

Alan Perez

Primero al crear el GUI FORM le agregamos los botones, textpanels, etc. Que luego lo vamos a programar

CORRECCION PRUEBA ALAN PEREZ			
CODIGO	<input type="text"/>	BUSCAR POR CODIGO	
CEDULA	<input type="text"/>		
NOMBRE	<input type="text"/>	BUSCAR POR NOMBRE	
FECHA DE NACIMIENTO	<input type="text"/>		
SIGNO DEL	ELEJIR <input type="text"/>	BUSCAR POR SIGNO	
<hr/>			
BORRAR REGISTRO	ACTUALIZAR REGISTRO	INGRESAR REGISTRO	
<hr/>			
LIMPIAR FORMULARIO			
<hr/>			
ESPERANDO ACCION			

Aqui podemos ver como se llaman los botones, panels, combobox,etc y a demas he declarado los datos que se va a usar para la conexcion como servidor, base de datos, usuario, contraseña y certificado que necesita SQL Server.

```
public class PRUEBA {
    String Nombre, Fecha, Signo, Cedula;
    int Codigo;
    String conexion= "jdbc:sqlserver://localhost:1433;" +
        "database=CorreccionP2B;" +
        "user=root;" +
        "password=root_1;" +
        "trustServerCertificate=true;";

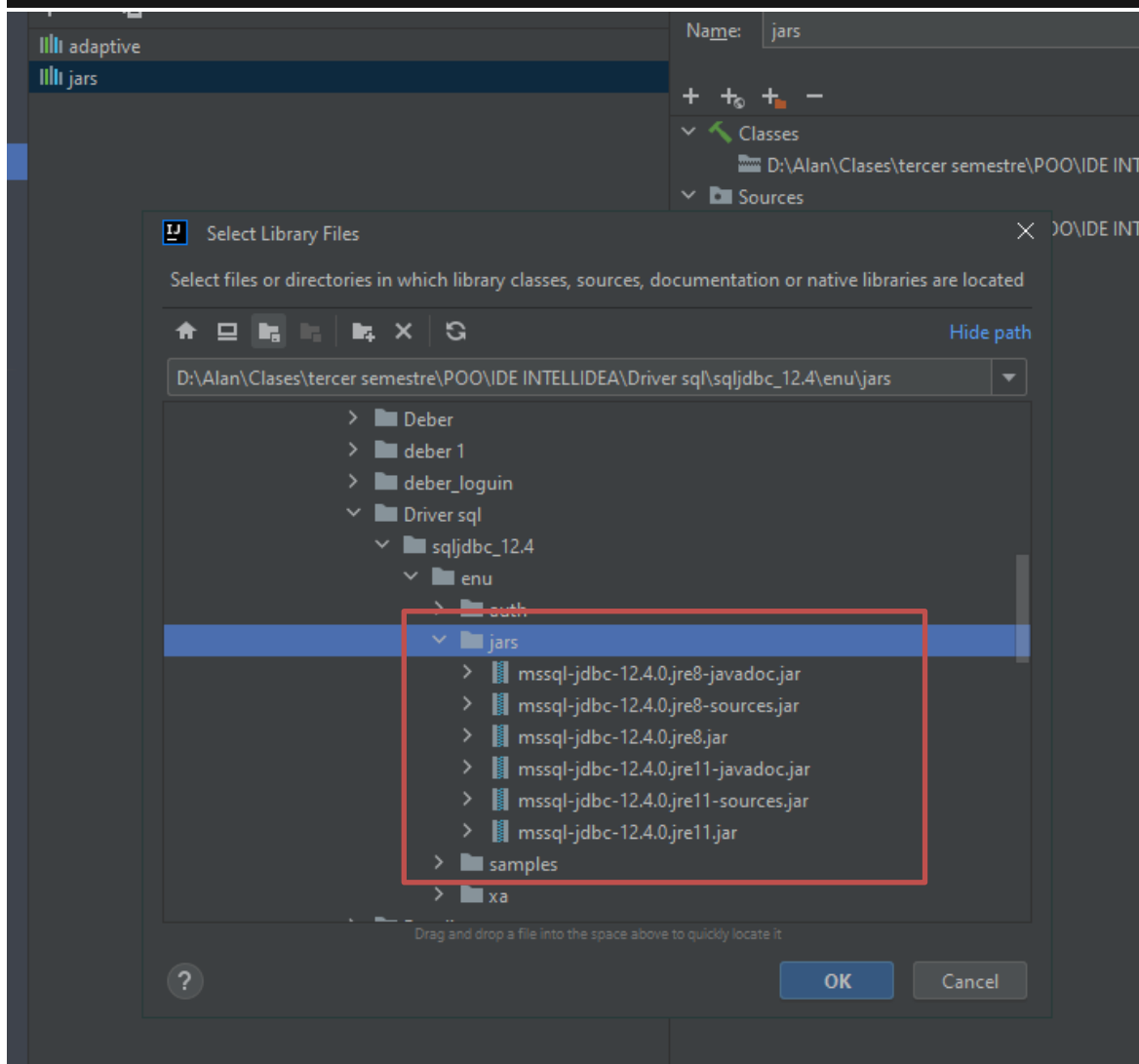
    private JButton BBORRAR;
    private JButton BACTUALIZAR;
    private JButton BINGRESAR;
    private JButton BLIMPIAR;
    private JButton BUSCARPORCODIGOButton;
    private JButton BUSCARPORNOMBREButton;
    private JButton BOTONBUSCARPORSIGNOButton;
    private JTextField TCODIGO;
    private JTextField TCEDULA;
    private JTextField TNOMBRE;
    private JTextField TFECHA;
    private JComboBox CSIGNO;
    private JLabel LCODIGO;
    private JLabel LCEDULA;
    private JLabel LNOMBRE;
    private JLabel LSIGNO;
    private JLabel LFECHA;
    private JPanel PANTALLA;
    private JLabel LABEL1;
    private JLabel TITULO;
```

