



Asignatura:

Obligatoria

Optativa

Horas:

Teóricas

Prácticas

Total (horas):

Semana

16 Semanas

Modalidad: Curso.

Asignatura obligatoria antecedente: Ninguna.

Asignatura obligatoria consecuente: Ninguna.

Objetivo(s) del curso:

El alumno comprenderá y aplicará los conocimientos de protocolos, métodos y estándares sobre redes de datos, así como de criptografía y seguridad para que usando un lenguaje de programación cree programas bajo el esquema cliente/servidor.

Temario

NÚM.	NOMBRE	HORAS
1.	Conceptos básicos	6.0
2.	Creación de socket servidor y cliente	10.0
3.	Servidores y clientes sincronizados	10.0
4.	Sockets broadcasting y multicasting	6.0
5.	Implantación de servidores con criptografía y código seguro	8.0
6.	Creación de algoritmos de routing	8.0
Total		48.0



1 Conceptos básicos

Objetivo: El alumno definirá y explicará las funciones de los sockets y puertos como sus principales características y estándares.

Contenido:

- 1.1 Definición de Puerto.
- 1.2 Socket.
 - 1.2.1 Definición de Socket.
 - 1.2.2 Socket en UNIX.
 - 1.2.3 Socket en MAC.
 - 1.2.4 Socket en DOS.
 - 1.2.5 Familia de Sockets
 - 1.2.5.1 AF_UNIX
 - 1.2.5.2 AF_INET
 - 1.2.5.3 AF_CCITT
 - 1.2.5.4 AF_NS
- 1.3 Concepto Cliente/Servidor.
 - 1.3.1 Proceso Cliente.
 - 1.3.2 Proceso Servidor.
- 1.4 Definición de stream.
- 1.5 Concurrencia, Multiprogramación y Multitarea.
- 1.6 Servidores Interactivos y Concurrentes.
- 1.7 Estándares
 - 1.7.1 POSIX.
 - 1.7.2 RFC.
 - 1.7.3 X/OPEN.

2 Creación de socket cliente/servidor

Objetivo: El alumno interpretará e identificará los sockets en TCP y UDP y entenderá los conceptos de hilos en el modelo cliente/servidor.

Contenido:

- 2.1 Socket en TCP.
 - 2.1.1 Servidor eco.
 - 2.1.1.1 Creación del Servidor Socket.
 - 2.1.1.2 Conexión del Servidor Socket.
 - 2.1.1.3 Terminación del Servidor Socket.
 - 2.1.2 Cliente Eco.
 - 2.1.2.1 Creación del Cliente Socket.
 - 2.1.2.2 Conexión del Cliente Socket.
 - 2.1.2.3 Terminación del Cliente Socket.



- 2.2 Socket en UDP.
 - 2.2.1 Servidor eco.
 - 2.2.1.1 Creación del Servidor Socket.
 - 2.2.1.2 Conexión del Servidor Socket.
 - 2.2.2 2.3.1.3 Terminación del Servidor Socket.
 - 2.2.3 Cliente eco.
 - 2.2.4 2.2.2.1 Creación del Cliente Socket.
 - 2.2.5 2.2.2.2 Conexión del Cliente Socket.
 - 2.2.6 2.2.2.3 Terminación del Cliente Socket.
- 2.3 Definición de DAEMON.
 - 2.3.1 Servidor eco.
 - 2.3.2 Cliente eco.
- 2.4 Concepto de Hilos.
 - 2.4.1 Servidor eco.
 - 2.4.2 Cliente eco.
 - 2.4.3 Servidor Talk.
 - 2.4.4 Cliente Talk.

3 Servidores y clientes sincronizados

Objetivo: El alumno aplicará y conocerá los servidores orientados y no orientados a conexión involucrando algunos conceptos como semáforos y sincronización.

Contenido:

- 3.1 Procesos.
- 3.2 Semáforos.
- 3.3 Lectura y Escritura de Archivos.
- 3.4 Sincronización.
- 3.5 Desarrollo de Aplicaciones
 - 3.5.1 Servidor finger.
 - 3.5.2 Cliente finger.
 - 3.5.3 Servidor de ejecución remota.
 - 3.5.4 Cliente de ejecución remota.
 - 3.5.5 Servidor HTTP.
- 3.6 Condición de Variables.
 - 3.6.1 Servidor FTP.
 - 3.6.2 Servidor SMTP.
- 3.7 Servidores Orientados a Conexión.
- 3.8 Servidores No Orientados a Conexión.

