



MAestrÍA EN CIENCIAS DE LA COMPUTACIÓN PROPUESTA DE CURSO DE POSGRADO

1- DATOS GENERALES DE LA ACTIVIDAD CURRICULAR					
1.1 Título del Curso	Tópicos Avanzados en Programación Concurrente				
1.2 Área temática ¹	Algoritmos y Lenguajes				

2- COMPOSICION DEL EQUIPO DOCENTE					
2.1 Responsable a cargo de la actividad curricular	Dra. Nadina Martínez Carod				
2.2 Docentes					

3- CARGA HORARIA					
Carga horaria teórica	30				
Carga horaria práctica	30				
Carga horaria total	60 hs				
Distribución horaria semanal	Lu	Ma	Mie	Jue	Vie
Fecha de inicio sugerida					

4- BREVE RESUMEN DE CONTENIDOS (hasta 400 palabras)					
<p>Se analizan los conceptos y técnicas de programación concurrente utilizadas para resolver los problemas de comunicación y sincronización de procesos. El curso aborda diferencias y similitudes respecto a las herramientas para la sincronización proporcionadas en diferentes lenguajes de programación, para luego enfocarse en el lenguaje Java. El curso abarca clases, interfaces, objetos y colecciones avanzadas incorporadas al paquete de concurrencia de Java. También se muestran algunos tópicos para testing. Se analizan formas de utilización de concurrencia en visualizaciones y animaciones. Luego, desde un enfoque pragmático, se realiza implementación y testing de una aplicación en un escenario concurrente.</p>					

5- CONOCIMIENTOS PREVIOS REQUERIDOS					
Conocimientos básicos de Programación Concurrente					

6- OBJETIVOS					
<ul style="list-style-type: none">• Conocer, analizar e implementar las diferentes alternativas utilizadas en la sincronización de procesos.• Estudiar tópicos de concurrencia en distintos lenguajes de programación• Documentar políticas de sincronización para el mantenimiento de procesos.• Analizar seguridad de procesos concurrentes .• Realizar testing de algoritmos concurrentes.• Implementar soluciones a problemas en escenarios concurrentes utilizando los tópicos avanzados de programación vistos.					

7- CONTENIDOS (organizados en unidades, ejes, módulos, otros)					

¹ Corresponde a uno de los siguientes tópicos: Algoritmos y Lenguajes; Teoría de la Computación; Ingeniería de Software, Bases de Datos y Sistemas de Información; Arquitecturas, Sistemas Operativos y Redes.



Conceptos de concurrencia: Nociones de multiprocesamiento, Multiprogramación, Computación Distribuida. Sistemas concurrentes. Problemas de la programación concurrente. Sincronización y exclusión mutua. Propiedades. Productor - Consumidor. Procesos e Hilos. Estructura y ciclo de vida de los procesos. Límites de la concurrencia. Startvation, deadlock, livelock. Concurrencia basada en actores. Escalabilidad

Modelos: Modelos de Concurrencia, Paradigmas de resolución de programas concurrentes, Programación Concurrente Orientado a Objetos, Objetos activos, pasivos y mixtos. Objetos inmutables. Algoritmos concurrentes, paralelos y distribuidos.

Mecanismos de sincronización: Algoritmos de Exclusión mutua. Técnicas de sincronización: Barreras, monitores, semáforos, métodos sincronizados, locks. Técnicas de progreso. Procesos con prioridades diferentes e iguales. Región crítica. Mecanismo de seguridad. Construcción de clases thread-safe. Sincronizadores para coordinar procesos cooperativos.

Tópicos Avanzados del Lenguaje: Librerías avanzadas de concurrencia en Java, Variable atómicas, números aleatorios para múltiples hilos, colecciones concurrentes, colecciones parametrizadas. Problemas con colecciones sincronizadas. Executors (objetos, implementaciones, interfaces), clases para coordinar grupos de hilos. Herramientas para documentar políticas de sincronización.

Aplicaciones: Descripción de problemas característicos y su resolución: Lectores/Escritores, Productor/Consumidor, Filósofos cenando, Barbero dormilón. Análisis de casos de estudio con diferentes mecanismos de sincronización. Protocolo para casos de Prueba.

