



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

CARRERA	PLAN DE ESTUDIO	CLAVE ASIGNATURA	NOMBRE DE LA ASIGNATURA
IC y LSC	2003-1	5038	Programación Orientada a Objetos II

PRÁCTICA No.	LABORATORIO DE		DURACIÓN (HORA)
3	NOMBRE DE LA PRÁCTICA	Sockets	4

1 INTRODUCCIÓN

Un socket de Internet, socket de red o simplemente socket, es una herramienta para permitir la comunicación bidireccional entre dos computadoras conectadas en una red basada en IP. En este contexto, el socket es una abstracción de software en la cual se combina una dirección IP con un número de puerto en una sola entidad. Así, el socket se convierte en una interfaz entre un proceso o hilo de una aplicación y la pila de protocolos IP proporcionada por el sistema operativo, siendo el primer paso en el establecimiento del flujo de datos entre una aplicación con otro proceso o servicio.

En Java, las clases de sockets se utilizan para representar la conexión entre aplicaciones funcionando bajo una arquitectura cliente-servidor. El paquete java.net proporciona dos clases – Socket y ServerSocket – que implementan el lado del cliente y el lado del servidor en la conexión, respectivamente.

2 OBJETIVO (COMPETENCIA)

Crear una aplicación que utilice un servicio de la red para obtener información y representarla de una manera adecuada.

Formuló M.C. Jorge Eduardo Ibarra Esquer	Revisó M.C. Gloria Etelbina Chavez Valenzuela y LSC Mónica Lam Mora	Aprobó	Autorizó M.C. Miguel Ángel Martínez Romero
Maestro	Coordinador de carrera	Gestión de la Calidad	Director de la Facultad



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

3 FUNDAMENTO

Un socket de Internet es identificado por el sistema operativo como una combinación de los siguientes elementos:

- Protocolo (TCP, UDP, etc.)
- Dirección IP local
- Número de puerto local

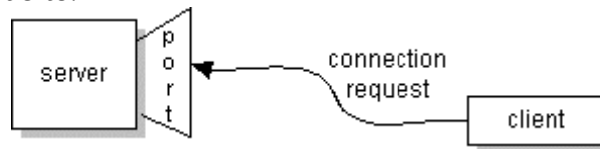
Cuando el socket ya se ha establecido, también se incluyen:

- Dirección IP remota
- Número de puerto remoto

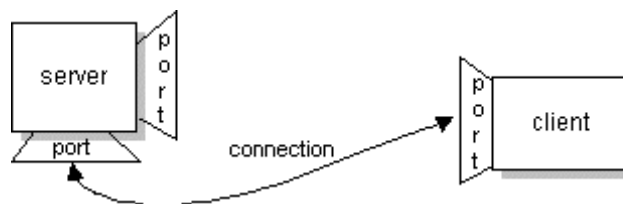
El sistema operativo se encarga de redirigir la información recibida a través del socket hacia el proceso correspondiente.

Por medio de sockets, podemos escribir aplicaciones basadas en el modelo cliente-servidor. Normalmente, un servidor se ejecuta en una computadora específica y tiene un socket vinculado a un puerto específico. El servidor simplemente espera, permitiendo que el socket “escuche” las solicitudes de conexión de los clientes.

En el lado del cliente, éste conoce la dirección de la computadora en que se ejecuta el servidor y el número del puerto al que se encuentra conectado. Para solicitar una conexión, el cliente trata de encontrarse con el servidor en su computadora y puerto.



Si todo funciona correctamente, el servidor acepta la conexión. Una vez aceptada, éste abre un nuevo socket en un puerto diferente. Se necesita un nuevo socket (y, en consecuencia, un número de puerto distinto) de manera que el socket original pueda seguir recibiendo solicitudes de conexión de los clientes.



En el cliente, si se acepta la conexión, se crea un socket que puede utilizarse para comunicarse con el servidor. A este socket se le asigna un puerto local en la computadora en la que se ejecuta el cliente, el cual no necesariamente es el mismo número de puerto que el del socket del servidor.

En este momento, el cliente y el servidor pueden comunicarse escribiendo y leyendo a través de sus sockets.



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD

Formato para prácticas de laboratorio

La edición estándar de Java proporciona algunas clases para trabajar con sockets. Estas clases se encuentran en el paquete `java.net`. La clase `java.net.Socket` nos permite establecer la comunicación entre dos aplicaciones a través de un socket.

