TECNOLÓGICO NACIONAL DE MÉXICO

INSTITUTO TECNOLÓGICO DE OAXACA

TÓPICOS AVANZADOS DE PROGRAMACIÓN

Practica4-2Conexión a base de datos con Derby

PRESENTA ESTUDIANTE DE LA CARRERA INGENIERÍA EN SISTEMAS COMPUTACIONALES:

ZARATE CARREÑO JOSÉ VALENTÍN

DOCENTE: HERNANDEZ ABREGO ANAYANSI

GRUPO: 4SA **HORA: 9:00- 10:00**

Oaxaca de Juárez, Oax, 24 de marzo de 2020.

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.

MATERIAL Y EQUIPO NECESARIO

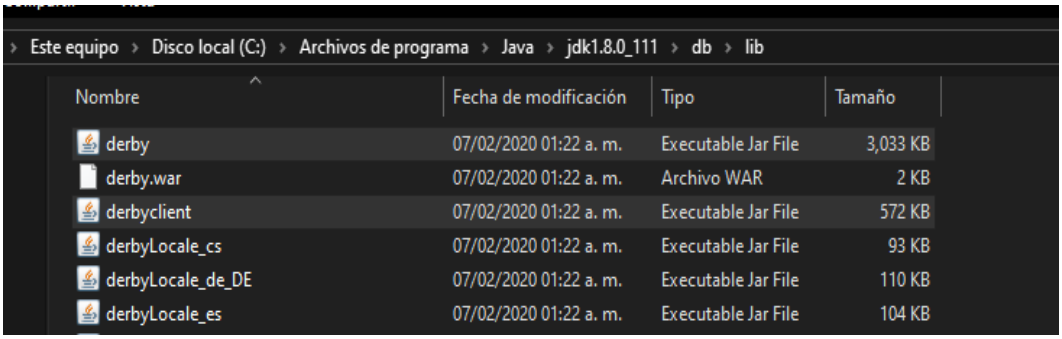
- Equipo de cómputo: Laptop o PC
- Software: java con un jdk versión 1.7.0 o posterior, verificar que cuente con el Derby. IDE NetBeans, verificar que cuente en sus bibliotecas (Libraries) el conector de java.

METODOLOGÍA

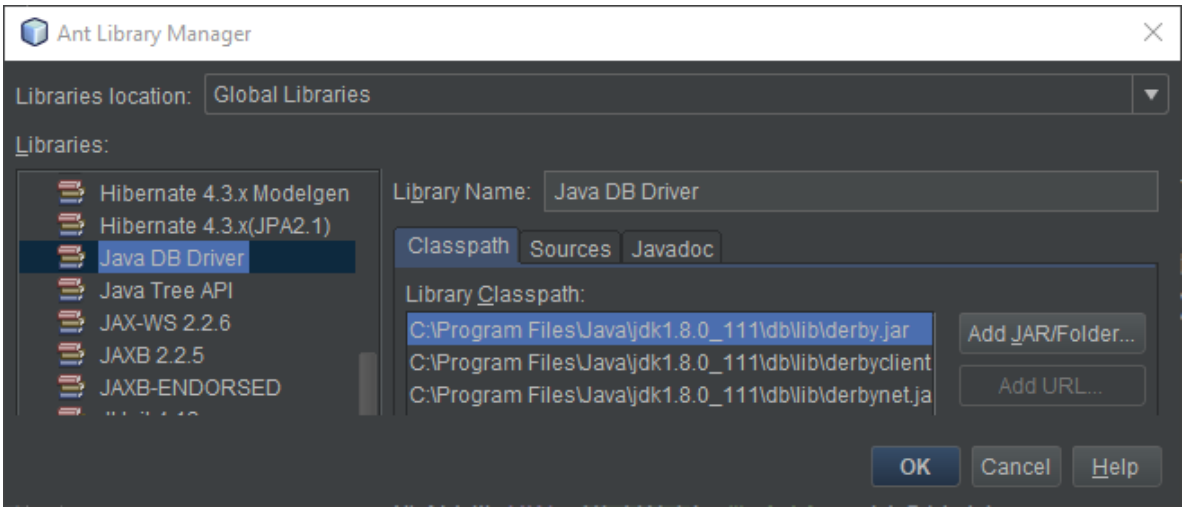
Primero se presentan en forma sucinta los pasos para establecer una conexión a nivel de programación con java, luego se exponen las entidades de información para la base de datos que debe crearse a la cual se tiene que conectar y finalmente durante el desarrollo de va implementando cada paso. Para hacer 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.

