

PROGRAMA DE LA ASIGNATURA:
Introducción a la Concurrencia (IF038)**CÓDIGO:** IF038
AÑO DE UBICACIÓN EN EL PLAN DE ESTUDIOS:
3 año
FECHA ULTIMA REVISIÓN DE LA ASIGNATURA:
2021-11-04
CARRERA/S: Analista Universitario de Sistemas
050/2017, Licenciatura en Sistemas 049/2017,**CARÁCTER:** CUATRIMESTRAL (1ro)
TIPO: OBLIGATORIA
NIVEL: GRADO
MODALIDAD DEL DICTADO: PRESENCIAL
MODALIDAD PROMOCION DIRECTA: SI
CARGA HORARIA SEMANAL: 4 HS
CARGA HORARIA TOTAL: 60 HS**EQUIPO DOCENTE**

Nombre y Apellido	Cargo	e-mail
PENDENTI HORACIO A	Prof. Adjunto	hpendenti@untdf.edu.ar
Rojas Luis	Asistente	lrojasflores@untdf.edu.ar

1. FUNDAMENTACION

Ubicada en el tercer año de la carrera, momento en que el Alumno ya se encuentra afianzado con la programación secuencial, la materia permite introducir el problema y la oportunidad que implica la concurrencia en la computación actual. Esta Asignatura plantea las cuestiones básicas de la programación concurrente y las alternativas para su solución, como un primer paso en la elaboración de soluciones correctas y acordes al estado del arte. De este modo, Introducción a la Concurrencia establece un nexo armónico entre la programación secuencial y las soluciones más sofisticadas y específicas que se plantean posteriormente en Sistemas Distribuidos y Sistemas Paralelos.

Esta asignatura se dictará en modalidad presencial-en línea.

Las clases teóricas y prácticas se dictan mediante herramientas como Zoom, Skype y/o Meet; Utilizando las pizarras que brindan o, en casos que se requiera una participación colaborativa, un documento compartido de google docs. Además, las clases teóricas se graban en video y se suben a un canal de youtube dedicado a la asignatura (https://www.youtube.com/channel/UC6TcCyY3QAP8nEG7bEuHpkw?view_as=subscriber), junto a videos especiales sobre temas específicos centrales a la materia. Como siempre, todo el material de consulta de la asignatura, programa, prácticas completas, lenguajes de programación, ejemplos prácticos, ejemplos de exámenes y bibliografía se encuentran disponibles para los Alumnos en la nube desde el comienzo hasta el final de la cursada.

2. OBJETIVOS

a) OBJETIVOS GENERALES

El programa está diseñado en base a los contenidos mínimos con el fin de lograr que el Alumno que completa su cursado conozca la problemática que plantea la concurrencia y los fundamentos de la programación concurrente para resolverlos. De este modo se espera que esté en

condiciones de utilizar los distintos mecanismos para asegurar la sincronización de procesos en sus dos formas básicas: exclusión mutua y sincronización por condición. Se introducen además los conceptos fundamentales de comunicación y sincronización entre procesos, tanto en entornos de memoria compartida como en sistemas distribuidos para lograr un dominio de soluciones más amplio en arquitecturas modernas. Se espera entonces que el Alumno cuente con las herramientas necesarias para diseñar soluciones concurrentes y paralelas tanto en arquitecturas mono y multiprocesador como en sistemas distribuidos.

b) OBJETIVOS ESPECIFICOS

Comprensión de los problemas y oportunidades que representa la concurrencia.
Incorporar técnicas y herramientas que permitan diseñar soluciones concurrentes a problemas de la vida real.

3. CONDICIONES DE REGULARIDAD Y APROBACION DE LA ASIGNATURA

En concordancia con los términos expuestos en el Reglamento General de Estudios, se establecen los siguientes criterios para regularizar y aprobar la asignatura:

Para regularizar la asignatura:

El Alumno que haya entregado soluciones correctas a los ejercicios prácticos previstos en cada Trabajo Práctico y que apruebe con una nota de seis (6) o más los dos (2) parciales prácticos previstos, habrá regularizado la asignatura. Cada parcial práctico consta además de un recuperatorio para quienes no lo hayan aprobado en primera instancia.

La asignatura se puede aprobar de dos modos:

a) en situación regular:

Para aprobar la Asignatura, el Alumno que hubiere regularizado previamente el cursado, debe aprobar con una nota de cuatro (4) o más un examen final de contenido principalmente teórico, en forma escrita y oral. Es condición además que al momento de finalizar el cuatrimestre correspondiente, el Alumno tenga aprobadas las Asignaturas correlativas.

b) por promoción:

Los alumnos que obtengan una nota de ocho (8) o más en cada uno de los dos parciales prácticos en su primera instancia, tienen la opción de rendir dos parciales teóricos que deberán aprobar en primera instancia. En el caso de obtener una nota de siete (7) o más en cada uno de los parciales teóricos, habrá aprobado la materia por promoción, siendo la nota final el promedio de los dos parciales teóricos.

