

- genericidad.java:

```
import java.sql.Connection;
import java.sql.Date;
import genericidad.OperacionDB;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.sql.Statement;
import genericidad.ModeloPair.Pair;

public class Genericidad {

    public static void main(String[] args) {

        Connection con = null;
        try {
            con = OperacionDB.getConnection();

            Statement stmt = con.createStatement();
            ResultSet rs = stmt.executeQuery("SELECT id FROM DetoxPersonas");

            while (rs.next()) {
                int personaId = rs.getInt("id");
                OperacionDB.actualizarIdad(personaId);
            }

            String nombre = "Paula Gomez";
            Date fechaNacimiento = Date.valueOf("2011-09-11");
            String ego = "La Nueva";
            AgregarPersonas.agregarPersona(nombre, fechaNacimiento, ego);

            stmt = con.createStatement();
            rs = stmt.executeQuery(
                "SELECT dp.Nombre, dp.Edad, ap.Ego * " +
                "FROM DetoxPersonas dp * " +
                "JOIN SaludPersonas ap ON dp.id = ap.id");

            while (rs.next()) {
                String nombrePersona = rs.getString("Nombre");
                int edad = rs.getInt("Edad");
                String egoObtenida = rs.getString("Ego");
                ModeloPair modeloPair = new ModeloPair();
                Pair<Integer, String> Nombre = modeloPair.new Pair<>(edad, nombrePersona);
                Pair<String, String> DetoxPersona = modeloPair.new Pair<>(egoObtenida, nombrePersona);
                System.out.println("Personas: " + Nombre.toString() + ", Detox adicionales: " + DetoxPersona.toString());
            }

        } catch (SQLException e) {
            System.out.println("Error: " + e.getMessage());
        } finally {
            try {
                if (con != null) con.close();
            } catch (SQLException e) {
                System.out.println("Error al cerrar la conexión: " + e.getMessage());
            }
        }
    }
}
```

- ModeloPair.java

```
package genericidad;

/**
 *
 * @author Andrés
 */
public class ModeloPair {

    public class Pair<T, U> {

        private T first;
        private U second;

        public Pair(T first, U second) {
            this.first = first;
            this.second = second;
        }

        public T getFirst() {
            return first;
        }

        public U getSecond() {
            return second;
        }

        @Override
        public String toString() {
            return "(" + first + ", " + second + ")";
        }
    }
}
```

- ConexionGen.java

```
package Genericidad;  
  
import java.sql.Connection;  
import java.sql.DriverManager;  
import java.sql.SQLException;  
import javax.swing.JOptionPane;  
  
public class ConexionGen {  
  
    public static Connection getCon() {  
        Connection con = null;  
        try {  
            Class.forName("com.mysql.cj.jdbc.Driver");  
            con = DriverManager.getConnection("jdbc:mysql://localhost:3306/genericidaddatos", "root", "");  
        } catch (Exception e) {  
            JOptionPane.showMessageDialog(null, e, "Error", JOptionPane.ERROR_MESSAGE);  
        }  
        return con;  
    }  
  
    public static void closeConnection(Connection con) {  
        if (con != null) {  
            try {  
                if (!con.isClosed()) {  
                    con.close();  
                }  
            } catch (SQLException e) {  
                JOptionPane.showMessageDialog(null, e, "Error", JOptionPane.ERROR_MESSAGE);  
            }  
        }  
    }  
}
```

- Operaciondb.java

```

package genericidad;

import java.awt.HeadlessException;
import javax.swing.JOptionPane;
import java.sql.*;
import Genericidad.ConexionGen;

public class Operaciondb {

    public static Connection getConnection() {
        Connection con = null;
        try {
            con = ConexionGen.getCon();
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e, "Mensaje", JOptionPane.ERROR_MESSAGE);
        }
        return con;
    }

    public static void setData(String query, String message) {
        try (Connection con = ConexionGen.getCon();
            Statement st = con.createStatement()) {
            st.executeUpdate(query);
            if (!message.equals("")) {
                JOptionPane.showMessageDialog(null, message);
            }
        } catch (HeadlessException | SQLException e) {
            JOptionPane.showMessageDialog(null, e, "Mensaje", JOptionPane.ERROR_MESSAGE);
        }
    }

    public static ResultSet getData(String query) {
        Connection con = null;
        Statement st = null;
        ResultSet rs = null;
        try {
            con = ConexionGen.getCon();
            st = con.createStatement();
            rs = st.executeQuery(query);
            return rs;
        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e, "Mensaje", JOptionPane.ERROR_MESSAGE);
            return null;
        }
    }
}

```

```

public static PreparedStatement prepareStatement(String query) {
    try {
        Connection con = ConexionGen.getCon();
        return con.prepareStatement(query);
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e, "Mensaje", JOptionPane.ERROR_MESSAGE);
        return null;
    }
}