DESARROLLO

1. Creación de la base de datos con el manejador de BD nativo de java
    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

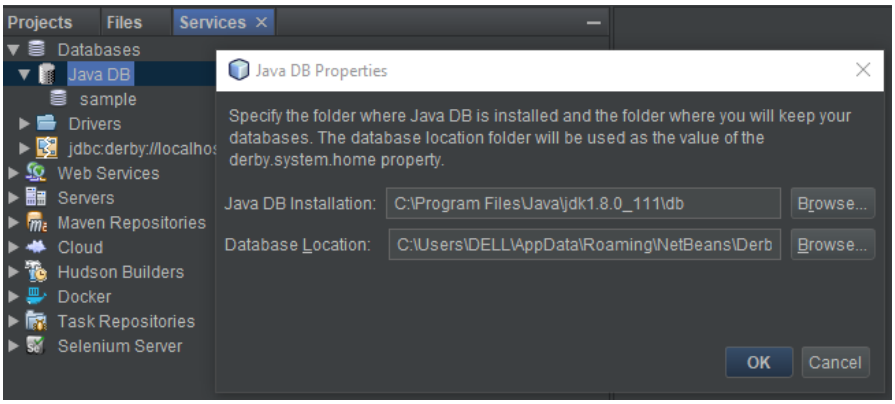


    b) En caso de usar la biblioteca de NetBeans, asegurarse que se cuente con los elementos referidos. Dentro de la barra del menú, en ‘Herramientas’, seleccione a ‘Biblioteca’:

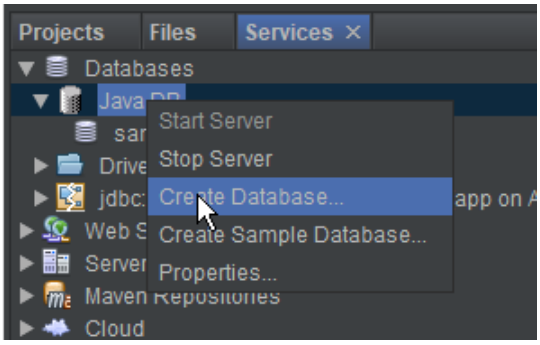


Nota: Puede usarse el servidor y conector desde la biblioteca genérica de NetBeans o bien directamente desde donde está java.

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

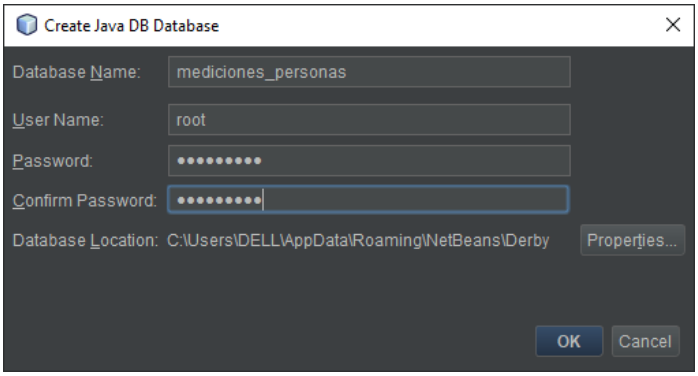


Ubicarse en la ventana de 'Prestaciones o Services', activada mediante: ctrl+5. Ubicar el cursor en JavaDB, dar clic derecho, selecciona propiedades y elige el servidor, ya sea el propuesto por en NetBeans o bien elegir la versión requerida localizándola al dar clic en [Browse], lo mismo para ubicar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

