



000027

Lenguajes Informáticos 3

INSTITUTOS/S: **INSTITUTO DE TECNOLOGÍA E INGENIERÍA**

CARRERA/S: **Licenciatura en Informática**

RESPONSABLE DE LA ASIGNATURA Y EQUIPO DOCENTE:

Pedro Giuffrida
Carlos Jimenez

AÑO: 2025

CRÉDITOS: 5

CARGA HORARIA DE INTERACCIÓN PEDAGÓGICA: 64

CARGA HORARIA TOTAL: 125

CÓDIGO DE LA MATERIA EN SIU:

000027

1. Fundamentación

En el panorama actual de la informática y el desarrollo de software, la programación concurrente, paralela y basada en eventos ha dejado de ser un nicho especializado para convertirse en un requisito fundamental y transversal en casi todas las áreas de aplicación. Con la proliferación de procesadores con múltiples núcleos (multicore) y a arquitecturas distribuidas (clusters, computación en la nube) y para explotar el verdadero potencial de estos recursos, es imperativo que el software sea capaz de ejecutar múltiples tareas de forma simultánea o casi simultánea.

Los usuarios modernos esperan aplicaciones que respondan instantáneamente, sin bloqueos, y que puedan manejar múltiples operaciones en segundo plano (por ejemplo, una interfaz de usuario que no se congela mientras se descarga un archivo grande o se procesan datos). Esto solo se logra a través de técnicas de concurrencia y programación asíncrona/basada en eventos. Todo esto y más (Servidores web de alto rendimiento, APIs RESTful, WebSockets, JavaScript asíncrono; En Sistemas Operativos: Gestión de procesos, hilos, comunicación interproceso; en Bases de Datos: Transacciones concurrentes, control de concurrencia; Sistemas Embebidos/IoT: Procesamiento en tiempo real, manejo de eventos.) hace esta materia indispensable para formar profesionales y desarrolladores de software que puedan construir sistemas modernos, escalables, eficientes, reactivos y robustos en un entorno dominado por arquitecturas multicore y distribuidas.

2. Propósitos y/u objetivos

- *Diferenciar claramente entre concurrencia, paralelismo y programación basada en eventos, explicando sus motivaciones y aplicaciones.*
- *Comprender los modelos de cómputo asociados a cada uno de estos paradigmas.*
- *Identificar y Resolver Problemas de Concurrencia, evaluando y optimizando el rendimiento y la seguridad de aplicaciones concurrentes.*
- *Programar en Entornos Basados en Eventos.*
- *Comprender la Computación de Alto Rendimiento.*
- *Aplicar técnicas de concurrencia y paralelismo en lenguajes de programación contemporáneos (ej. Java, C#, Python, Go, Rust).*

000027



- *Comprender los fundamentos de la programación en GPU y los criterios de eficiencia energética en sistemas concurrentes y paralelos.*

3. Contenidos mínimos

Estudio de los fundamentos de la programación concurrente, el paralelismo y la programación basada en eventos. Modelos de concurrencia y sus motivaciones; diferencias conceptuales y operativas entre concurrencia, paralelismo y eventos. Modelos de memoria compartida y pasaje de mensajes, incluyendo mecanismos de sincronización, exclusión mutua y comunicación entre procesos e hilos. Análisis de problemas asociados a la concurrencia: condiciones de carrera, inanición, interbloqueos, seguridad y progreso. Exploración de modelos de interacción como cliente-servidor y productor-consumidor. Programación orientada a eventos en entornos concurrentes. Aplicación de técnicas de concurrencia, paralelismo y programación basada en eventos en lenguajes de programación contemporáneos. Introducción a la programación sobre unidades de procesamiento gráfico (GPU) y estudio de criterios de eficiencia energética en entornos de ejecución concurrente y paralelo.