Para que funcione el programa tambien necesitaremos el driver de SQL Server que se descarga desde la pagina de Microsoft y que debemos poner en las librerias del proyecto

Download

Version 12.4 is the latest general availability (GA) version. It supports Java 8, 11, 17, and 20. If you need to use an older Java runtime, see the [Java and JDBC specification support matrix](#) to see if there's a supported driver version you can use. We're continually improving Java connectivity support. As such we highly recommend that you work with the latest version of Microsoft JDBC driver.

- Download Microsoft JDBC Driver 12.4 for SQL Server ([zip](#))
- Download Microsoft JDBC Driver 12.4 for SQL Server ([tar.gz](#))



Ahora ya con estos datos ya podemos empezar a codificar los botones en los cuales utilizaremos los datos que se envíen desde la ventana java y luego con un try catch, realizaremos la conexión en la cual establecemos el Query que va a tener parametros ? que luego los enviaremos con Statement.setInt o setString según sea requerido, con esto enviaremos el query dependiendo de lo que necesitemos, como en este caso se usa Insert Into Datos

```
public PRUEBA() {
    BINGRESAR.addActionListener(new ActionListener() {
        @Override
        public void actionPerformed(ActionEvent e) {

            Codigo = Integer.parseInt(TCODIGO.getText());
            Cedula = TCEDULA.getText();
            Nombre = TNOMBRE.getText();
            Fecha = TFECHA.getText();
            Signo = (String) CSIGNO.getSelectedItem();
            String QUERY="INSERT INTO DATOS(CODIGO,CEDULA,NOMBRE,FECHA_NACIMIENTO,SIGNO)" +
                "VALUES(?,?,?,?,?)";
            try(Connection conn=DriverManager.getConnection(conexion))
            {
                PreparedStatement statement = conn.prepareStatement(QUERY);
                // Establecer valores para los parámetros de la sentencia SQL
                statement.setInt( parameterIndex: 1,Codigo);
                statement.setString( parameterIndex: 2, Cedula);
                statement.setString( parameterIndex: 3, Nombre);
                statement.setString( parameterIndex: 4, Fecha);
                statement.setString( parameterIndex: 5, Signo);
                // Ejecutar sentencia SQL
                int rowsInserted = statement.executeUpdate();
                if (rowsInserted > 0) {
                    LABEL1.setText("Datos insertados correctamente");
                }
            }
            catch (SQLException ex){
                throw new RuntimeException(ex);
            }
        }
    });
    BACTUALIZAR.addActionListener(new ActionListener() {
        @Override
```

