

TECNOLÓGICO NACIONAL DE MÉXICO	MANUAL DE PRÁCTICAS
INSTITUTO TECNOLÓGICO DE OAXACA	TÓPICOS AVANZADOS DE PROGRAMACIÓN
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN	DR. ROGELIO LIMÓN CORDERO (2018)

Práctica 8. Crear conexiones para base de datos y acceder a ellos.

Competencia a desarrollar

Establece conexiones a diferentes orígenes de datos para su manipulación y visualización de información.

Introducción

Para desarrollar en los estudiantes la capacidad de establecer conexiones con orígenes de datos, se tiene que realizar todo un proceso de preparación del servidor y conformación de la base de datos. Partiendo del hecho que se dispone de un servidor de base de datos y sus conectores correspondientes y su configuración apropiada.

Por lo tanto, esta práctica abarca todo el proceso necesario para conformar la base de datos y el software para manejarla. Luego se realizan las tareas que implican llevar a cabo la conexión a la base de datos y el acceso a ella para su manipulación, aunque aquí solo se lleguen a visualizar los datos de las tablas.

Correlación con los temas y aplicación en el contexto

Aquí se aborda de lleno el tema 4.2 que se refiere a la conexión de datos y se inicia con el tema 4.3 con inserción y recuperación de datos en memoria, del tema 4.4 solo se hace la visualización en consola.

En cuanto al contexto donde se aplica es con conectores de base de datos para java (JDBC) dentro del manejador nativo de java(Derby) utilizando un programa java para crear la conexión y acceder a los datos.

Material y equipo necesario

Equipo de cómputo: Laptop o PC

Software: java con un jdk versión 1.7.0 o posterior, verificar que cuente con el Derby.

IDE NetBeans, verificar que cuente en sus bibliotecas (Libraries) el conector de java.

Metodología

Primero se presentan en forma sucinta los pasos para establecer una conexión a nivel de programación con java, luego se exponen las entidades de información para la base de datos que debe crearse a la cual se tiene que conectar y finalmente durante el desarrollo de va implementando cada paso. Para hacer que el estudiante se involucre efectivamente durante el desarrollo se le pide que complemente ciertas partes del código, responda a unas preguntas, cuyas respuestas deberá anotar en su bitácora de seguimiento entre otras cosas.

Pasos para establecer una conexión con una base de datos utilizando JDBC mediante programación.

- 1) Contar con un servidor de base datos que permita crear y manejar la base de datos.

- 2) Disponer del conector JDBC descargado del sitio correspondiente del manejador de base de datos y hacer que este pueda ser accedido desde cualquier ruta, para el caso Windows se agrega la dirección del conector en la variable de entorno CLASSPATH.
- 3) Crear la base de datos dentro del manejador de base datos
- 4) Crear un proyecto simple de java:
 - i) Dentro de la carpeta de biblioteca (library) del proyecto agregar el “driver”.jar correspondiente al manejador de base de datos (nativo de java, MySQL, PostgreSQL u otro).
 - ii) Dentro de un paquete fuente crear el programa java que cargue el drive apropiado y permita crear la conexión utilizando clases contenidas en el paquete java.sql.
 - iii) Crear otra clase java que realice la conexión utilizando el programa anterior y acceda a los datos para su visualización y/o manipulación.

Descripción de la base de datos

La base de datos propuesta (puede ser cualquier otra propuesta por el catedrático) contiene información sobre mediciones antropométricas y datos de personas para poder calcular indicadores de salud básicos como los vistos en las practicas 1 y 3, la cual está compuesta de tres tipos de entidades de información como se aprecia en la descripción de abajo.