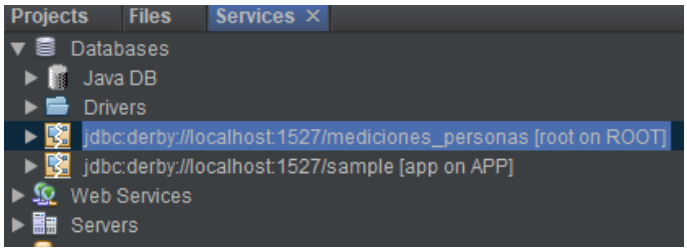


1.3. Creación del esquema de la base de datos Ubicarse en Services,Java DB; clic derecho y seleccionar Créate Database, =>

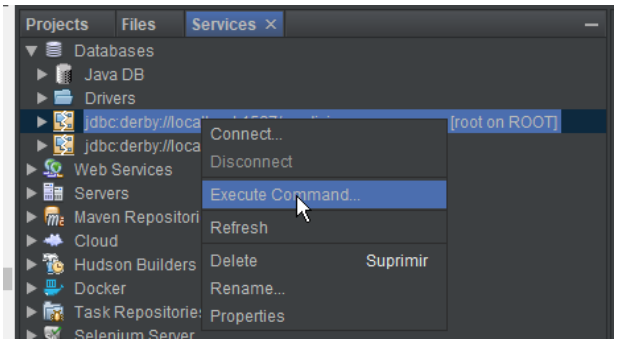
Especificar los datos de la base de datos: nombre,usuario y password,para terminar dar clic en [OK],=>



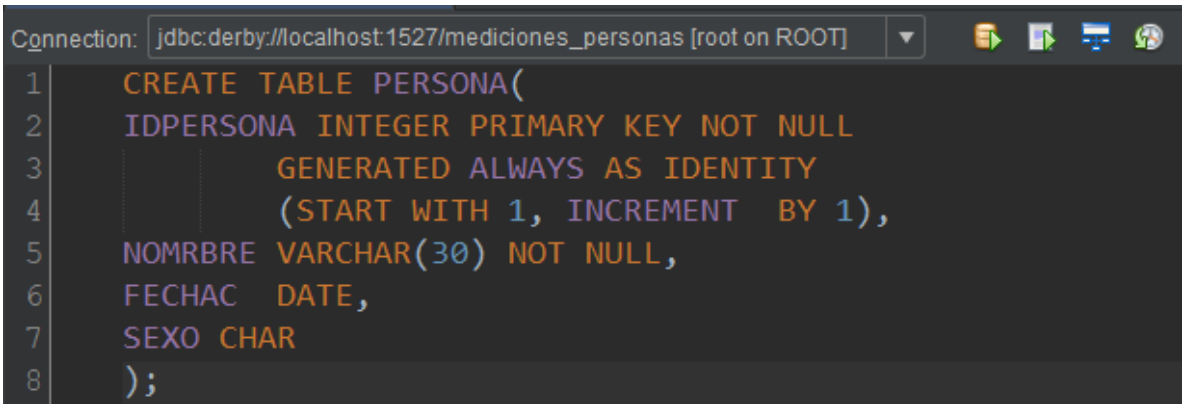
Debiendo aparecer:



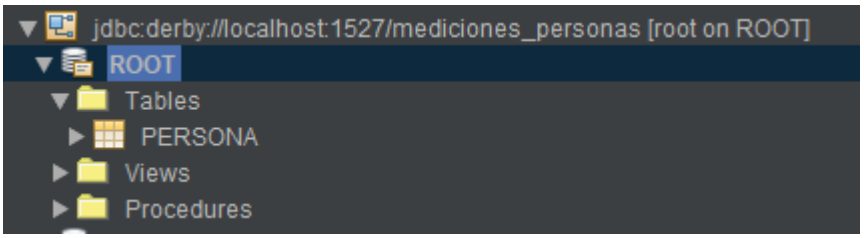
Creación de las tablas dentro del asistente de NetBeans. Ubicarse en el esquema de la base de datos, expandir el nodo llamado ROOT(o como le hayas llamado), seleccionar 'Tables', dar clic derecho y seleccionar 'Execute Command..'