```

```

public static ResultSet getAllPersonIds() {
    String query = "SELECT id FROM DatosPersona";
    Connection con = null;
    Statement st = null;
    ResultSet rs = null;
    try {
        con = ConexionGen.getCon();
        st = con.createStatement();
        rs = st.executeQuery(query);
        return rs;
    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, e, "Mensaje", JOptionPane.ERROR_MESSAGE);
        return null;
    }
}

```

```

public static void actualizarEdad(int personaId) {
    String query = "SELECT TIMESTAMPTDIFF(YEAR, s.FechaNacimiento, CURDATE()) AS edad_calculada "
        + "FROM SaludPersona s WHERE s.id = ?";
    try (Connection con = ConexionGen.getCon();
        PreparedStatement pst = con.prepareStatement(query)) {
        pst.setInt(1, personaId);
        try (ResultSet rs = pst.executeQuery()) {
            int edad = 0;
            if (rs.next()) {
                edad = rs.getInt("edad_calculada");
            }
            String updateQuery = "UPDATE DatosPersona SET Edad = ? WHERE id = ?";
            try (PreparedStatement updatePst = con.prepareStatement(updateQuery)) {
                updatePst.setInt(1, edad);
                updatePst.setInt(2, personaId);
                updatePst.executeUpdate();
            }
        }
    }
} catch (SQLException e) {
    JOptionPane.showMessageDialog(null, e, "Mensaje", JOptionPane.ERROR_MESSAGE);
}
}

```

```

public static void main(String[] args) {
    try (ResultSet rs = getAllPersonIds()) {
        if (rs != null) {
            while (rs.next()) {
                int personaId = rs.getInt("id");
                actualizarEdad(personaId);
            }
        }
    } catch (SQLException e) {
        JOptionPane.showMessageDialog(null, e, "Mensaje", JOptionPane.ERROR_MESSAGE);
    }
}
}

```

- TablasGen.java

```
package genericidad;

import javax.swing.JOptionPane;

/**
 *
 * @author Andrés
 */
public class TablasGen {
    public static void main(String[] args) {
        try{
            String datosTable = "CREATE TABLE `DatosPersona` (id int AUTO_INCREMENT primary key, Nombre varchar(30), Edad int(3))";
            Operaciondb.setData(datosTable, "Tabla DatosPersona creada correctamente");

            String saludTable = "CREATE TABLE `SaludPersona` (id int AUTO_INCREMENT primary key, Eps varchar(30), FechaNacimiento DATE, " +
                                "FOREIGN KEY (id) REFERENCES DatosPersona(id) )";
            Operaciondb.setData(saludTable, "Tabla SaludPersona creada correctamente");

        } catch (Exception e) {
            JOptionPane.showMessageDialog(null, e);
        }
    }
}
```