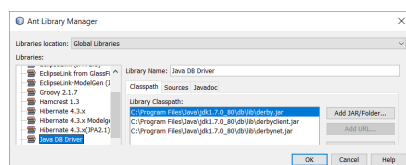
Persona	Mediciones	TipoActividad
idPersona (número entero) Nombre (texto) Fecha de nacimiento (fecha) Sexo (un carácter :‘H’ o‘M’)	idMedidas (número entero) Fecha medición (Fecha) Estatura (número entero, se mide en cms.) Peso (número decimal, se mide en kgs.) Cintura (número entero, se mide en cms) Cadera (número entero, se mide en cms)	idTipoAct (número entero) Nombre (texto corto) Descripción (texto de hazta 50 caracteres)

Desarrollo

1. Creación de la base de datos con el manejador de BD nativo de java
 - 1.1. Preparativos:
 - a) Verificar que se cuente con el manejador (derby) donde está instalado java, por ejemplo si está en: C:\Program Files\Java\jdk1.8.0_111\ debe haber una carpeta ‘db’ y dentro de ella está la carpeta ‘lib’ que contiene entre otros archivos al servidor y al conector:

jrama > Java > jdk1.8.0_111 > db > lib			
Nombre	Fecha de modifica...	Tipo	Tamaño
derby	08/03/2017 08:03 ...	Executable Jar File	3,033 KB
derby.war	08/03/2017 08:03 ...	Archivo WAR	2 KB
derbyclient	08/03/2017 08:03 ...	Executable Jar File	572 KB

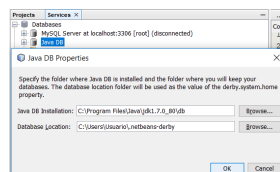
- b) En caso de usar la biblioteca de NetBeans, asegurarse que se cuente con los elementos referidos. Dentro de la barra del menú, en ‘Herramientas’, seleccione a ‘Biblioteca’:



Nota: Puede usarse el servidor y conector desde la biblioteca genérica de NetBeans o bien directamente desde donde está java.

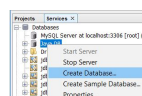
1.2. Especificación del servidor y ruta donde estará la BD.

Ubicarse en la ventana de ‘Prestaciones o Services’, activada mediante: **ctrl+5**. Ubicar el cursor en **JavaDB**, dar clic derecho, selecciona propiedades y elige el servidor, ya sea el propuesto por en NetBeans o bien elegir la versión requerida localizándola al dar clic en [Browse], lo mismo para ubicar la base de datos a crear, para aceptar hacer clic en [OK].

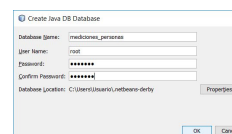
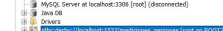


1.3. Creación del esquema de la base de datos

Ubicarse en **Services, Java DB**; clic derecho y seleccionar **Create Database**, =>



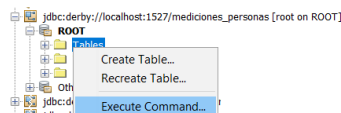
Especificar los datos de la base de datos: nombre, usuario y password, para terminar dar clic en [OK], => debiendo aparecer =>



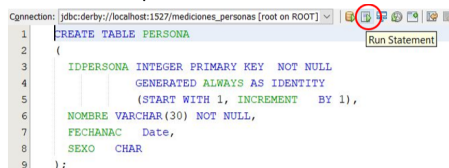
Esto indica que el esquema de la BD está creado, falta ahora crear las tablas dentro de él.

1.4. Creación de las tablas dentro del asistente de NetBeans.

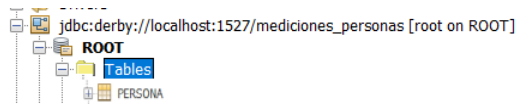
Ubicarse en el esquema de la base de datos, expandir el nodo llamado **ROOT** (o como le hayas llamado), seleccionar ‘Tables’, dar clic derecho y seleccionar ‘Execute Command..’



En la ventana que aparece a la derecha del esquema transcribir el código enseguida mostrado correspondiente a la creación de la tabla 'Persona':



Al terminar la transcripción, activar el botón marcado entre un círculo rojo, que está en la parte superior derecha de la ventana, el cual ejecuta la orden del 'Sql'. La tabla creada debe mostrarse en la carpeta 'Tables' del esquema (llamado ROOT).