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



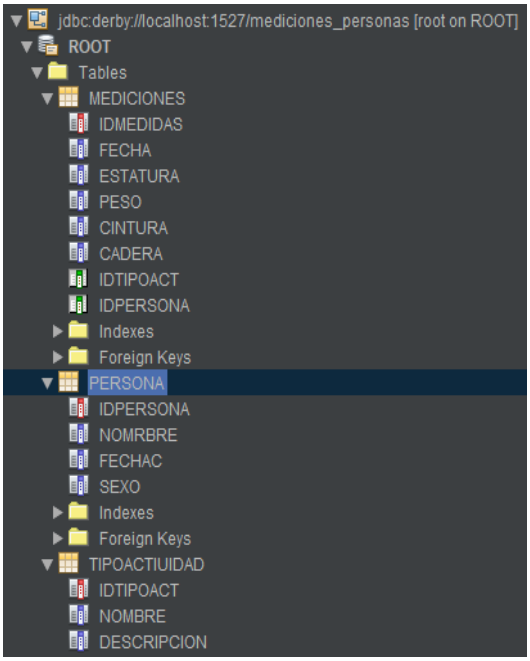
Al terminar la trascripción, activar el botón marcado entre un círculo rojo, que está en la parte superior derecha de la ventana, el cual ejecuta la orden del ‘Sql’. La tabla creada debe mostrarse en la carpeta ’Tables’ del esquema (llamado ROOT).



1.4. Ahora con la creación de la tabla TIPOACTIVIDAD

1.5. Por último crea la tabla MEDICIONES

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 crea otra instancia se regrese la misma conexión creada al inicio

   2.3. Atributos de la clase.

   2.4. Constructor. Utiliza como parámetros: url, usuario y password de la base de datos. El constructor es privado por razones de seguridad ya que es ahí donde se maneja los datos de súper usuario de la base de datos.

   2.5. Método para crear la conexión: Este método crea la conexión.

   2.6. Método que regresa la conexión creada: Este método su función es regresar la conexión que se realiza en la petición para acceder a la base de datos.

   2.7. Método para cerrar la conexión: una vez que el usuario ya no hace peticiones a la base de datos tiene que haber una función que la cierre en este caso es este método.

3. Creación de la clase de manejo de datos Esta clase llama a la conexión para acceder a los datos.

   3.1. Se crean los atributos de dicha clase.

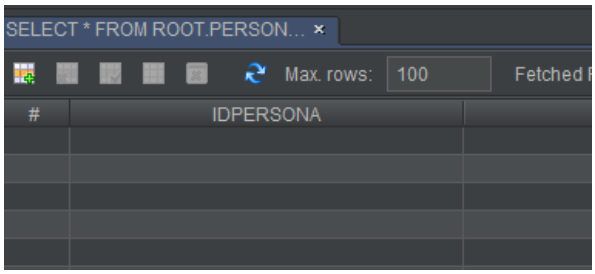
   3.2. Constructor. Aquí se crea la conexión y se obtiene un objeto de esta para el manejo de los datos.

   3.3. Consulta de persona. Aquí se hace la consulta a la tabla persona y se regresa en una lista. Consulta de tipo de actividad. Aquí se hace la consulta a la tabla actividad y el resultado se regresa en una lista. Se crea el dato CAMPOS_ACTIVIDAD.

