

**UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO****Licenciatura en Ciencias de la Computación****Facultad de Ciencias**

Programa de la asignatura

**Denominación de la asignatura:*****Computación Concurrente***

Clave:	Semestre: 8	Eje temático: Computación Teórica	No. Créditos: 10
Carácter: Obligatoria	Horas		Horas por semana
Tipo: Teórica	Teoría: 4	Práctica: 2	Total de Horas 96
Modalidad: Curso	Duración del programa: Semestral		

Asignatura con seriación obligatoria antecedente: Análisis de Algoritmos; Organización y Arquitectura de Computadoras; Computación Distribuida

Asignatura con seriación obligatoria subsecuente: Ninguna

Asignatura con seriación indicativa antecedente: Redes de Computadoras

Asignatura con seriación indicativa subsecuente: Ninguna

Objetivos generales:

Resolver problemas complejos de cómputo concurrente de manera crítica y creativa. Entender los aspectos de concurrencia y tolerancia a fallas de: sistemas operativos y bases de datos.

Índice temático

Unidad	Temas	Horas	
		Teóricas	Prácticas
I	Introducción	6	3
II	Exclusión mutua	12	6
III	Sincronización con y sin bloqueo	12	6
IV	Recursos múltiples	10	5
V	Problemas clásicos de sincronización	10	5
VI	Consenso	8	4
VII	Temas selectos	6	3
Total de horas:		64	32
Suma total de horas:		96	

Contenido temático

Unidad	Tema
I Introducción	
I.1	Sincronización.
I.2	El problema de la exclusión mutua.
I.3	Medidas de complejidad.
II Exclusión mutua	
II.1	Algoritmos para dos procesos.
II.2	Algoritmos de torneo.
II.3	Algoritmos de exclusión mutua.
II.4	Algoritmos libres de hambruna.
II.5	Algoritmos con espera ocupada (<i>busy waiting</i>).
II.6	Algoritmos adaptativos.
II.7	Algoritmos tolerantes a fallas.
II.8	Algoritmos simétricos.
III Sincronización con y sin bloqueo	
III.1	Primitivas de sincronización.
III.2	Bits <i>test-and-set</i> y evitar colisiones.
III.3	El algoritmo de incidente.
III.4	Otros algoritmos con espera ocupada y primitivas fuertes.
III.5	Estructuras de datos concurrentes.
III.6	Semáforos.
III.7	Monitores.
III.8	Justicia y objetos compartidos.
IV Recursos múltiples	
IV.1	Abrazos mortales.
IV.2	Prevención de abrazos mortales.
IV.3	El problema de los filósofos comensales.
IV.4	Estrategia de obtener y esperar.
V Problemas clásicos de sincronización	
V.1	Productor y consumidor.
V.2	Lectores y escritores.
V.3	El barbero dormilón.
V.4	El fumador de cigarrillos.
V.5	Escritura y lectura de relojes.
VI Consenso	
VI.1	Introducción y algoritmos básicos.
VI.2	Consenso utilizando una cola compartida.
VI.3	Imposibilidad de consenso con un proceso fallible.
VI.4	El poder relativo de las primitivas de sincronización.
VI.5	Universalidad del consenso.
VII Temas selectos	
VII.1	Tolerancia a fallas benignas y bizantinas.
VII.2	Transacciones en memoria compartida

Bibliografía básica:

1. Gadi Taubenfeld, *Synchronization Algorithms and Concurrent Programming*, Pearson / Prentice Hall, 2006.
2. Maurice Herlihy y Nir Shavit, *The Art of Multiprocessor Programming*, Morgan Kaufmann, 2008.

Bibliografía complementaria:

1. Leslie Lamport, *Specifying Systems: The TLA+ Language and Tools for Hardware and Software Engineers*, Pearson Education, Inc., 2002.

Sugerencias didácticas:

Exposición oral	(X)
Exposición audiovisual	()
Ejercicios dentro de clase	(X)
Ejercicios fuera del aula	()
Seminarios	()
Lecturas obligatorias	()
Trabajo de investigación	()
Prácticas de taller o laboratorio	(X)
Prácticas de campo	()

Otras: _____

Métodos de evaluación:

Exámenes parciales	(X)
Examen final escrito	()
Trabajos y tareas fuera del aula	(X)
Prácticas de laboratorio	()
Exposición de seminarios por los alumnos	()
Participación en clase	(X)
Asistencia	()
Proyectos de programación	(X)
Proyecto final	()
Seminario	()

Otras: _____

Perfil profesiográfico:

Egresado preferentemente de la Licenciatura en Ciencias de la Computación o Matemático con especialidad en computación con amplia experiencia de programación. Es conveniente que posea un posgrado en la disciplina. Con experiencia docente.