8- PROPUESTA DIDÁCTICA (metodología de trabajo de clases teóricas y prácticas)

La modalidad del curso es presencial. La metodología que se utilizará es la explicación de conceptos, discusión de ideas o propuestas, y realización de actividades prácticas pequeñas. Las actividades prácticas están programadas en ejercicios simples para incorporar los conceptos que se van adquiriendo, siguiendo un mismo hilo conductor. Antes del fin del curso se propone una actividad integral que se analizará dentro del curso pero tendrá un plazo final entrega dentro de los 60 días de finalización del curso.

Los alumnos contarán con un cronograma sugerido de temas y actividades para realizar. Las actividades serán realizadas en grupos de hasta 3 personas fomentando el desarrollo colaborativo en los integrantes del curso. Las actividades varían en temas de investigación, de desarrollo y de testing.

Al inicio del curso se les facilitará a los estudiantes el programa del curso y el calendario sugerido para la entrega de las distintas actividades a realizar. Para el trabajo con cada eje temático se les proveerá información, se les sugerirá bibliografía y direcciones de internet para consultar. Cada actividad se deberá desarrollar en grupo y deberá ser entregada dentro en un corto plazo estipulado, salvo el trabajo final integrador el cual será realizado en forma individual y será enviado para su posterior evaluación. Para aquellos alumnos que así lo deseen el trabajo final integrador podrá ser reemplazado por un paper con contenido desarrollado a partir de los temas abarcados e investigados.

Además de las instancias presenciales, se utilizará un espacio en la plataforma PEDCO de la Universidad para la comunicación, material ofrecido y entrega de las actividades realizadas.

9- MODALIDAD DE EVALUACIÓN Y CONDICIONES DE ACREDITACIÓN²

² Son condiciones mínimas para la aprobación de todos los cursos: cumplir con un mínimo del 80% de asistencia a las clases, realizar las tareas y aprobar las evaluaciones que se hayan propuesto en el programa, con una calificación no menor a 7 (puntos). Los trabajos de



ACREDITACIÓN: 80% de asistencia a las clases, realización de todas las actividades prácticas propuestas.

EVALUACIÓN: Se evaluará las actividades prácticas propuestas junto con el análisis realizado sobre un tema particular. Además se deberá aprobar el integrador final o elaborar un paper con contenidos vistos en el curso.

10– BIBLIOGRAFÍA DE LECTURA OBLIGATORIA CORRESPONDIENTE A CADA UNIDAD Y GENERAL

- Doug Lea. Concurrent Programming in Java: Design Principles and Patterns, Addison Wesley 1997
- Gomaa Hassan. Designing concurrent, distributed and real-time applications with UML. Addison Wesley 2002.
- Scott Oaks & Henry Wong. Java Threads. O'Reilly - 1999
- Brian Goetz et al., Java Concurrency in Practice, Pearson Education, Inc. 2006.
- Bruce Eckel. Thinking in JAVA. President, MindView Inc - 2008
- Andrew P. Black et al. Squeak by Example, 2009. (Disponible en www.squeak.org.)
- Paul Butcher. Seven Concurrency Models in Seven Weeks, When Threads Unravel. The Pragmatic Programmer 2014
- Venkat Subramanian. Programming Concurrency on the JVM: Mastering Synchronization, STM, and Actors. Pragmatic Programmers 2011

Direcciones útiles:

- <https://docs.oracle.com/javase/tutorial/essential/concurrency/>
- <http://tutorials.jenkov.com/java-concurrency/index.html>

11– INFRAESTRUCTURA E INSUMOS REQUERIDOS³

Sala de informática.

12 – OTRA INFORMACIÓN RELEVANTE

evaluación pautados y la calificación de los alumnos deberán realizarse dentro de los 60 días posteriores a la finalización del curso.

³ Deberá constar aquí si la realización del curso requiere contar con instalaciones especiales (laboratorio, sala de informática, equipamiento audiovisual, etc). Explicitar si se estima que el curso debe tener un número máximo determinado de asistentes para poder ser dictado.