Podemos crear sockets cliente utilizando los diversos constructores de esta clase. Uno de ellos, recibe como parámetros el nombre del servidor y el puerto donde se desea abrir la conexión. Su declaración es la siguiente:

```
public Socket(String host, int port);
```

Un ejemplo de su uso es el siguiente:

```
Socket s;  
s=new Socket("yaqui.mx1.uabc.mx", 80);
```

En este caso, al invocar al constructor de la clase, se abrirá una conexión al puerto 80 de la computadora denominada "yaqui.mx1.uabc.mx". Este primer parámetro puede ser también la dirección IP de la computadora.

Al intentar abrir un socket se pueden presentar diversas excepciones. Para el constructor del ejemplo, si no se puede determinar la dirección de destino, se arrojará una excepción del tipo `UnknownHostException`. Además, si por alguna razón la conexión no pudiera establecerse, se arrojará una excepción del tipo `IOException`.

La comunicación a través de sockets se realiza a bajo nivel. Para que las aplicaciones interactúen de manera adecuada, ambas deben conocer los detalles del protocolo que se está utilizando. De esta forma, el servidor puede entender las solicitudes del cliente y responder de manera adecuada; y el cliente puede entender la información enviada por el servidor para poder representarla de la manera que más le convenga.

Una vez abierto el socket, la comunicación se realiza a través de los flujos de entrada y salida asociados al mismo. Estos pueden obtenerse de la siguiente forma:

```
BufferedReader entrada=new BufferedReader(new  
InputStreamReader(s.getInputStream()));  
  
PrintWriter salida=new PrintWriter(s.getOutputStream());
```

Las clases `BufferedReader` y `PrintWriter` se encuentran en el paquete `java.io`. De la manera ejemplificada, se obtienen objetos a partir de los cuales podemos leer y escribir flujos de caracteres. En caso de flujos de bytes, se pueden utilizar algunas de las otras clases del paquete `java.io`.

Un protocolo sencillo de utilizar es el Finger. Este protocolo nos proporciona información acerca de los usuarios de un sistema. Una sesión de finger es como sigue:



UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD

Formato para prácticas de laboratorio

Servidor: Espera conexiones en el puerto 79.

Cliente: Abre conexión al puerto 79 del servidor.

Cliente: Envía comando al servidor.

Servidor: Envía respuesta.

Servidor: Cierra la conexión.

Cliente: Cierra la conexión.

El comando puede ser un nombre de usuario, una parte del nombre real de algún o algunos usuarios o un comando nulo (caracter de salto de línea). En los primeros dos casos, la respuesta será la información completa de todos los usuarios que cumplan con el criterio de búsqueda. En el último caso, la respuesta serán los datos de los usuarios con una sesión abierta en ese momento.

4 PROCEDIMIENTO (DESCRIPCIÓN)

A**EQUIPO NECESARIO****MATERIAL DE APOYO**

Computadoras con una versión reciente del Java
Development Kit JDK y acceso a Internet.



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

B**DESARROLLO DE LA PRÁCTICA**

1. Escribir una aplicación en Java que funcione como un cliente para el protocolo finger.
 - Utilizar para la conexión el servidor `tiburón.mxl.uabc.mx`.
 - La aplicación se conectará al servidor a través de un socket. El servidor finger espera conexiones en el puerto 79.
 - Escribir en una clase en los métodos necesarios para la conexión e interacción con el servidor.
 - Probar la aplicación enviando los criterios de búsqueda desde la línea de comando. Mostrar la respuesta del servidor en la consola.
 - Al terminar de ejecutarse la aplicación, deberán cerrarse la sesión con el servidor y el socket.
2. Programar una aplicación gráfica con la cual se pueda interactuar con el servidor finger. Utilizar la clase programada en la parte anterior de la práctica para el funcionamiento del cliente.
 - La entrada de los criterios de búsqueda será por medio de un campo de texto.
 - La respuesta se presentará en un área de texto. Agrega barras de desplazamiento para poder ver la respuesta completa, en caso de que ésta sea muy extensa.
 - Se deberá separar la lógica de la aplicación de la presentación. Es decir, se tendrá una clase con los métodos y atributos necesarios para el funcionamiento del cliente, y otra(s) para la interfaz gráfica.

C**CÁLCULOS Y REPORTE**

Se verificará el funcionamiento adecuado de la aplicación y se solicitará un reporte con el código de la misma.

5 RESULTADOS Y CONCLUSIONES

El alumno será capaz de elaborar aplicaciones que hagan uso de diversos servicios de red.



**UNIVERSIDAD AUTÓNOMA DE BAJA CALIFORNIA
FACULTAD DE INGENIERÍA (UNIDAD MEXICALI)
DOCUMENTO DEL SISTEMA DE CALIDAD**

Formato para prácticas de laboratorio

6 ANEXOS