Continúa ahora con la creación de la tabla TIPOACTIVIDAD, completa la especificación de los campos 'NOMBRE' y 'DESCRIPCION'

```

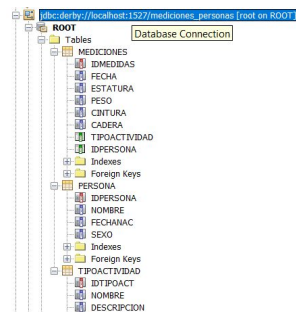
//Complementa los campos de la tabla TIPOACTIVIDAD
CREATE TABLE TIPOACTIVIDAD
(IDTIPOACT INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),
NOMBRE // texto ,
DESCRIPCION // texto,
);
```

Por último crea la tabla MEDICIONES, completando la última referencia indicada en el esquema

```

CREATE TABLE MEDICIONES
(
IDMEDIDAS INTEGER PRIMARY KEY NOT NULL
GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),
FECHA DATE,
ESTATURA INT,
PESO DOUBLE,
CINTURA INT,
CADERA INT,
IDTIPOACT INT REFERENCES TIPOACTIVIDAD(IDTIPOACT),
IDPERSONA INT REFERENCES //Completa la especificacion
);
```

Al final la base de datos debe estar creada como aparece abajo



2. Creación de la conexión

- 2.1. Crea un proyecto simple java, puedes llamarlo 'practica_08' con un paquete fuente del mismo nombre, dentro del paquete crea una clase java, llámala 'Conexion'
- 2.2. Especificación del código de la clase conexión. La estrategia aquí es crear una sola instancia de ésta, independientemente del número de veces que se trate de crear otras, esto se logra creando un constructor privado para controlar internamente la creación de instancias y así, si se vuelve a tratar de crear otra instancia se regrese la misma conexión creada al inicio
- 2.3. Atributos de la clase

```
private static Connection coneccion; //contenida en el paquete sql
private static Conexion coneccion; // instancia a utilizar
private static int numConexiones=0; //controla el numero de veces que se accedió
```

- 2.4. Constructor. Utiliza como parámetros: url, usuario y password de la base de datos.

```
private Conexion(String url,String usuario,String password){
    try{
        //Clase usada para una conexion con Derby
        Class.forName("org.apache.derby.jdbc.ClientDriver");
        // para MySql : "com.mysql.jdbc.Driver"
        // Investiga cual es para postgresql
        try{
            coneccion = (Connection) DriverManager.getConnection(url,usuario,password);
        } catch (SQLException ex) {
            Logger.getLogger(Conexion.class.getName()).log(Level.SEVERE, null, ex);
        }
    } catch (ClassNotFoundException ex) {
        Logger.getLogger(Conexion.class.getName()).log(Level.SEVERE, null, ex);
    }
}
```

¿Por qué crees que este constructor es privado?

- 2.5. Método para crear la conexión

```
public static Conexion getConexion(String url,String usuario,String password){
    numConexiones++;
    if(coneccion == null)
    {
        coneccion = new Conexion(url,usuario,password);
    }
    return coneccion;
}
```

TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

MANUAL DE PRÁCTICAS
TÓPICOS AVANZADOS DE PROGRAMACIÓN
DR. ROGELIO LIMÓN CORDERO (2018)

2.6. Haz un seguimiento manual del método anterior y ve explicando cómo funciona, anótalo en la bitácora de seguimiento.

2.7. Método que regresa la conexión creada

```
public static Connection getConeccion()
{
    return coneccion;
}
```

2.8. Método para cerrar la conexión

```
public boolean cerrarConexion()
{
    try{
        if(coneccion!=null)
        {
            if(numConexiones>1)
            {
                coneccion.close();
                return true;
            }
            else
            {
                numConexiones--;
                return false;
            }
        }
        catch(SQLException e)
        {
            System.err.println("Error al tratar de cerrar la conexion "+e);
        }
        return false;
    }
}
```

Haz un seguimiento manual del método anterior y ve explicando cómo funciona, anótalo en la bitácora de seguimiento.

3. Creación de la clase de manejo de datos

Esta clase llama a la conexión para acceder a los datos.

