

**1. Datos Generales de la asignatura**

<b>Nombre de la asignatura:</b>	<b>Arquitectura Cliente Servidor</b>
<b>Clave de la asignatura:</b>	<b>Sli-1603</b>
<b>SATCA<sup>1</sup>:</b>	<b>3-4-7</b>
<b>Carrera:</b>	<b>Ingeniería en Sistemas Computacionales</b>

**2. Presentación**

<b>Caracterización de la asignatura</b>
Esta asignatura proporciona dominio de las herramientas básicas para poder configurar y administrar servicios e infraestructuras de redes con arquitectura cliente servidor e implementar políticas de seguridad con el propósito de mejorar la fiabilidad y el desempeño de las mismas. Se debe privilegiar el uso de diferentes plataformas en el desarrollo de las prácticas de la materia, de tal suerte que esta aporte a la capacidad del futuro ingeniero de integrar soluciones con diferentes tecnologías, plataformas y dispositivos
<b>Intención didáctica</b>
La asignatura debe ser abordada desde un enfoque eminentemente práctico, aplicando los atributos funcionales de la arquitectura cliente servidor a la implementación de servicios, monitoreo, administración de la configuración y desempeño para la resolución de casos diseñados para simular situaciones reales con herramientas de software.  Asimismo, deberán de instalarse y configurarse diferentes servicios de red, monitorear y analizar el tráfico que se genera con su uso y proveer mecanismos básicos de seguridad física y lógica. En la Unidad I se exponen conceptos básicos como son la definición de puerto y socket, así como los conceptos básicos de cliente servidor y multitarea que le permitirán al estudiante tener bases sólidas para el uso de esta arquitectura. En la unidad II se aplican los conocimientos teóricos para poder crear de forma práctica un socket, socket udp, cliente eco y servidores eco En la Unidad III se abordó todo lo relacionado a servidores y clientes, en donde el estudiante conocerá todo lo relacionado a procesos, semáforos y sincronización así como servidores orientados a conexión y no orientados a conexión. En la Unidad IV se estudiarán los sockets, broadcasting y multicasting. En la Unidad V el alumno implantará servidores con criptografía y métodos con códigos seguros utilizando el cliente http y ftp. En la Unidad VI el alumno será capaz de elaborar e implementar algoritmos rip.

<sup>1</sup> Sistema de Asignación y Transferencia de Créditos Académicos

### 3. Participantes en el diseño y seguimiento curricular del programa

Lugar y fecha de elaboración o revisión	Participantes	Observaciones
Instituto Tecnológico de Tláhuac, México D.F. 18 de Mayo de 2012.	Academia de Sistemas Y Computación.	Revisión y actualización de contenidos temáticos del programa de estudios para esta especialidad.
Instituto Tecnológico de Tláhuac, CDMX. 11 de Abril de 2016.		

### 4. Competencia(s) a desarrollar

Competencia(s) específica(s) de la asignatura
Configurar y administrar servicios de red utilizando la arquitectura cliente servidor para el uso eficiente y confiable de los recursos computacionales.

### 5. Competencias previas

<ul style="list-style-type: none"> <li>• <i>Identificar y aplicar conceptos fundamentales de las telecomunicaciones, para analizar redes computacionales.</i></li> <li>• <i>Utilizar metodologías para el análisis de requerimientos, planeación, diseño e instalación de una red.</i></li> <li>• <i>Utilizar normas y estándares de la industria para diseñar e integrar soluciones de red dentro de las organizaciones.</i></li> <li>• <i>Seleccionar, conocer y usar adecuadamente los diferentes sistemas operativos para lograr un uso más eficiente así como diferenciar y aplicar las técnicas de manejo de recursos para el diseño, organización, utilización y optimización de los sistemas operativos. También conocer y saber usar técnicas y/o herramientas de administración de los sistemas operativos para la optimización de recursos existentes</i></li> </ul>
---

**6. Temario**

No.	Temas	Subtemas
	Conceptos básicos	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 un socket Cliente – Servidor	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 Terminación del Servidor Socket. 2.2.3 Cliente eco. 2.2.4 Creación del Cliente Socket. 2.2.5 Conexión del Cliente Socket. 2.2.6 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	3.1 Procesos. 3.2 Semáforos.