4. Carga horaria

<i>Créditos</i>	<i>Interacción pedagógica</i>		<i>Trabajo Autónomo</i>	<i>Total</i>
<i>5</i>	<i>Teoría: 32</i>	<i>Práctica: 32</i>	<i>61</i>	<i>125</i>

4.1 Trabajo autónomo del/la estudiante

<i>Actividad</i>	<i>Carga horaria</i>
<i>Lectura de contenidos teóricos apuntes o bibliografía.</i>	<i>20</i>
<i>Preparación de evaluaciones.</i>	<i>21</i>
<i>Desarrollo de los ejercicios prácticos propuestos.</i>	<i>20</i>

5. Programa analítico

UNIDAD 1: *Fundamentos de la concurrencia, paralelismo y programación basada en eventos.*

Motivaciones de la concurrencia y el paralelismo en sistemas modernos. Diferencias entre concurrencia, paralelismo y programación basada en eventos. Modelos de ejecución:

000027



procesos, hilos, eventos, multitarea y paralelismo masivo. Introducción a modelos de memoria compartida, pasaje de mensajes y flujos de eventos.

UNIDAD 2: *Sincronización y problemas de concurrencia.*

Sincronización en memoria compartida: secciones críticas, locks, semáforos, monitores, condition variables, barriers y rendezvous. Comunicación en modelos de pasaje de mensajes: comunicación sincrónica y asincrónica. Modelos de seguridad y progreso: race conditions, deadlocks, starvation, fairness.

UNIDAD 3: *Modelos de interacción y programación orientada a eventos.*

Modelos de interacción: cliente-servidor, productor-consumidor, publicador-suscriptor. Programación orientada a eventos: principios, flujo de control dirigido por eventos, asincronía y callbacks. Herramientas y abstracciones de concurrencia y eventos en distintos lenguajes de programación.

UNIDAD 4: *Paralelismo en GPU y eficiencia energética.*

Introducción a la arquitectura de unidades de procesamiento gráfico (GPU). Paralelismo masivo y programación heterogénea. Impacto del diseño paralelo en el consumo energético y la eficiencia de ejecución.

6. Ejes y enunciados Multidimensionales y transversales

Eje	Nivel de logro de Aprendizaje	Acciones de enseñanza
C1. Identificación, formulación y resolución de problemas de informática	Alta	Debates, casos de estudio, resolución de ejercicios y problemas.
C2. Concepción, diseño y desarrollo de proyectos de informática	Alto	Debates, casos de estudio, resolución de ejercicios y problemas.
C3. Gestión, planificación, ejecución y control de proyectos de informática	Alto	Debates, casos de estudio, resolución de ejercicios y problemas.
C4. Utilización de técnicas y herramientas de aplicación en la informática	Alto	Debates, casos de estudio, resolución de ejercicios y problemas.

000027

C5. Generación de desarrollos tecnológicos y/o innovaciones tecnológicas	N/A	
C6. Fundamentos para el desempeño en equipos de trabajo	Medio	Aprendizaje Cooperativo.
C7. Fundamentos para la comunicación efectiva	Medio	Exposición.
C8. Fundamentos para la acción ética y responsable	N/A	
C9. Fundamentos para evaluar y actuar en relación con el impacto social de su actividad en el contexto global y local	N/A	
C10. Fundamentos para el aprendizaje continuo	Alto	Debates, casos de estudio, resolución de ejercicios y problemas.
C11. Fundamentos para la acción emprendedora	N/A	

7. Bibliografía y recursos

7.1 Bibliografía obligatoria

Manuel Capel Tuñón (2022) ""Programación concurrente y en tiempo real: Fundamentos y aplicaciones" Editorial Ibergarceta Publicaciones.

David Vallejo Fernández, Carlos González Morcillo y Javier Alonso Albusac Jiménez.(2014) "Programación Concurrente y Tiempo Real" 2da ed. (Edlibrix)
Disponible en:

<https://bibliotecavirtualapure.wordpress.com/wp-content/uploads/2015/06/programacion-concurrente-y-tiempo-real.pdf>

Tanenbaum A., Van Steen (2008) "Sistemas distribuidos: principios y paradigmas" 2da Ed. Editorial: Pearson Educación.

Disponible en:

https://moodle.uneq.edu.ve/pluginfile.php/114008/mod_page/content/6/Sistema-Distribuidos-esp.pdf

000027

7.2 Bibliografía optativa

David B. Kirk and Wen-mei W. Hwu (2022) . "Programming Massively Parallel Processors: A Hands-on Approach" 4ta Ed. Morgan Kaufmann.

Juan Ujaldón (2015) "Programación de GPUs con CUDA" Universidad de Málaga.