- AgregarPersona.java

```
package genericidad;

import ...? linea

public class AgregarPersona {

    // Método para agregar una persona a la base de datos
    public static void agregarPersona(String nombre, Date fechaNacimiento, String eps) {
        Connection con = null;
        PreparedStatement psDatosPersona = null;
        PreparedStatement psSaludPersona = null;
        try {
            con = Operaciondb.getConnection();

            // Insertar en la tabla DatosPersona
            String queryDatosPersona = "INSERT INTO DatosPersona (Nombre) VALUES (?)";
            psDatosPersona = con.prepareStatement(queryDatosPersona, PreparedStatement.RETURN_GENERATED_KEYS);
            psDatosPersona.setString(1, nombre);
            psDatosPersona.executeUpdate();

            // Obtener el ID generado para la nueva persona
            ResultSet rs = psDatosPersona.getGeneratedKeys();
            int personaId = 0;
            if (rs.next()) {
                personaId = rs.getInt(1);
            }

            // Insertar en la tabla SaludPersona
            String querySaludPersona = "INSERT INTO SaludPersona (id, Eps, FechaNacimiento) VALUES (?, ?, ?)";
            psSaludPersona = con.prepareStatement(querySaludPersona);
            psSaludPersona.setInt(1, personaId);
            psSaludPersona.setString(2, eps);
            psSaludPersona.setDate(3, fechaNacimiento);
            psSaludPersona.executeUpdate();

            JOptionPane.showMessageDialog(null, "Persona agregada correctamente.");

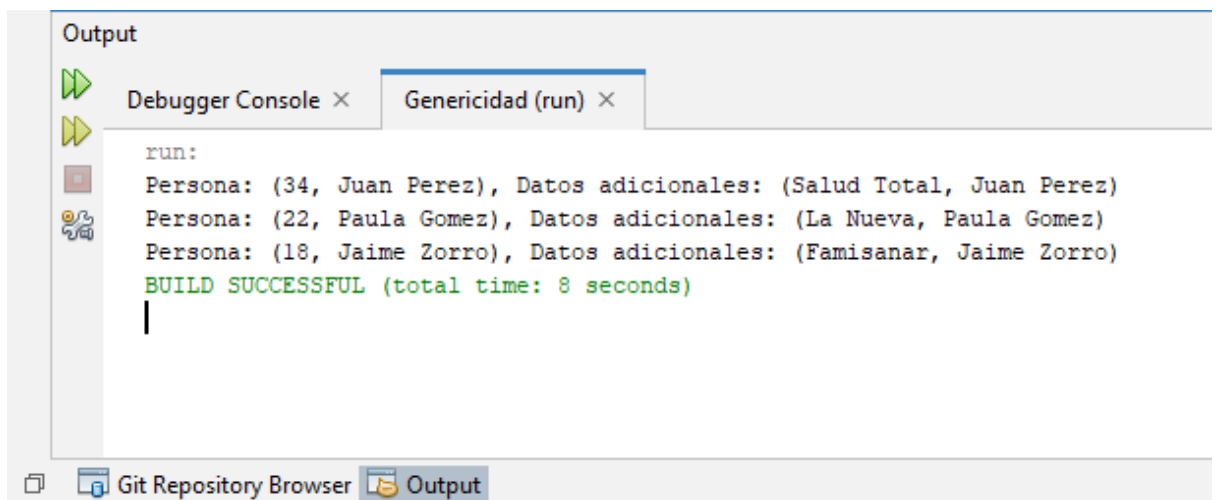
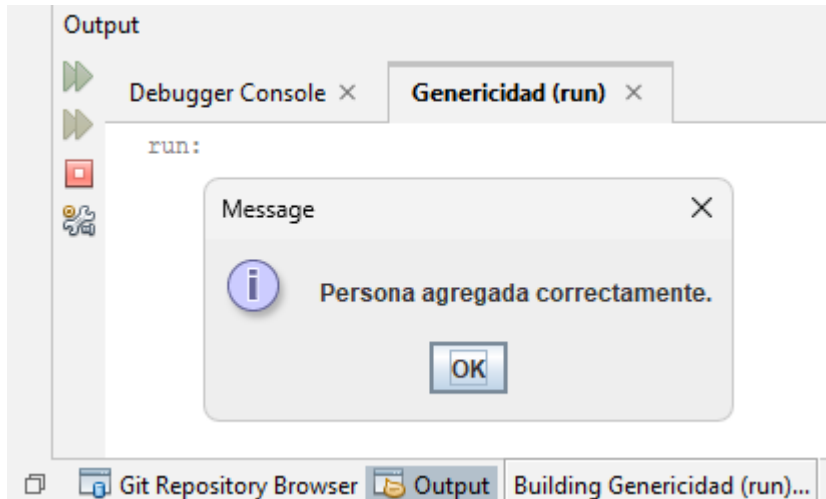
            Operaciondb.actualizarEdad(personaId);

        } catch (SQLException e) {
            JOptionPane.showMessageDialog(null, "Error al agregar la persona: " + e.getMessage());
        } finally {
            try {
                if (psDatosPersona != null) psDatosPersona.close();
                if (psSaludPersona != null) psSaludPersona.close();
                if (con != null) con.close();
            } catch (SQLException e) {
                JOptionPane.showMessageDialog(null, "Error al cerrar los recursos: " + e.getMessage());
            }
        }
    }
}
```

- Funcionamiento:
- Persona a ingresar:

```
String nombre = "Jaime Zorro";
Date fechaNacimiento = Date.valueOf("2005-11-21");
String eps = "Famisanar";
AgregarPersona.agregarPersona(nombre, fechaNacimiento, eps);
```







- Terminal:











- Base de datos:

Table	Action	Rows	Type	Collation	Size	Overhead
<input type="checkbox"/> datospersona	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	16.0 KiB	-
<input type="checkbox"/> saludpersona	★ Browse Structure Search Insert Empty Drop	2	InnoDB	utf8mb4_general_ci	32.0 KiB	-
2 tables	Sum	4	InnoDB	utf8mb4_general_ci	48.0 KiB	0 B

- datospersona:

<div>↩ T ↪</div>				id	Nombre	Edad
<input type="checkbox"/>	 Edit	 Copy	 Delete	23	Paula Gomez	22
<input type="checkbox"/>	 Edit	 Copy	 Delete	26	Jaime Zorro	18

- saludpersona:

				id  1	personaid	Eps	FechaNacimiento
<input type="checkbox"/>	 Edit	 Copy	 Delete	23	23	La Nueva	2001-09-11
<input type="checkbox"/>	 Edit	 Copy	 Delete	26	26	Famisanar	2005-11-21