		<p>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</p>
4	Sockets Broadcasting y Multicasting	<p>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).</p>
5	Implantación de servidores con criptografía y código seguro	<p>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.</p>
6	Creación de Routing	<p>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.</p>

## 7. Actividades de aprendizaje de los temas

Unidad 1 Conceptos básicos	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> Comprender los conceptos básicos para la arquitectura cliente servidor para aplicarlas en el aseguramiento y optimización del desempeño de las mismas</p> <p><b>Genéricas:</b></p> <p><b>Competencias instrumentales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis</li> <li>• Capacidad de organizar información</li> <li>• Habilidad para buscar y analizar información proveniente de fuentes diversas</li> <li>• Iniciativa y autonomía dentro de los parámetros del diseño.</li> <li>• Solución de problemas</li> <li>• Responsabilidad por materiales y equipos, personal y trabajos asignados</li> </ul> <p><b>Competencias interpersonales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad crítica y autocrítica</li> <li>• Trabajo en equipo</li> <li>• Habilidades interpersonales</li> </ul> <p><b>Competencias sistémicas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica</li> <li>• Habilidades de investigación</li> <li>• Capacidad de aprender</li> <li>• Capacidad de generar nuevas ideas</li> <li>• Habilidad para trabajar en forma Autónoma</li> </ul>	<p>Investigar el funcionamiento de puerto y socket e identificar su correcto funcionamiento con la finalidad de aplicarlos en la arquitectura cliente servidor.</p> <p>Realizar un ensayo relacionado a la multiprogramación y multitarea y posteriormente plantear un problema real el cual sea solucionado con los temas anteriores.</p> <p>Discutir de manera grupal el funcionamiento de los servidores interactivos y concurrentes, así como aclarar sus ventajas y sus desventajas.</p> <p>Realizar una lectura relacionada a la arquitectura cliente servidor y posteriormente hacer un pequeño resumen, se deberá mencionar la fuente y el autor de esta.</p>
Unidad 2 Creación de un socket cliente servidor.	
Competencias	Actividades de aprendizaje
<p><b>Específica(s):</b> Crear, configurar y administrar un socket cliente/servidor para satisfacer las necesidades de una organización</p>	<p>Identificar las características de los procesos y los semáforos y explicar de manera breve su funcionamiento así como alguna de sus características.</p> <p>Recopilar información relacionada al</p>

<p><b>Genéricas:</b></p> <p><b>Competencias instrumentales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis</li> <li>• Capacidad de organizar información</li> <li>• Habilidad para buscar y analizar información proveniente de fuentes diversas</li> <li>• Iniciativa y autonomía dentro de los parámetros del diseño.</li> <li>• Solución de problemas</li> <li>• Responsabilidad por materiales y equipos, personal y trabajos asignados</li> </ul> <p><b>Competencias interpersonales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad crítica y autocrítica</li> <li>• Trabajo en equipo</li> <li>• Habilidades interpersonales</li> </ul> <p><b>Competencias sistémicas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica</li> <li>• Habilidades de investigación</li> <li>• Capacidad de aprender</li> <li>• Capacidad de generar nuevas ideas</li> </ul> <p>Habilidad para trabajar en forma Autónoma</p>	<p>servidor FTP y SMPT, posteriormente realizar un cuadro comparativo en donde se listen las ventajas y las desventajas de cada uno de ellos.</p> <p>Elaborar un ensayo relacionado a los servidores orientados a conexión y los servidores no orientados a conexión, y ejemplificar detalladamente una aplicación con la vida cotidiana.</p>
<p><b>Unidad 3 Servidores y Clientes.</b></p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s): Planear, diseñar e implementar el uso y funcionamiento de los sockets broadcasting y multicasting, así como definir cada una de sus principales aplicaciones.</p> <p><b>Genéricas:</b></p> <p><b>Competencias instrumentales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis</li> <li>• Capacidad de organizar información</li> <li>• Habilidad para buscar y analizar información proveniente de fuentes diversas</li> <li>• Iniciativa y autonomía dentro de los parámetros del diseño.</li> </ul>	<p>Investigar los conceptos relacionados con el funcionamiento de los sockets broadcasting y multicasting y posteriormente crear un cuadro comparativo en donde se detalle cada una de las características de los conceptos mencionados.</p> <p>Desarrollar una solución para una aplicación de la vida cotidiana utilizando los sockets.</p> <p>Identificar los componentes básicos de los sockets broadcasting y multicasting y enlistar sus principales ventajas y desventajas.</p>