Los exámenes parciales tanto prácticos como teóricos, serán efectuados también en modalidad presencial-en línea y escritos en papel como es habitual. Finalizado el horario establecido para la entrega del mismo, el Alumno deberá escanearlo (fotografiarlo) y enviarlo por mail a la cuenta del docente: hpendenti@untdf.edu.ar.

4. CONTENIDOS DE LA ASIGNATURA

Contenidos mínimos:

- Conceptos de concurrencia.
- Especificación de la ejecución concurrente.
- Comunicación y sincronización.
- Concurrencia con variables compartidas.
- Concurrencia con pasaje de mensajes.
- Sistemas multiprocesador para concurrencia real.

- Lenguajes de programación concurrente.
- Diseño y programación de algoritmos concurrentes.

Respetando los contenidos mínimos enunciados en el plan de estudios y teniendo en cuenta la ubicación de la asignatura en el tramo, se han organizado los siguientes contenidos en unidades, cada una de ellas con una duración de dictado estimada en módulos de 4 hs. Reloj.

Programa simplificado:

Unidad 1: CBC – Conceptos básicos de Concurrencia.

Unidad 2: CYS – Concurrencia y Sincronización.

Unidad 3: CVC – Concurrencia con Variables Compartidas.

Unidad 4: SCS – Sincronización con Semáforos.

Unidad 5: SCM – Sincronización con Monitores.

Unidad 6: CPM – Concurrencia con Pasaje de Mensajes.

Unidad 7: LPC – Lenguajes de Programación Concurrente.

Programa detallado:

Unidad 1: CBC - Conceptos Básicos de Concurrencia: Evolución histórica. Nociones previas de Sistemas Operativos. Procesos: Conceptos de proceso. Estados de un proceso. Descripción de procesos, estructuras del S.O.. Control de procesos. Nociones y objetivos de los sistemas concurrentes. Procesamiento secuencial concurrente y paralelo. Características. Programa concurrente. No determinismo. Clases de aplicaciones. Multiprocesamiento, Multithreading, Cómputo paralelo y distribuido. Concurrencia y Paralelismo. Relación con la arquitectura. Monoprocesadores. Multiprocesadores. Relación con el sistema operativo. Requerimientos para el S.O. Relación con el lenguaje. Requerimientos para el S.O.

Unidad 2: CYS – Concurrencia y Sincronización: Comunicación y Sincronización. Sincronización por exclusión mutua y por condición. Comunicación por memoria compartida y por mensajes. Estados, acciones, historias y propiedades. Paralelización, requerimientos. Acciones atómicas y sincronización. Atomicidad de grano fino y grano grueso. La propiedad de “a lo sumo una vez”. La sentencia await. Semántica. Especificación de la sincronización. Especificación y semántica de la ejecución concurrente, la sentencia co y process. El problema de interferencia. Historias válidas e inválidas. Técnicas para evitar interferencia. Propiedades de seguridad y vida. Políticas de scheduling y fairness. Deadlock, Livelock y Starvation: Principios. Prevención. Detección. Evasión.

Unidad 3: CVC – Concurrencia con Variables Compartidas: Sincronización por variables compartidas. Sincronización de grano fino y grueso. Secciones críticas. Definición del problema. Propiedades necesarias de las soluciones. Soluciones de tipo spin-lock al problema de la SC. Aporte del hardware a las soluciones propuestas de la SC.. Deshabilitación de interrupciones. TestAndSet y TestAndTestAndSet. Algoritmos clásicos de soluciones fair al problema de la SC (tie-breaker, ticket, bakery). Implementación de sentencias await arbitrarias. Sincronización barrier. Definición y soluciones (contador compartido, flags y coordinadores, árboles, barreras simétricas, butterfly).

Unidad 4: SCS – Sincronización con Semáforos: Inconvenientes de la sincronización por variables compartidas. Semáforos, sintaxis y semántica. Usos básicos y técnicas de programación. Soluciones a SC y barreras. Semáforos binarios divididos (split). Exclusión mutua selectiva. Sincronización por condición general: La técnica “passing the baton”. Definición y aplicaciones. Alocaación de recursos. Ejemplo: SJN. Ejemplos clásicos: filósofos, lectores y escritores, productores y consumidores con buffer limitado, etc.. Semáforos en lenguajes reales: PascalFC, Pthreads. Ejemplos.