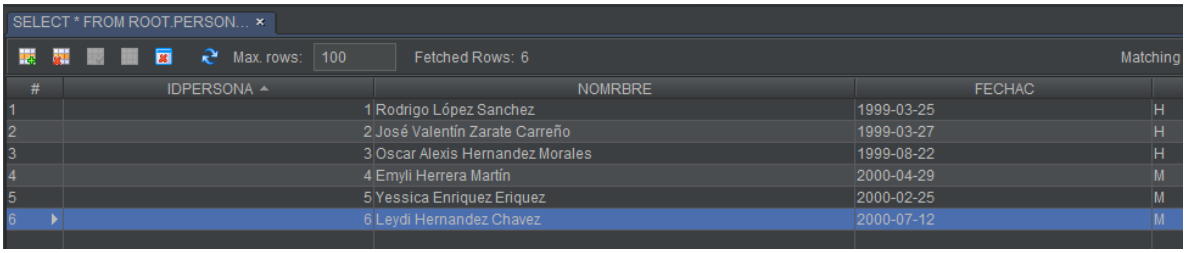
4. . Inserción de datos para prueba Inserción de datos con el asistente.

   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’.

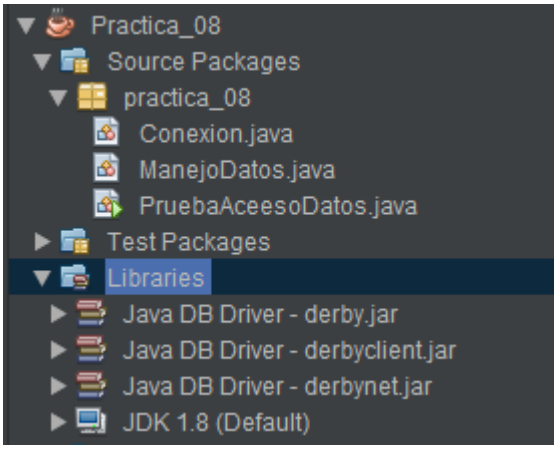


4.3. En la ventana que aparece agrega datos de personas



5. Creación de la clase prueba
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’.

   Al terminar esta acción en la carpeta ‘Libraries’ deben mostrarse las bibliotecas incorporadas:



   6.2. . Activar el servidor. Ubicarse en la ventana ‘Services’ y en JavaDB, dar clic derecho, activando ‘Start’.
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:

```
run:

1                       Rodrigo López Sanchez           1999-03-25              H
2                       José Valentín Zarate Carreño    1999-03-27              H
3                       Oscar Alexis Hernandez Morales  1999-08-22              H
4                       Emyli Herrera Martín            2000-04-29              M
5                       Yessica Enriquez Eriquez        2000-02-25              M
6                       Leydi Hernandez Chavez          2000-07-12              M
```

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

Max. rows: 100 Fetched Rows: 4 Matching Rows:

#	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:

```
run:

1                       SEDENTARIO              NO hacer practicamente nada de ejercicio
2                       LIGERAMENTE ACTIVAS     Hacer ejercicio suave de 1 a 3 veces por semana
3                       MODERADAMENTE ACTIVAS   Hacer deporte 3 a 5 veces por semana
4                       MUY ACTIVAS             Hacer deporte 6 a 7 dias por semanaBUILD SUCCESSFUL (total time: 0 seconds)
```

