación
 - Identificar, evaluar y resolver los problemas de concurrencia
- **Pruebas del Software:** Ser capaz de diseñar y realizar pruebas en sistemas que tengan concurrencia
 - Aprender la diferencia entre Pruebas de validación y pruebas de defectos
 - Aprender los principios de pruebas de sistema y pruebas de componentes
 - Distintas estrategias para generar casos de pruebas
 - Comprender las características esenciales de herramientas de software que soportan la automatización de la prueba, de rendimiento y eficiencia en sistemas con concurrencia.
 - Hacer uso de herramientas que detectan y sugieren corrección a problemas de concurrencia

El dictado se orienta a capacitar al alumno para **Identificar y construir modelos de sistemas donde se hace uso y explotan las ventajas de la concurrencia;** para lograr sistemas con **eficiencia y eficacia donde los requerimientos son volátiles y flexibles.** Todo expresado como un diseño, implementación y las pruebas de sistema.

METODOLOGIA DE ENSEÑANZA

Las clases impartidas son teóricas, prácticas y de laboratorio.

Las actividades teóricas se realizan a través de exposiciones del docente orientadas a desarrollar en los alumnos la capacidad de Saber identificar la concurrencia, la interacción de la aplicación con su entorno, el diseño y las pruebas de sistemas.

Durante el desarrollo de los Trabajos Prácticos se realizan actividades que le permiten al estudiante poner en práctica las habilidades y verificar los criterios y técnicas de modelado, diseño y prueba.

Por otra parte en las clases de Laboratorio el alumno verifica, a través de la implementación, el funcionamiento de sistemas y realizara los casos de pruebas

EVALUACION

Condiciones para la promoción de la materia

1. Tener aprobadas las materias correlativas.-
2. Asistir al 80% de las clases teóricas y prácticas.-
3. Rendir y aprobar los dos parciales con 50% , el alumno podrá recuperar una vez cada parcial
4. Presentar y aprobar las implementaciones que se exijan durante el desarrollo de los trabajos prácticos.-
5. Aprobar los trabajos de Laboratorio.-
6. Aprobar un coloquio integrador

Los alumnos que cumplan las exigencias referidas en los puntos 1 al 6 serán considerados promocionados.

Los alumnos que cumplan las exigencias referidas en los puntos 1 al 4 serán considerados regulares.
El resto será considerado libre.

CONTENIDOS TEMATICOS

Unidad 1. Introducción a los sistemas informáticos

1. Elementos básicos
2. Registros del procesador
3. Ejecución de instrucciones
4. Interrupciones
5. Jerarquía de la memoria
6. Memoria cache
7. Técnicas de comunicación de E/S

Unidad 2. Introducción a los sistemas operativos

1. Funciones y objetivos de los sistemas operativos
2. Evolución de los sistemas operativos
3. Logros principales
4. Características de los sistemas operativos modernos
5. Introducción a Windows 2000, Sistemas UNIX clásicos, Sistemas UNIX modernos
6. Definición de los conceptos de concurrencia y paralelismo frente a la secuencialidad
9. Problemática de la programación concurrente

Unidad 3. Descripción y control de procesos

1. Estados de un proceso
2. Descripción de procesos
3. Control de procesos
4. Gestión de procesos en UNIX SVR4
5. Resumen

Unidad 4. Hilos, SMP y micronúcleos

1. Procesos e hilos
2. Multiproceso simétrico
3. Micronúcleos
4. Hilos y SMP en Windows 2000
5. Hilos y SMP en Solaris
6. Hilos y procesos en LINUX

Unidad 5. Concurrencia: exclusión mutua y sincronización

1. Principios generales de la concurrencia
2. Exclusión mutua: soluciones por software
3. Exclusión mutua: soluciones por hardware
4. Semáforos
5. Monitores
6. Paso de mensajes
7. Problema de los lectores/escritores

Unidad 6. Concurrencia: interbloqueo e inanición

1. Principios del interbloqueo
2. Prevención del interbloqueo
3. Predicción del interbloqueo
4. Detección del interbloqueo
5. Una estrategia integrada de interbloqueo
6. El problema de la cena de los filósofos
7. Mecanismos de concurrencia
8. Primitivas de sincronización de hilos
9. Mecanismos de concurrencia en Windows 2000
10. Resumen

Unidad 7. Proceso distribuido, cliente/servidor y agrupaciones

1. Proceso cliente/servidor
2. Paso distribuido de mensajes
3. Llamadas a procedimiento remoto
4. Agrupaciones
5. RMI

Capítulo 8. Autómatas finitos

1. Maquinas de Turing y equivalencia entre maquinas de Turing
2. Autómatas finitos deterministas y no deterministas
3. Autómatas a pila deterministas y no deterministas

Capítulo 9. Redes de Petri

1. Redes de Petri ordinarias
2. Matriz de Incidencia
3. Propiedades de las redes de Petri (Vivacidad, Interbloqueo, etc.)
4. Modelo de proceso Concurrente
5. Invariantes de Transición y Plaza
6. Redes de Petri Coloreadas

LISTADO DE ACTIVIDADES PRACTICAS Y/O DE LABORATORIO

Actividades Prácticas

1.- Herramientas para implementar Hilos

Conceptos Básicos y avanzados de Java
Conceptos Básicos y avanzados de Hilos

2.- La jerarquía de Hilos en java

Clases Relacionadas con los Hilos
Creación de Hilos

3.- Control y de Hilos

Estado y Control de un Hilos
Planificación y Prioridad de Hilos
Sincronización

4.- Mecanismos de comunicación

Java.net
RMI

5. Redes de Petri

Simulación
Modelado de un Problema

Actividades de Laboratorio

Implementación de modelos a partir de requerimientos, detección de recurrencia
Diseño e Implementación de sistemas con Hilos
Implementación y pruebas de sistemas con Hilos
Diseño e implementación de sistemas con comunicación

DISTRIBUCION DE LA CARGA HORARIA

ACTIVIDAD	HORAS
TEÓRICA	43
FORMACIÓN PRACTICA:	
○ FORMACIÓN EXPERIMENTAL	9
○ RESOLUCIÓN DE PROBLEMAS	22
○ ACTIVIDADES DE PROYECTO Y DISEÑO	22
○ PPS	
TOTAL DE LA CARGA HORARIA	96

DEDICADAS POR EL ALUMNO FUERA DE CLASE

ACTIVIDAD	HORAS
PREPARACION TEÓRICA	45
PREPARACION PRACTICA	
○ EXPERIMENTAL DE LABORATORIO	10
○ EXPERIMENTAL DE CAMPO	
○ RESOLUCIÓN DE PROBLEMAS	35
○ PROYECTO Y DISEÑO	35
TOTAL DE LA CARGA HORARIA	125

BIBLIOGRAFIA

Principal

- Doug Lea, Programacion Concurrente en Java, 2ª Edición Addison Wesley
- Jose T Palma Méndez, Programación Concurrente, Tomson 2003
- William Stallings, Sistemas Operativos, 4ª Edición Pearson Hall
- Orlando Micolini, Apuntes de Catedra

Complementaria

- Jean Bacon, Concurrent Systems, 1993 Addison Wesley
- Silberschatz Galvin, Sistemas Operativos, 5ª Edición Addison Wesley
- BEN-AIR M. *Principles of concurrent and distributed programming*. Mc GRAW hill. 1990.
- Bruce Eckel, Thinking in Java President, MindView Inc. Prentice Hall PTR
- Enrique Alfonseca Cubero, Manuel Alfonseca Moreno, Teoria de Automatas y Lenguajes Formales, McGrawHill, 2007