



Número de Práctica: 1

Nombre de la Práctica: Comunicación Cliente – Servidor Vía Sockets Java con Sistema Manejador de Bases de Datos

Problemática:

Desarrollar aplicaciones Cliente – Servidor utilizando Sockets y Base de Datos relacionales

Elemento de Competencia:

Utiliza Sistemas Manejadores de Bases de Datos para manipular Bases de Datos y dar soporte persistente a los datos procesados por la aplicación a realizar, así mismo Sockets Java para mantener comunicación Cliente – Servidor entre la Aplicación a desarrollar y el sistema Manejador de Bases de Datos.

Requerimientos:

Hardware:

- Laboratorio de cómputo
- Red computacional
- Proyector

Software:

- Sistema operativo
- Sistema Manejador de Bases de Datos (MySQL, SQL Server, ORACLE, DB2, etc.)
- Java Development Kit (JSDK 7.x)

Actividades Previas:

El alumno deberá presentarse en el laboratorio con el diseño lógico de una Base de Datos Relacional, con las características solicitadas por el docente titular de la Unidad de Aprendizaje.



Fundamentos Teóricos:

- ✓ Bases de Datos.- Instrucciones básicas de SQL en su forma mas simple para interactuar con el manejador MySql

Instrucción	Detalle
create database nombreBaseDatos;	Comando para crear una base de datos.
use database nombreBaseDatos;	Le indica a MySQL que use la base de datos nombreBaseDatos como la base de datos por defecto para los comandos siguientes.
CREATE TABLE [IF NOT EXISTS] <i>tbl_name</i>	crea una tabla con el nombre dado.
SELECT select_expr, ...[FROM table_references] [WHERE where_definition]	<p>se usa para recibir registros seleccionados desde una o más tablas.</p> <p>Cada select_expr indicata una columna que quiere recibir.</p> <p>table_references indicata la tabla o tablas desde la que recibir registros</p> <p>where_definition consiste en la palabra clave WHERE seguida por una expresión que indica la condición o condiciones que deben satisfacer los registros para ser seleccionados.</p>
INSERT [INTO] <i>tbl_name</i> [(col_name,...)] VALUES { VALUE}	Inserta una nueva fila en una tabla existente.
UPDATE <i>tbl_name</i> SET col_name1=expr1 [, col_name2=expr2 ...] [WHERE where_definition]	<p>El comando UPDATE actualiza columnas en registros de tabla existentes con nuevos valores.</p> <p>La cláusula SET indica qué columna modificar y los valores que puede recibir.</p> <p>La cláusula WHERE , si se da, especifica qué registros deben actualizarse. De otro modo, se actualizan todos los registros.</p>
DELETE FROM <i>tbl_name</i> [WHERE where_definition]	<p>Borra los registros de <i>tbl_name</i> que satisfacen la condición dada por where_definition, y retorna el número de registros borrados.</p> <p>Si realiza un comando DELETE sin cláusula WHERE se borran todos los registros</p>



- ✓ Sockets Java
 - Servidor

```
public class ServidorHora {  
    public static final int puerto = 5000;  
  
    private static String mostarHora() {  
        GregorianCalendar calendario = new GregorianCalendar();  
        Date fecha = calendario.getTime();  
        DateFormat hora = DateFormat.getInstance(DateFormat.SHORT);  
        return hora.format(fecha);  
    }  
  
    public static void main(String[] args) throws IOException {  
        ServerSocket socketServidor = new ServerSocket(puerto);  
        while (true) {  
            Socket socketCliente = socketServidor.accept();  
            PrintWriter pw = new PrintWriter(socketCliente.getOutputStream());  
            pw.println(mostarHora());  
            pw.close();  
            socketCliente.close();  
        }  
    }  
}
```

Puerto Escucha

Bind, Listen

Espera peticiones

- Cliente

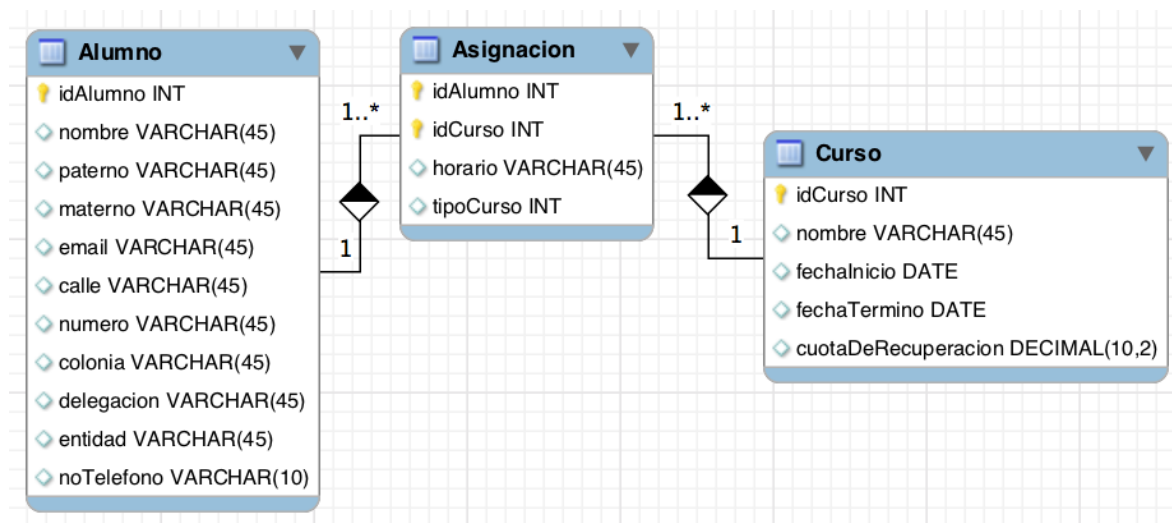
- Crea un nuevo socket en el mismo puerto que esta escuchando el servidor.

```
public class ClienteHora {  
  
    public static void main(String[] args) throws  
        UnknownHostException, IOException {  
  
        Socket socket = new Socket("127.0.0.1", ServidorHora.puerto);  
        InputStream is = socket.getInputStream();  
        BufferedReader br = new BufferedReader(new InputStreamReader(is));  
        System.out.println(br.readLine());  
        br.close();  
        socket.close();  
    }  
}
```



Instrucciones:

1. Cree la siguiente base de datos.



2. Crear una interfaz Gráfica en Java Swing para dar soporte a las diferentes actividades de Administracion en cada una de las entidades del punto Anterior, esta interfaz gráfica sera el cliente, que realizará peticiones al Servidor.
3. Crear un SocketServidor que permita mantener comunicación con el servidor de bases de datos MySql y permita dar soporte a las operaciones CRUD Basicas para cada una de las Entidades mostradas en la imagen del punto 1.