Código

Script base de datos

```sql
CREATE TABLE PERSONA(
IDPERSONA INTEGER PRIMARY KEY NOT NULL
        GENERATED ALWAYS AS IDENTITY
        (START WITH 1, INCREMENT  BY 1),
NOMRBRE VARCHAR(30) NOT NULL,
FECHAC  DATE,
SEXO CHAR
);

-- Complementa los campos de la tabla TIPOACTIUIDAD
CREATE TABLE TIPOACTIUIDAD
(IDTIPOACT INTEGER PRIMARY KEY NOT NULL GENERATED ALWAYS AS IDENTITY (START WITH 1, INCREMENT BY 1),
NOMBRE  VARCHAR(30) NOT NULL,
DESCRIPCION  VARCHAR(60) NOT NULL
);

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 TIPOACTIUIDAD(IDTIPOACT),
IDPERSONA INT REFERENCES PERSONA(IDPERSONA)
);
```

```java
package practica_08;

import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.SQLException;
import java.util.logging.Level;
import java.util.logging.Logger;

public class Conexion {

    private static Connection coneccion; //contenida en el paquete sql
    private static Conexion conexion;// instacia a utilizar
    private static int numConexiones;//controla el numero de veces que se accedió

    private Conexion(String url, String usuario, String password) throws ClassNotFoundException {
        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);
        }
    }
```

```
33          public static Conexion getConexion(String url, String usuario, String password)
34                  throws ClassNotFoundException {
35              numConexiones++;
36              if (conexion == null) {
37                  conexion = new Conexion(url, usuario, password);
38              }
39              return conexion;
40          }
41          public static  Connection getConeccion(){
42          return  coneccion;
43          }
44
45          public  boolean cerrarConexion(){
46              try {
47                  if (coneccion != null){
48                  if(numConexiones==1){
49                  coneccion.close();
50                  return true;
51                  }
52                  else
53                  numConexiones--;
                    return  false;
55
56                  }
57              } catch (SQLException ex) {
58                  System.out.println("Error a tratar de cerrar la conexion "+ex);
59              }
60          return false;
61          }
62      }
63
```

```
1   package practica_08;
2
3   import java.sql.Connection;
4   import java.sql.Date;
5   import java.sql.ResultSet;
6   import java.sql.Statement;
7   import java.text.DateFormat;
8   import java.text.SimpleDateFormat;
9   import java.util.ArrayList;
10  import java.util.List;
11
12  public class ManejoDatos {
13
15      private Connection conexion;//acceso a conexion
15      private Conexion crearConexion;//Crea conexion
16      private final int CAMPOS_PERSONA = 4;
17      private final int CAMPOS_ACTIUIDAD = 3;
18
19      public ManejoDatos() throws ClassNotFoundException {
20          crearConexion =
        crearConexion.getConexion("jdbc:derby://localhost:1527/mediciones_personas", "root", "123456789");
            conexion = crearConexion.getConeccion();
23      }
24
```

```java
    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((Integer) rs.getInt(1));
                dat[1] = rs.getString(2);
                dat[2] = fecha.format((Date) rs.getDate(3));
                dat[3] = rs.getString(4);
                datos.add(dat);
            }
        } catch (Exception e) {
            System.err.println("Error al conexion consultar persona" + e);

        }
        return datos;

    }
```

```java
    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_ACTIUIDAD];
                //Estructura del registro actividad
                dat[0] = String.valueOf((Integer) rs.getInt(1));
                dat[1] = rs.getString(2);
                dat[2] = rs.getString(3);
                datos.add(dat);
            }
        } catch (Exception e) {
            System.err.println("Error en conexion consultar actividad" + e);
        }
        return datos;

    }
}
```

```java
package practica_08;

import java.util.ArrayList;

public class PruebaAceesoDatos {

    public static void main(String arg[]) throws ClassNotFoundException {
//        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("%-33s", reg[c]);
//            }
//
//        }
     String consulta = "SELECT * FROM ROOT.TIPOACTIUIDAD";
        ManejoDatos baseDatos = new ManejoDatos(); // atributo de la clase ManejoDatos
        ArrayList<Object[]> actividad = (ArrayList<Object[]>) baseDatos.conexionconsultaActividad(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("%-33s", reg[c]);
            }

        }

    }
}

```

Pruebas:

```
run:
Consulta tabla PERSONA
1                               Rodrigo López Salinas           2020-03-25                      H
2                               José Valentín Zarate Carreño    1999-03-27                      H
3                               Oscar Alexis Hernandez Morales  1999-08-22                      H
4                               Emyli Herrea Martín             2020-04-29                      M
5                               Yessica Hernandez Chavez        2000-02-25                      M
6                               Leidy Hernandez Chavez          2000-07-12                      H
Consulta tabla TIPOACTIUIDAD
1                               Rodrigo López Salinas           2020-03-25                      H
2                               José Valentín Zarate Carreño    1999-03-27                      H
3                               Oscar Alexis Hernandez Morales  1999-08-22                      H
4                               Emyli Herrea Martín             2020-04-29                      M
5                               Yessica Hernandez Chavez        2000-02-25                      M
6                               Leidy Hernandez Chavez          2000-07-12                      H
BUILD SUCCESSFUL (total time: 0 seconds)
```