3.1. Crea una clase java llamada 'ManejoDatos' y agrégale los siguientes atributos:

```
private Connection coneccion;//acceso a conexion
private Conexion crearConexion;//Crea conexion
private final int CAMPOS_PERSONA=4;
```

3.2. Constructor. Aquí se crea la conexión y se obtiene un objeto de esta para el manejo de los datos

```
public ManejoDatos() //
{
    crearConexion = crearConexion.getConeccion("jdbc:derby://localhost:1527/mediciones_personas","root","rogelio");
    coneccion = crearConexion.getConeccion();
}
```

3.3. Consulta de persona. Aquí se hace la consulta a la tabla persona y se regresa en una lista

```

public List<Object []> conexionConsultaPersona(String sql)
{
    //Regresa los registros de las personas en una lista
    List<Object []> datos= new ArrayList<Object []>();
    DateFormat fecha = new SimpleDateFormat("yyyy-MM-dd");
    try{
        Statement ps = conexion.createStatement();
        ResultSet rs = ps.executeQuery(sql);
        while(rs.next())
        {
            String dat[]= new String[CAMPOS_PERSONA];
            //Estructura del registro persona pasado como cadena
            dat[0]= String.valueOf(rs.getInt(1));
            dat[1]= rs.getString(2);
            dat[2]= fecha.format((Date)rs.getDate(3));
            dat[3]= //Complementa lo requerido
            datos.add(dat);
        }
    } catch (Exception e)
    {
        System.err.println("Error al conexion consultar persona "+e);
    }
    return datos;
}

```

Complementa lo que debe asignarse al dat[3]

- 3.4. Consulta de tipo de actividad. Aquí se hace la consulta a la tabla actividad y el resultado se regresa en una lista. Complementa lo que hace falta y crea el dato
CAMPOS_ACTIVIDAD

```

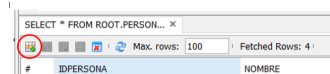
public List<Object []> conexionConsultaActividad(String sql)
{
    //Regresa los registros de tipo de actividad
    List<Object []> datos= new ArrayList<Object []>();
    try{
        Statement ps = conexion.createStatement();
        ResultSet rs = ps.executeQuery(sql);
        while(rs.next())
        {
            String dat[]= new String[CAMPOS_ACTIVIDAD];
            //Estructura del registro actividad
            dat[0]= // ?
            dat[1]= // ?
            dat[2]= // ?
            datos.add(dat);
        }
    } catch (Exception e)
    {
        System.err.println("Error en conexion consultar actividad "+e);
    }
    return datos;
}
}

```

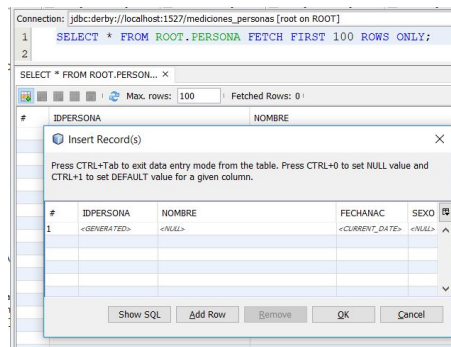
4. Inserción de datos para prueba

Inserción de datos con el asistente.

- 4.1. Ubicarse en la ventana de 'Service', seleccionar la base de datos: jdbc:derby://localhost:1527/mediciones_personas [root on ROOT]. Ampliar el contenido dando clic en [+]. Seleccionar el esquema ROOT, ampliar su contenido.
- 4.2. Seleccionar 'Tables', seleccionar la tabla 'Persona' clic derecho, selecciona 'View Data', dar clic en el icono marcado con el círculo rojo.



- 4.3. En la ventana que aparece agrega datos de personas



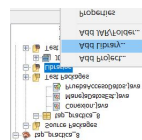
5. Creación de la clase prueba

Crea una clase java como la mostrada abajo

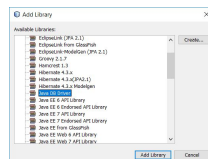
```
public class PruebaAccesoDatos {
    public static void main(String arg[])
    {
        String consulta = "SELECT * FROM ROOT.PERSONA";
        ManejoDatos baseDatos = new ManejoDatos(); // atributo de la clase ManejoDatos
        ArrayList<Object[]> actividad = (ArrayList<Object[]>) baseDatos.conexionConsultaPersona(consulta);
        for(int ne=0; ne< actividad.size(); ne++)
        {
            Object reg[] = actividad.get(ne);
            System.out.println();
            for(int c=0; c< reg.length; c++)
            {
                System.out.printf("%-25s", reg[c]);
            }
        }
    }
}
```

