

 <p>UNIVERSIDAD NACIONAL DE CÓRDOBA Facultad de Ciencias Exactas, Físicas y Naturales República Argentina</p>	<p>Programa de:</p> <p style="text-align: center;">Programación Concurrente</p> <p>Código: 7415</p>
<p>Carrera: <i>Ingeniería en Computación</i> Escuela: <i>Ingeniería en Computación</i> Departamento: <i>Computación</i></p>	<p>Plan: 285-05 Puntos: 4 Carga Horaria: 96 Hs. Semanales: 6 Semestre: <i>Séptimo</i> Año: <i>Cuarto</i> Carácter: <i>Obligatoria</i> Bloque: <i>Tec. Aplicadas</i></p>
<p>Objetivos:</p> <p>Que el alumno:</p> <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> 	
<p>Programa Sintético:</p> <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> 	
<p>Programa Analítico: de foja 2 a foja 5.</p>	
<p>Programa Combinado de Examen (si corresponde): de foja a foja .</p>	
<p>Bibliografía: de foja 5 a foja 5.</p>	
<p>Correlativas Obligatorias: <i>Modelos Y Simulación</i></p>	
<p>Correlativas Aconsejadas:</p>	
<p>Rige: 2005</p>	
<p>Aprobado HCD, Res.: Fecha:</p>	<p>Modificado / Anulado / Sust. HCD Res.: Fecha:</p>
<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>	
<p>Carece de validez sin la certificación de la Secretaría Académica:</p>	

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 asignatura 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 cómo se representa un diseño de software
 - Aprender las actividades más 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.

METODOLOGÍA DE ENSEÑANZA

Las clases impartidas son teóricas, prácticas y de laboratorio.

Las actividades teóricas se realizan a través de exposiciones dialogadas 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 realizará los casos de pruebas

EVALUACIÓN

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 aprobados los dos parciales con 50% , el alumno podrá recuperar una vez cada parcial
4. Presentar y aprobar los trabajos 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 5 serán considerados regulares.

El resto será considerado libre

CONTENIDOS TEMÁTICOS

Unidad 1. Introducción a los sistemas informáticos

1. Elementos básicos
2. Registros del procesador, Ejecución de instrucciones, Interrupciones
3. Jerarquía de la memoria, Memoria cache
4. 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. Características de los sistemas operativos modernos
4. Definición de los conceptos de concurrencia y paralelismo frente a la secuencialidad
5. 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 procesos en LINUX

Capítulo 5. Autómatas finitos y lenguajes regulares

1. Máquinas de Turing y equivalencia entre máquinas de Turing y lenguajes
2. Autómatas finitos deterministas y no deterministas
3. Autómatas a pila deterministas y no deterministas

Capítulo 6. 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 Coloreada

Unidad 7. 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 8. Concurrencia: interbloqueo e inanición

1. Hacer uso de Redes de Petri para: Identificar, detectar, predecir y prevenir el interbloqueo

2. Mecanismos de concurrencia

3. Primitivas de sincronización de hilos

2. El problema de la cena de los filósofos

3 El problema de la barbería y otros

10. Resumen

Capítulo 6. Redes de Petri Temporales

1. Redes de Petri temporales formalismos

2. Semántica de las Redes de Petri Temporales

3. Propiedades de las redes de Petri Temporales

4. Grado de sensibilización y la prioridad

5. Interpretaciones con tiempo

6. Semántica de tiempo débil y fuerte

7. Estado de una Rede de Petri Temporal

8. Modelado de sistemas de tiempo real mediante RdP con Tiempo

Unidad 7. Proceso distribuido, cliente/servidor y agrupaciones

1. Proceso cliente/servidor

2. Paso distribuido de mensajes

LISTADO DE ACTIVIDADES PRÁCTICAS 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

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

DISTRIBUCIÓN DE LA CARGA HORARIA

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

BIBLIOGRAFÍA**BIBLIOGRAFIA Principal**

- Méndez, J.T.P., Programación concurrente2003: Editorial Paraninfo.
- Diaz, M., Petri nets: fundamental models, verification and applications2013: John Wiley & Sons.
- Cueva, J., et al., Lenguajes, Gramáticas y Autómatas en Procesadores de Lenguaje, 2003, Servitec.
- González, J.F., Java 7 Concurrency Cookbook2012: BIRMINGHAM - MUMBAI.

BIBLIOGRAFÍA Secundaria

- Reisig, W., Understanding Petri Nets, Modeling Techniques, Analysis Methods, Case Studies2013, Berlin, Germany: Springer-Verlag Berlin Heidelberg 2013.
- Tsvetinov, N., Learning reactive programming with Java 8. 2015.