

PROGRAMA DE ESTUDIOS: SISTEMAS DISTRIBUIDOS

**PROTOCOLO**

Fechas	Mes/año
Elaboración	10-2007
Aprobación	
Aplicación	10-2007

Clave			Semestre	9°	
Nivel	Licenciatura	X	Maestría	Doctorado	
Ciclo	Integración		Básico	Superior	X
Colegio	H. y C.S.		C. y T.	X C. y H.	

**Plan de estudios del que forma parte:** Ingeniería de Software

**Propósito(s) general(es):**  
 El objetivo del curso es comprender los aspectos generales de los sistemas distribuidos. Uno de los aspectos prácticos del curso es la parte de sockets que se imparte. Aparte de sockets se tocan los temas de comunicación grupal, modelo cliente/servidor, NIS, NFS, sistemas de archivos distribuidos y memoria compartida distribuida.

Carácter		Modalidad				Horas de estudio semestral (16 semanas)					
Indispensable	X	Seminario		Taller		Con Docente	Teóricas	36	Autónomas	Teóricas	20
		Curso	X	Curso-taller			Prácticas	36		Prácticas	36
Optativa *		Laboratorio	X	Clínica		Carga horaria semanal: 4.5 + 3.5 = 8			Carga horaria semestral: 128		

Asignaturas Previas:	Asignaturas Posteriores:
Sistemas operativos, Arquitectura de computadoras, Redes de computadoras.	

<b>Requerimientos para cursar la asignatura</b>	Conocimientos: dominio de conceptos fundamentales de programación y redes de computadoras. Habilidades: para la búsqueda sistemática de soluciones, capacidad para plantear y resolver problemas por medio del razonamiento lógico.
---	--

<b>Perfil deseable del profesor:</b>	Con Maestría en Ingeniería de Software o algún grado de maestría afín, o bien, amplios conocimientos y experiencia laboral en el área. Además, amplia experiencia docente y de investigación.
--------------------------------------	---

<b>Academia responsable del programa:</b> Informática	<b>Diseñador (es):</b> Profr. Jorge Wals Selvas
--	--

\* Aquellas en las que se ofrece la posibilidad de cursar una de las asignaturas, para cubrir un requisito INDISPENSABLE será considerada INDISPENSABLE.

## INTRODUCCIÓN

La función de la materia Sistemas Distribuidos dentro de la Licenciatura en Ingeniería de software es la de proporcionar el estudiante conocimientos sólidos, así como métodos y técnicas que sirvan para utilizar y desarrollar sistemas que funcionen en un entorno distribuido.

Para su estudio se deben tener conocimientos de Redes de computadoras (1-CT-IS-20).

## PROPÓSITOS GENERALES

El objetivo del curso es comprender los aspectos generales de los sistemas distribuidos. Uno de los aspectos prácticos del curso es la parte de sockets que se imparte. Aparte de sockets se tocan los temas de comunicación grupal, modelo cliente/servidor, NIS, NFS, sistemas de archivos distribuidos y memoria compartida distribuida.

## CONTENIDOS

TEMAS Y SUBTEMAS	PROPÓSITOS ESPECIFICOS
<b>1. Introducción Sistemas Distribuidos</b>	El estudiante comprenderá la importancia de los Sistemas distribuidos, así como las aplicaciones que actualmente los utilizan.
1.1. Concepto de Sistemas Distribuidos 1.2. Aplicaciones actuales de los Sistemas Distribuidos	
<b>2. Comunicación a través de mensajes: los sockets de Unix</b>	El estudiante conocerá el concepto de sockets y sus aplicaciones en los Sistemas distribuidos.
2.1. Comunicación entre procesos 2.2. Sockets 2.3. Caso de estudio: comunicación entre procesos en Unix	
<b>3. Sistemas de archivos distribuidos</b>	
3.1. NIS: Network Information Service 3.2. NFS: Network File System	El estudiante analizará y comparará las ventajas y desventajas de los diferentes sistemas de archivos distribuidos, para comprender cuales son más eficientes.
<b>4. Procesos y procesadores distribuidos</b>	El estudiante identificará las principales arquitecturas de capas de software y también las principales arquitecturas de procesadores distribuidos.
4.1. Modelos arquitectónicos: capas de software 4.2. Modelos arquitectónicos: arquitecturas de procesadores en Sistemas distribuidos	
<b>5. Memoria compartida distribuida: DSM</b>	El estudiante analizará las formas de gestión y almacenamiento de datos en un Sistema distribuido.
5.1. Introducción a DSM 5.2. Implementación a DSM	
<b>6. El RPC de Unix</b>	El estudiante revisará los aspectos principales de un Remote Procedure Call, tomando como caso de estudio utilizado en el Sistema Operativo UNIX.
6.1. Concepto de RPC 6.2. Caso de estudio: RPC de Unix	