6. Incorporación del conector e inicio del servidor

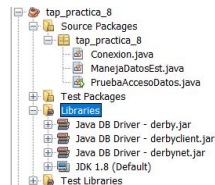
- 6.1. Dentro del proyecto creado en esta práctica, ubicar el cursor en la carpeta 'Libraries' (bibliotecas), dar clic derecho y del menú mostrado selecciona y activa la opción 'add Library'



En la ventana mostrada selecciona la biblioteca Java DB Driver y da clic en el botón [add Library]



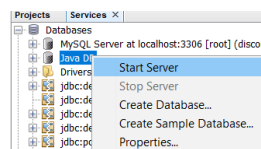
Al terminar esta acción en la carpeta 'Libraries' deben mostrarse las bibliotecas incorporadas



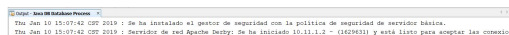
El proyecto está listo para reaccionar a una conexión con la base de datos , pero....aún hacer falta algo...

6.2. Activar el servidor.

Ubicarse en la ventana 'Services' y en JavaDB, dar clic derecho, activando 'Start'

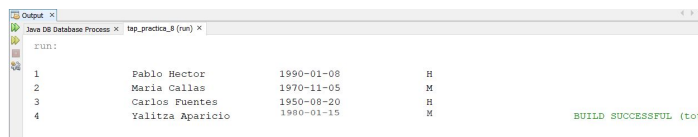


En la ventana de salida se muestran los mensajes de que el servidor está activo



7. Ejecución de la clase prueba.

7.1. Al ejecutar la clase prueba se deben mostrar los datos de las personas que hayas insertado, como por ejemplo:



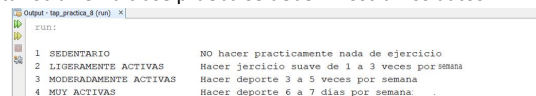
7.2. Mostrar los datos de tipos de actividad.

a) Insertar los siguientes datos en la tabla 'TIPOACTIVIDAD' usando el asistente de base de datos de NetBeans

#	IDTIPOACT	NOMBRE	DESCRIPCION
1	1	SEDENTARIO	NO hacer practicamente nada de ejercicio
2	2	LIGERAMENTE ACTIVAS	Hacer ejercicio suave de 1 a 3 veces por semana
3	3	MODERADAMENTE ACTIVAS	Hacer deporte 3 a 5 veces por semana
4	4	MUY ACTIVAS	Hacer deporte 6 a 7 dias por semana

b) Modificar la clase prueba para obtener los datos de la tabla referida

Al ejecutar otra vez la clase prueba se deben mostrar los datos:



TECNOLÓGICO NACIONAL DE MÉXICO
INSTITUTO TECNOLÓGICO DE OAXACA
DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

MANUAL DE PRÁCTICAS
TÓPICOS AVANZADOS DE PROGRAMACIÓN
DR. ROGELIO LIMÓN CORDERO (2018)

8. Como percibes, en el constructor de la clase 'ManejoDatos' hay una vulnerabilidad de seguridad, pues se proporciona el password, ¿Cómo podrías evitarlo?, realiza el código necesario para esto y anótalo en tu bitácora de seguimiento.

Reporte del estudiante

Entrega del código de la aplicación y la base de datos implementada

Entrega de imágenes de pruebas de ejecución

Entrega de la bitácora de seguimiento

Bibliografía

1. <https://netbeans.org/kb/docs/ide/mysql.html#configuring>, Último acceso 10/01/19
2. <https://docs.oracle.com/javase/tutorial/index.html> Último acceso 10/01/19
3. https://www.tamps.cinvestav.mx/~visosa/clases/bd/Uso_JavaJDBC.pdf Último acceso 10/01/19
4. Deitel y Deitel. Como programar en Java. Prentice Hall. Septima Edición, 2008