<ul style="list-style-type: none"> <li>• Solución de problemas</li> <li>• Responsabilidad por materiales y equipos, personal y trabajos asignados</li> </ul> <p><b>Competencias interpersonales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad crítica y autocrítica</li> <li>• Trabajo en equipo</li> <li>• Habilidades interpersonales</li> </ul> <p><b>Competencias sistémicas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica</li> <li>• Habilidades de investigación</li> <li>• Capacidad de aprender</li> <li>• Capacidad de generar nuevas ideas</li> </ul> <p>Habilidad para trabajar en forma Autónoma</p>	
<p><b>Unidad 4 Sockets Broadcasting y Multicasting.</b></p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s): Planear, diseñar e implementar el uso y funcionamiento de los sockets broadcasting y multicasting, así como definir cada una de sus principales aplicaciones</p> <p><b>Genéricas:</b></p> <p><b>Competencias instrumentales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis</li> <li>• Capacidad de organizar información</li> <li>• Habilidad para buscar y analizar información proveniente de fuentes diversas</li> <li>• Iniciativa y autonomía dentro de los parámetros del diseño.</li> <li>• Solución de problemas</li> <li>• Responsabilidad por materiales y equipos, personal y trabajos asignados</li> </ul> <p><b>Competencias interpersonales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad crítica y autocrítica</li> <li>• Trabajo en equipo</li> <li>• Habilidades interpersonales</li> </ul> <p><b>Competencias sistémicas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica</li> </ul>	<p>Investigar los conceptos relacionados con el funcionamiento de los sockets broadcasting y multicasting y posteriormente crear un cuadro comparativo en donde se detalle cada una de las características de los conceptos mencionados.</p> <p>Desarrollar una solución para una aplicación de la vida cotidiana utilizando los sockets.</p> <p>Identificar los componentes básicos de los sockets broadcasting y multicasting y enlistar sus principales ventajas y desventajas.</p>

<ul style="list-style-type: none"> <li>• Habilidades de investigación</li> <li>• Capacidad de aprender</li> <li>• Capacidad de generar nuevas ideas</li> </ul> <p>Habilidad para trabajar en forma Autónoma</p>	
<p><b>Unidad 5 Implantación de servidores con criptografía y código seguro.</b></p>	
<p>Competencias</p>	<p>Actividades de aprendizaje</p>
<p>Específica(s): Aplicar herramientas y políticas de seguridad para mejorar la integridad, disponibilidad y confidencialidad de los recursos de un servidor</p> <p><b>Genéricas:</b></p> <p><b>Competencias instrumentales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de análisis y síntesis</li> <li>• Capacidad de organizar información</li> <li>• Habilidad para buscar y analizar información proveniente de fuentes diversas</li> <li>• Iniciativa y autonomía dentro de los parámetros del diseño.</li> <li>• Solución de problemas</li> <li>• Responsabilidad por materiales y equipos, personal y trabajos asignados</li> </ul> <p><b>Competencias interpersonales:</b></p> <ul style="list-style-type: none"> <li>• Capacidad crítica y autocrítica</li> <li>• Trabajo en equipo</li> <li>• Habilidades interpersonales</li> </ul> <p><b>Competencias sistémicas:</b></p> <ul style="list-style-type: none"> <li>• Capacidad de aplicar los conocimientos en la práctica</li> <li>• Habilidades de investigación</li> <li>• Capacidad de aprender</li> <li>• Capacidad de generar nuevas ideas</li> </ul> <p>Habilidad para trabajar en forma Autónoma</p>	<p>Colaborar para el diseño de un esquema de seguridad en un servidor identificando los elementos más importantes en este proceso.</p> <p>Presentar de forma oral el trabajo así como discutir y retroalimentar los reportes.</p> <p>Identificar los componentes críticos de un servidor en una organización. Definir y aplicar un esquema de seguridad que mejore la confiabilidad, confidencialidad y disponibilidad de los servicios.</p>