<b>7. La comunicación grupal</b>	El estudiante identificará los esquemas básicos de comunicación grupal en Sistemas distribuidos y analizará los mecanismos implementados para su escalabilidad.
7.1. Esquemas básicos de comunicación grupal 7.2. Escalabilidad en sistemas de comunicación grupal	
<b>8. Introducción a los algoritmos distribuidos</b>	El estudiante conocerá las principales características de los algoritmos distribuidos y también de dos elementos fundamentales para los Sistemas distribuidos como el Middleware y los Objetos distribuidos.
8.1. Características de los algoritmos distribuidos 8.2. Conceptos fundamentales del Middleware 8.3. Objetos distribuidos	
<b>9. Relojes distribuidos</b>	
9.1. Sincronización 9.2. Relojes lógicos 9.3. Exclusiones mutuas	El estudiante identificará las formas para gestionar los relojes en de los Sistemas distribuidos.
<b>10. El modelo cliente/servidor</b>	El estudiante revisará los conceptos del modelo cliente/servidor desde la perspectiva específica de los Sistemas distribuidos.
10.1. Transparencia del lado del cliente para software distribuido 10.2. Variaciones en el modelo cliente-servidor	
<b>11. El interbloqueo distribuido</b>	El estudiante identificará un interbloqueo (deadlock) en un Sistema distribuido, también conocerá los principales protocolos distribuidos para manejo de transacciones.
11.1. Concepto de interbloqueo 11.2. Protocolos distribuidos de transacciones 11.3. Detección distribuida de interbloqueos	

### METODOLOGÍA PARA EL CURSO

Se trata de un curso teórico práctico, que se impartirá por medio de tres elementos: clases teóricas, prácticas y asesorías, de acuerdo al modelo educativo de la UACM. Estos tres elementos se efectuarán en cualquiera de las siguientes modalidades: conferencia, interrogatorio, resolución de ejercicios en clase y de tarea, lluvia de ideas, uso de software, grupos de discusión, trabajo en equipo, análisis y demostración de problemas, diseño de un prototipo, investigación bibliográfica.

### EVALUACIÓN DIAGNÓSTICA

Se aplicará un examen escrito para verificar si el estudiante cumple con los conocimientos previos requeridos, los conceptos previos a evaluar serán principalmente los adquiridos en el curso de Redes de computadoras.

### EVALUACIÓN FORMATIVA

Se recomienda realizar al menos una evaluación formativa al finalizar cada una de las unidades temáticas para que el estudiante tenga una retroalimentación adecuada sobre la adquisición de los conocimientos durante el semestre y no hasta el final del mismo, de esta forma se cuenta con tiempo para que el estudiante realice ajustes sobre su desempeño académico en el curso.

### EVALUACIÓN DE CERTIFICACIÓN

La evaluación de certificación, constara de dos partes que se explican a continuación:

La parte I es el instrumento o evaluación *conceptual-práctica*, tiene una ponderación del 60% y se puede aplicar a lo largo del semestre con prácticas de laboratorio definidas por cada Profesor, o bien, presentar una evaluación escrita final definida por el comité de certificación, que consta de preguntas y ejercicios de acuerdo con el contenido temático.

La parte II que es el restante 40% de la evaluación consiste en un *proyecto de Sistemas Distribuidos* en donde el estudiante aplicará los conocimientos adquiridos. El estudiante deberá entregar a su profesor un documento impreso con la documentación completa de su diseño final, además de una copia electrónica con los archivos que lo integran.

### BIBLIOGRAFÍA

La bibliografía que se enlista a continuación se considera como básica para estudiante y profesor:

Bibliografía	Temas para los que se recomienda
1. Andrew S. Tanenbaum, Maarten van Steen; Distributed Systems: Principles and Paradigms, Ed. Prentice Hall, 2001.	Todos
2. K. Robbins, S. Robbins; Unix Systems Programming: Communication, Concurrency and Threads, Ed. Prentice Hall; 2003, 2nd edition.	Todos
3. George Coulouris, Jean Dollimore, Tim Kindberg; Distributed Systems, Concepts and Design, Tercera Edición, Ed. Addison Wesley, 2000.	Todos
4. Andrews, G.A. "Concurrent Programming – Principles and Practice"; The Benjamin/Cumming Publishing Company, 1991.	Todos
5. Lewis, Bill; Programación Multithread en JAVA; Sun Microsystems; 2003.	Todos
6. Nancy Lynch; Distributed Algorithms, Ed. Morgan Kaufmann Publishers 1996.	Todos
7. Sape Mullender; Distributed Systems. 2a. Edición Addison-Wesley 1993.	Todos
8. Andrew S. Tanenbaum; Sistemas Operativos Distribuidos, Ed. Prentice Hall, 1995	Todos

### OTROS RECURSOS

El laboratorio de cómputo, deberá contar con los siguientes recursos: 30 computadoras personales con procesador a 3GH conectadas en red y con internet, dispositivo óptico cdrw/dvd 24x, 2 Gb de RAM, 160 GB en disco, monitor a color de alta resolución, 1 Impresora laser, 1 Servidor para ambiente de redes, procesador xeon 2.8ghz/1mb con bus de 800 mhz, 2Gb de RAM, disco duro SCSI de 160 gb, dispositivo óptico cdrw/dvd 24x, adaptador de red (necesario a partir del tercer año de la carrera).