Unidad 5: SCM – Sincronización con Monitores: Noción de Regiones Críticas Condicionales. Monitores. Sintaxis y semántica. Sincronización en monitores. “signal and wait” y “signal and continue”. Diferencias. La técnica “passing the condition”. Problemas clásicos: buffer limitado, lectores y escritores, productor consumidor. El problema del peluquero. Rendezvous. Scheduling

de discos. Ejemplo. Enfoques alternativos para sincronización. Monitores en lenguajes reales: PascalFC, Java, Pthreads. Ejemplos. Conceptos de implementación de procesos en arquitecturas mono y multiprocesador. Kernel monoprocesador y multiprocesador.

Unidad 6: CPM - Concurrencia con Pasaje de Mensajes: Programación distribuida. Control de concurrencia en sistemas distribuidos. Diseño de un sistema de pasaje de mensajes. Primitivas. Mecanismos de comunicación y sincronización. Direccionamiento. Formato de mensajes.

Disciplinas de colas. Mensajes asincrónicos, sintaxis y semántica. Canales. Operaciones.

Exclusión mutua. Clases básicas de procesos. Filtros. Productores y consumidores. Clientes / Servidores: Monitores activos. Continuidad conversacional. Mensajes asincrónicos en lenguajes reales. Extensión de lenguajes secuenciales con bibliotecas específicas. Mensajes sincrónicos, sintaxis y semántica. Conceptos de CSP. Comunicación guardada, sintaxis y semántica. Filtros. Mensajes sincrónicos en lenguajes reales. Comunicación y sincronización. Ejemplos.

Unidad 7: LPC - Lenguajes de Programación Concurrente: Características de PascalFC, Linda, Java, ADA. Ejemplos. Comparación de alternativas. Resolución de problemas mediante diferentes paradigmas de interacción entre procesos: Servidores replicados. Algoritmos heartbeat. Algoritmos pipeline. Prueba-eco. Broadcast. Token passing. Manager/workers.

5. RECURSOS NECESARIOS

- Conexión A Internet, Pantalla TV Con HDMI

6. PROGRAMACIÓN SEMANAL

Semana	Unidad / Módulo	Descripción	Bibliografía
01	U01	CBC – Conceptos básicos de Concurrencia	(1) (2) (3) (4)
02	U01	CBC – Conceptos básicos de Concurrencia	(1) (2) (3) (4)
03	U02	CYS – Concurrencia y Sincronización	(1) (2) (3) (4)
04	U02	CYS – Concurrencia y Sincronización	(1) (2) (3) (4)
05	U03	CVC – Concurrencia con Variables Compartidas	(1) (2) (3) (4)
06	U03	CVC – Concurrencia con Variables Compartidas	(1) (2) (3) (4)
07	U04	SCS – Sincronización con Semáforos	(1) (2) (3) (4)
08	U04	SCS – Sincronización con Semáforos	(1) (2) (3) (4)
09	U05	SCM – Sincronización con Monitores	(1) (2) (3) (4)
10	U05	SCM – Sincronización con Monitores	(1) (2) (3) (4)
11	U06	CPM – Concurrencia con Pasaje de Mensajes	(1) (2) (3) (4)
12	U06	CPM – Concurrencia con Pasaje de Mensajes	(1) (2) (3) (4)
13	U06	CPM – Concurrencia con Pasaje de Mensajes	(1) (2) (3) (4)
14	U07	LPC – Lenguajes de Programación Concurrente	(1) (2) (3)
15	Apoyo y Exámenes	Parciales y entregas finales.	
16	Revisión, Cierre e Informe	Recuperatorios, Teóricos, cierre de notas e Informe final.	

7. BIBLIOGRAFIA DE LA ASIGNATURA

Autor	Año	Título	Capítulo/s	Lugar de la Edición	Editor / Sitio Web
(1) Gregory R. Andrews	2000	Foundations of Multithreaded, Parallel, and Distributed Programming.	1-8,10,12 (obligatoria)		Addison-Wesley
(2) Gregory R. Andrews	1991	Concurrent Programming. Principles and Practice.	1-10 (complem.)		Addison-Wesley
(3) William Stallings.	1995	Operating Systems	1,3-5 (obligatoria)		Prentice-Hall, Inc.
(4) M. Ben-Ari.	2006	Principles of Concurrent and Distributed Programming	1-10 (complem.)		Addison-Wesley
(5) Ronald Olsson, Aaron Keen	2004	The JR Programming Language: Concurrent programming in an Extended Java	completo		Springer
(6) Davies, Univ. Bradford, UK	1992	Apunte: Pascal-FC v5	completo. (complem.)		Univ. Bradford, UK
(7) Coppens, Univ. Católica Cba.	2008	Apunte: Introducción a Pascal-fc2	completo. (complem.)		Univ. Católica Cba.

Firma del docente-investigador responsable

VISADO		
COORDINADOR DE LA CARRERA	DIRECTOR DEL INSTITUTO	SECRETARIO ACADEMICO UNTDF
Fecha :	Fecha :	

Este programa de estudio tiene una validez de hasta tres años o hasta que otro programa lo reemplace en ese periodo