## 8. Práctica(s)

1. El objetivo de esta práctica es la familiarización con aplicaciones elementales que usan los servicios de transporte de los sockets UDP. A lo largo de la práctica se realizarán diversas pruebas con una aplicación sencilla cliente/servidor escrita en lenguaje C. El servicio que realiza la aplicación distribuida consiste en una especie de servidor de hecho; es decir, el servidor recibe un mensaje y lo reenvía de nuevo al cliente que se lo envió. Recuerda que cuando la comunicación entre dos procesos de aplicación usa los servicios de transporte UDP, cada extremo de la comunicación identifica a su socket UDP por la dirección y puerto local (por lo tanto, un paquete que llega a un equipo será entregado al socket cuyo identificador local coincida con la dirección IP destino y puerto destino del segmento UDP).

2. partir de dos programas escritos en lenguaje Python, teniendo los alumnos que realizar las siguientes tareas:

1. Descargar desde la página web de la asignatura el fichero comprimido ejemplos\_c\_s.zip.

2. Descomprimir el contenido del fichero .zip dentro de la carpeta personal (/home/curso).

3. Abrir un terminal y entrar dentro de la carpeta c\_s\_basico: `$ cd ejemplos_c_s/c_s_basico`

4. Analizar los códigos proporcionados, leyendo y comprendiendo los comentarios. Para visualizar estos códigos usar un editor de texto: `$ gedit servidor_basico.py cliente_basico.py` 5. Ejecutar las aplicaciones servidor y cliente:

Primero, lanzar el servidor: `$ python servidor_basico.py 9000`

Segundo, lanzar en un nuevo terminal el cliente: `$ python cliente_basico.py localhost 9000` 6. Comprobar el correcto funcionamiento de la aplicación leyendo los mensajes que se imprimen en los terminales.

A continuación, responder brevemente a las siguientes cuestiones: 1. Dentro de la aplicación servidor, con la llamada a la función `listening_socket.listen`, ¿en qué estado solicitamos que se abra el conector (socket) creado con la llamada a esta función

2. Dentro de la aplicación cliente, con la llamada a la función `connection_socket.connect`, ¿con quién se está intentando establecer una conexión y en qué puerto?

3. Con la llamada a la función `listening_socket.accept`, dentro de la aplicación servidor, ¿qué estamos aceptando? ¿Qué se ha establecido?

4. ¿Qué ocurre si ejecutamos antes el cliente que el servidor? partiendo de la aplicación de la practica 1, vamos a mejorar los dos programas para que tanto el cliente como el servidor puedan interactuar (intercambiar mensajes de texto). Los alumnos realizarán las siguientes tareas: 1. Estando dentro del directorio c\_s\_basico, moverse al directorio c\_s\_interactivo: `$ cd ../c_s_interactivo`.

2. Analizar los códigos proporcionados, leyendo y comprendiendo los comentarios. Para visualizar estos códigos usar un editor de texto: `$ gedit servidor_interactivo.py cliente_interactivo.py` 3. Ejecutar las aplicaciones servidor y cliente:

Primero, lanzar el servidor: `$ python servidor_interactivo.py 9000`

Segundo, lanzar en un nuevo terminal el cliente: `$ python cliente_interactivo.py localhost 9000` 4. Escribir mensajes tanto en la parte del cliente como en la del servidor y comprobar el correcto funcionamiento de la aplicación. A continuación, responder brevemente a las siguientes cuestiones:

1. ¿Qué llamadas deben hacerse desde la línea de comando para conectarse a otra máquina que no sea el localhost? Conéctese a la máquina de su compañero e

interaccione con él.