7.3. Recursos

Se cuenta con la plataforma en línea que ofrece la Universidad: Moodle, para la gestión de materiales, foros de discusión y entregas de proyectos.

Documentación Oficial de Lenguajes: APIs de concurrencia de Java, C#, Go, Python, Rust. o cualquier lenguaje que se utilice.

Repositorios de Código: GitHub/GitLab para el control de versiones y el trabajo colaborativo en proyectos.

Herramientas de Simulación y Depuración: Visualizadores de concurrencia, debuggers especializados, herramientas de análisis de rendimiento.

8. Metodología de enseñanza

8.1 Modalidades u opciones pedagógicas

Para esta materia se propone una metodología que combine teoría sólida con una intensa práctica, enfocándose en la comprensión de los desafíos y la aplicación de soluciones en entornos contemporáneos. La modalidad es presencial con fuertes componentes de laboratorio, simulación y proyectos. Los conceptos abstractos de concurrencia se comprenden mejor al enfrentar y resolver problemas concretos que exhiben sus complejidades. Los proyectos impulsarán la aplicación integral de los conocimientos. Se proponen clases teóricas, demostraciones en vivo, laboratorios de implementación.

8.2 Formación práctica

Como toda materia de programación la formación práctica es fundamental para comprender los temas de la misma, se proponen entre otros las siguientes tareas:

000027

Estudiar implementaciones reales de problemas de concurrencia para entender las soluciones y sus potenciales fallas.

Implementación de problemas clásicos de concurrencia (ej. el problema del productor - consumidor, el de los filósofos comensales) utilizando distintos mecanismos de sincronización en lenguajes.

Utilización de herramientas de depuración avanzadas y visualizadores para identificar y resolver condiciones de carrera e interbloqueos. Simulación de escenarios de falla.

Mostrar y analizar ejemplos sencillos para entender los conceptos de kernels, hilos, bloques y grids.

9. Evaluación y régimen de aprobación

9.1 Modalidad de evaluación

La evaluación se realizará a través de dos exámenes parciales con posibilidad de recuperar cada uno de ellos una vez. A su vez evaluaremos a través de las actividades obligatorias y optativas y la observación directa en clase.

9.2 Aprobación de la cursada

Para aprobar la cursada y obtener la condición de regular, el régimen académico establece que debe obtenerse una nota no inferior a cuatro (4) puntos. Todas las instancias evaluativas deberán tener una instancia de recuperatorio. Podrán acceder a la administración de esta modalidad solo aquellos y aquellas estudiantes que hayan obtenido una nota inferior o igual a 6 (seis) puntos en el examen parcial.

Siempre que se realice una evaluación de carácter recuperatorio, la calificación que los/as estudiantes obtengan reemplazará la calificación obtenida en el examen que se ha recuperado y será la considerada definitiva a los efectos de la aprobación.

9.3 Acreditación de la materia

La materia puede aprobarse por promoción, evaluación integradora, examen final o libre.

Promoción directa: tal como lo establece el art°17 del Régimen Académico, para acceder a esta modalidad, el/la estudiante deberá aprobar la cursada de la materia con una nota no inferior a siete (7) puntos, no obteniendo en ninguna de las instancias de evaluación parcial menos de seis (6) puntos, sean evaluaciones parciales o recuperatorios. El

000027

promedio estricto resultante deberá ser una nota igual o superior a siete (7) sin mediar ningún redondeo.

Evaluación integradora: tal como lo establece el art°18 del Régimen Académico, podrán acceder a esta evaluación aquellos estudiantes que hayan aprobado la cursada con una nota de entre cuatro (4) y seis (6) puntos.

La evaluación integradora tendrá lugar por única vez en el primer llamado a exámenes finales posterior al término de la cursada. Deberá tener lugar en el mismo día y horario de la cursada y será administrado, preferentemente, por el/la docente a cargo de la comisión. Se aprobará tal instancia con una nota igual o superior a cuatro (4) puntos, significando la aprobación de la materia.

La nota obtenida se promediará con la nota de la cursada.

Examen final: Instancia destinada a quienes opten por no rendir la evaluación integradora o hayan regularizado la materia en cuatrimestres anteriores. Se evalúa la totalidad de los contenidos del programa de la materia y se aprueba con una calificación igual o superior a cuatro (4) puntos. Esta nota no se promedia con la cursada.