En este caso es lo mismo pero con Update necesitamos indicar un valor que indique donde se va a actualizar los datos, en este caso se utiliza otra variable en la cual se indica el ID, con el cual vamos a indicar que todos los datos que tengan ese ID van a cambiar, esto con el query Update que tiene la palabra **WHERE**

```
BACTUALIZAR.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        int idToUpdate; // Reemplaza 1 con el ID de la fila que deseas actualizar
        String nuevoNombre; //
        idToUpdate= Integer.parseInt(TCODIGO.getText());
        Cedula = TCEDULA.getText();
        Nombre = TNOMBRE.getText();
        Fecha = TFECHA.getText();
        Signo = (String) CSIGNO.getSelectedItem();
        String UPDATE_QUERY = "UPDATE DATOS SET CEDULA = ?, NOMBRE = ?, FECHA_NACIMIENTO = ? , SIGNO = ? WHERE CODIGO = ?";
        try (Connection conn = DriverManager.getConnection(conexion)) {
            PreparedStatement statement = conn.prepareStatement(UPDATE_QUERY);
            // Establecer nuevos valores para los campos siguiendo el orden del query
            statement.setString( parameterIndex: 1,Cedula);
            statement.setString( parameterIndex: 2,Nombre);
            statement.setString( parameterIndex: 3,Fecha);
            statement.setString( parameterIndex: 4,Signo);
            // Establecer valor para el parámetro del ID de la fila que deseas actualizar
            statement.setInt( parameterIndex: 5, idToUpdate);
            // Ejecutar sentencia UPDATE
            int rowsUpdated = statement.executeUpdate();

            if (rowsUpdated > 0) {
                LABEL1.setText("Se actualizaron los datos correctamente");
            } else {
                LABEL1.setText("No hay datos con ese código, ingrese nuevamente");
            }
        } catch (SQLException ex) {
            ex.printStackTrace();
        }
    }
}
```

En este caso es igual pero en este caso cambiaria de donde se obtienen los datos porque necesitamos `getSelectedItem`, de alli el codigo sigue siendo igual que `select` y `where`