4 Sockets broadcasting y multicasting

Objetivo: El alumno conocerá e identificará los diferentes sockets para difundir información mediante la red de datos.



Contenido:

- 4.1 Broadcast.
 - 4.1.1 Definición de Broadcast.
 - 4.1.2 Implantación del algoritmo Broadcast.
 - 4.1.3 Creación del socket Broadcast.
- 4.2 Multicast.
 - 4.2.1 Definición de Multicast.
 - 4.2.2 Implantación del algoritmo Multicast.
 - 4.2.3 Creación del socket Multicast.
- 4.3 Servidor SNTP (Simple Network Time Protocol).

5 Implantación de servidores con criptografía y código seguro

Objetivo: El alumno empleará técnicas de programación y algoritmos de cifrado para hacer robustas las aplicaciones de las redes de datos.

Contenido:

- 5.1 Servidores con criptografía.
 - 5.1.1 Servidor Eco.
 - 5.1.2 Servidor Talk.
 - 5.1.3 Servidor HTTP.
 - 5.1.4 Servidor FTP.
 - 5.1.5 Servidor Secure Shell.
- 5.2 Clientes con criptografía.
 - 5.2.1 Cliente Eco.
 - 5.2.2 Cliente Talk.
 - 5.2.3 Cliente HTTP.
 - 5.2.4 Cliente FTP.
 - 5.2.5 Cliente Secure Shell.
- 5.3 Servidores y clientes implantando código seguro.

6 Creación de algoritmos de routing

Objetivo: El alumno enunciará los diferentes algoritmos de ruteo para demostración de los mismos mediante programación.

Contenido:

- 6.1 RIP.
 - 6.1.1 Definición del Algoritmo.
 - 6.1.2 Creación del Algoritmo.
 - 6.1.3 Implantación del Algoritmo.
- 6.2 OSPF.
 - 6.2.1 Definición del Algoritmo.
 - 6.2.2 Creación del Algoritmo.
 - 6.2.3 Implantación del Algoritmo.

**Bibliografía básica:****Temas para los que se recomienda:**

STEVENS, Richard <i>UNIX Network Programming</i> U.S.A Prentice Hall, 1990	1, 2, 3, 4, 5 y 6
STEVENS, Richard <i>UNIX Network Programming, Volume I</i> 3rd. Edition U.S.A Addison Wesley, 2003	1, 2, 3, 4, 5 y 6
STEVENS, Richard <i>UNIX Network Programming, Volume II</i> 2nd. Edition U.S.A Prentice Hall, 1998	1, 2, 3, 4, 5 y 6
Bibliografía complementaria:	
COMER, Douglas E. <i>Interconectividad de Redes con TCP/IP Vol. I</i> <i>Principios Básicos y Arquitectura</i> 3a. Edición México Prentice Hall, 2000	1, 2, 3 y 4
COMER, Douglas E. <i>Internetworking with TCP/IP Vol. III.</i> <i>Client server Programming and applications</i> 3rd. Edition USA Prentice Hall, 2000	1, 2, 3 y 4
MÁQUEZ GARCÍA, Francisco Manuel <i>Unix Programación Avanzada</i> España Ra-ma , 1993	1, 2



OUALLINE, Steve
Practical C Programming
3rd. Edition
USA
O'Reilly, 1997

1, 3

ROBBINS, Kay A., ROBBINS, Steven
Unix Programación Práctica
[s.l.i] USA
Prentice Hall, 1997.

1, 2 y 3

Sugerencias didácticas:

Exposición oral
Exposición audiovisual
Ejercicios dentro de clase
Ejercicios fuera del aula
Seminarios

Lecturas obligatorias
Trabajos de investigación
Prácticas de taller o laboratorio
Prácticas de campo
Otras

Forma de evaluar:

Exámenes parciales
Exámenes finales
Trabajos y tareas fuera del aula

Participación en clase
Asistencias a prácticas
Otras

Perfil profesiográfico de quienes pueden impartir la asignatura

El profesor deberá contar con licenciatura, preferentemente de las carreras: Ingeniero en Computación, Ingeniero en Electrónica, Ingeniero en Telecomunicaciones, Licenciado en Ciencias Computacionales o formación equivalente y contar con amplia experiencia en seguridad en informática, diseño, desarrollo y control de arquitecturas cliente-servidor.