**3.-**Se propone el desarrollo de un servicio web que juegue al juego de Batalla Naval o Guerra de Barcos. Crear un servicio web utilizando XML-RPC que implemente la guerra de barcos. Que se trate de un servicio web implica que haya unas funciones accesibles remotamente a través del puerto 80 y con mensajes que funcionan sobre el protocolo http. En esta práctica se utilizara el puerto 18080 en vez del 80 para ofrecer el servicio web. Para conocer que el servicio está en marcha al iniciarse la aplicación se enviará un anuncio-presentación al grupo multicast 224.0.2.3 y puerto de destino 18080, pues no existe un directorio centralizado en el que anunciarse, donde deberían escuchar los posibles clientes del servicio como puede ser el árbitro de una liguilla de Batalla Naval. El alumno deberá implementar la aplicación del servicio web, no es necesario crear aplicaciones cliente.

## 9. Proyecto de asignatura

El objetivo del proyecto que planteé el docente que imparta esta asignatura, es demostrar el desarrollo y alcance de la(s) competencia(s) de la asignatura, considerando las siguientes fases:

- **Fundamentación:** marco referencial (teórico, conceptual, contextual, legal) en el cual se fundamenta el proyecto de acuerdo con un diagnóstico realizado, mismo que permite a los estudiantes lograr la comprensión de la realidad o situación objeto de estudio para definir un proceso de intervención o hacer el diseño de un modelo.
- **Planeación:** con base en el diagnóstico en esta fase se realiza el diseño del proyecto por parte de los estudiantes con asesoría del docente; implica planificar un proceso: de intervención empresarial, social o comunitario, el diseño de un modelo, entre otros, según el tipo de proyecto, las actividades a realizar los recursos requeridos y el cronograma de trabajo.
- **Ejecución:** consiste en el desarrollo de la planeación del proyecto realizada por parte de los estudiantes con asesoría del docente, es decir en la intervención (social, empresarial), o construcción del modelo propuesto según el tipo de proyecto, es la fase de mayor duración que implica el desempeño de las competencias genéricas y específicas a desarrollar.
- **Evaluación:** es la fase final que aplica un juicio de valor en el contexto laboral-profesión, social e investigativo, ésta se debe realizar a través del reconocimiento de logros y aspectos a mejorar se estará promoviendo el concepto de “evaluación para la mejora continua”, la metacognición, el desarrollo del pensamiento crítico y reflexivo en los estudiantes.

## 10. Evaluación por competencias

Son las técnicas, instrumentos y herramientas sugeridas para constatar los desempeños académicos de las actividades de aprendizaje.

## 11. Fuentes de información

1. Tanenbaum, Andrew S., Redes de Computadoras, Cuarta Edición, Pearson/Prentice-Hall, México, 2003, ISBN: 9702601622.
2. CISCO Systems, Guía del Primer año CCNA 1 y 2, Academia de Networking de Cisco Systems, Tercera edición, Pearson/Cisco Press, 2004, ISBN: 842054079X.
3. CISCO Systems, Guía del Segundo año CCNA 3 y 4, Academia de Networking de Cisco Systems, Tercera edición, Pearson/Cisco Press, 2004, ISBN: 842054079X.
4. Altamirano, Carlos A. Vicente; Julio de 2003, Un modelo funcional para la administración de redes, UNAM-DGSCA, Disponible vía web en: [http://teclapaz.iespana.es/materias\\_archivos/\(ADR\)%20Modelo-Funcionalarticulo.doc](http://teclapaz.iespana.es/materias_archivos/(ADR)%20Modelo-Funcionalarticulo.doc).
5. Andrew Lockhart, Network Security Hacks, O'Reilly, 2006, ISBN: 978- 0596527631.
6. Ross J. Anderson, Security Engineering, Wiley, 2008, ISBN: 978- 0470068526
7. Rob Flickenger, Linux Server Hacks, O'Reilly, 2003, ISBN: 0596004613
8. William Hagen, Brian Jones Linux Server Hacks Volume Two, O'Reilly, 2005, ISBN: 0596100825
9. Eric T. Peterson, Web Site Measurement Hacks, O'Reilly, 2005, ISBN: 0596009887.
10. Steve Maxwell, RedHat Linux, Herramientas para la administración de redes, Mc Graw Hill 2001, ISBN: 9789584102201.
11. Richard Bejtlich, El tao de la monitorización, Pearson, ISBN:9788420546001.