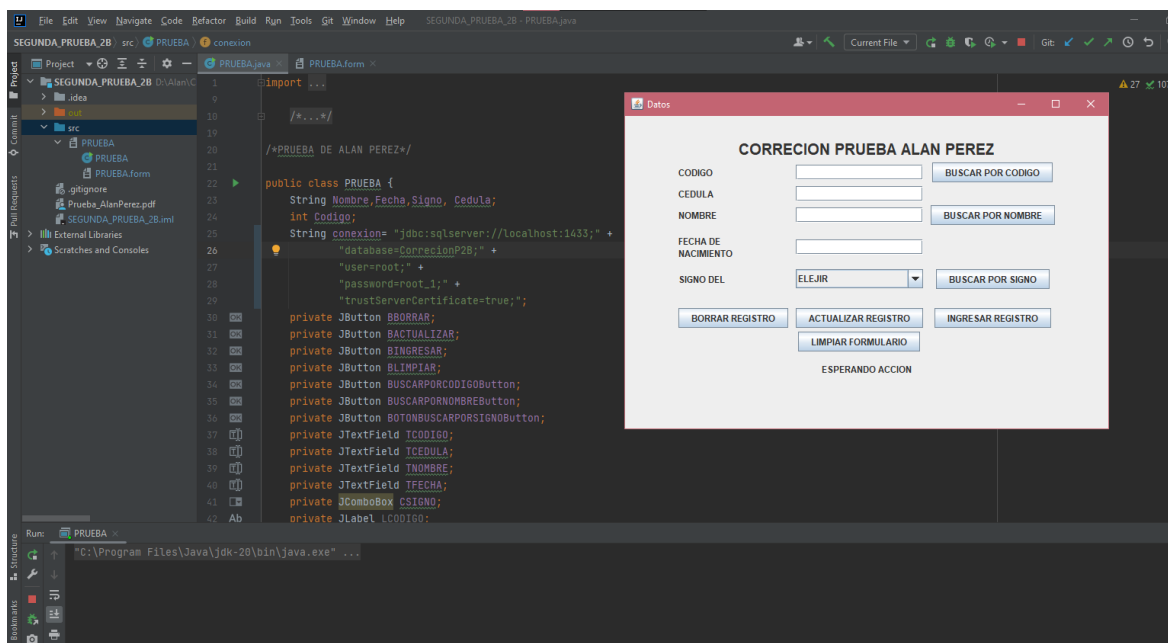
```

    }
}

BTBONBUSCARPORSIGNOButton.addActionListener(new ActionListener() {
    @Override
    public void actionPerformed(ActionEvent e) {
        String SIGNO1;
        SIGNO1 = (String) CSIGNO.getSelectedItem();
        String SELECT_QUERY="SELECT * FROM DATOS WHERE SIGNO = ?";
        try(Connection conn=DriverManager.getConnection(conexion);)
        {
            PreparedStatement statement = conn.prepareStatement(SELECT_QUERY);
            statement.setString( parameterIndex: 1, SIGNO1);
            ResultSet rs = statement.executeQuery();
            while (rs.next()) {
                Codigo = Integer.parseInt(rs.getString( columnLabel: "CODIGO"));
                Cedula = rs.getString( columnLabel: "CEDULA");
                Nombre = rs.getString( columnLabel: "NOMBRE");
                Fecha = rs.getString( columnLabel: "FECHA_NACIMIENTO");
            }
            TCODIGO.setText(String.valueOf(Codigo));
            TCEDULA.setText(Cedula);
            TNOMBRE.setText(Nombre);
            TFECHA.setText(Fecha);
        }
        catch (SQLException eX){
            throw new RuntimeException(eX);
        }
        LABEL1.setText("Datos impresos / 0 si es que no existe");
    }
});

BUSCARPORCODIGOButton.addActionListener(new ActionListener() {

```



Datos

CORRECCION PRUEBA ALAN PEREZ

CODIGO

BUSCAR POR CODIGO

CEDULA

NOMBRE

BUSCAR POR NOMBRE

FECHA DE NACIMIENTO

SIGNO DEL

ELEJIR

BUSCAR POR SIGNO

BORRAR REGISTRO

ACTUALIZAR REGISTRO

INGRESAR REGISTRO

LIMPIAR FORMULARIO

ESPERANDO ACCION

SQLQuery1.sql - DE...5BK60\alans (64))*

```

Create database CorreccionP2B
use Correccionp2b

CREATE TABLE DATOS(
    CODIGO INT,
    CEDULA INT,
    NOMBRE VARCHAR(25),
    FECHA_NACIMIENTO VARCHAR(25),
    SIGNO VARCHAR(25))

select * from DATOS
  
```

Results

Messages

	CODIGO	CEDULA	NOMBRE	FECHA_NACIMIENTO	SIGNO
1	1	172346274	Alan	20/12/2002	Aries

Datos

CORRECCION PRUEBA ALAN PEREZ

CODIGO

BUSCAR POR CODIGO

CEDULA

NOMBRE

BUSCAR POR NOMBRE

FECHA DE NACIMIENTO

SIGNO DEL

Aries

BUSCAR POR SIGNO

BORRAR REGISTRO

ACTUALIZAR REGISTRO

INGRESAR REGISTRO

LIMPIAR FORMULARIO

Datos insertados correctamente

SQLQuery1.sql - DE...5BK60\alans (64))*

```

Create database CorreccionP2B
use Correccionp2b

CREATE TABLE DATOS(
    CODIGO INT,
    CEDULA INT,
    NOMBRE VARCHAR(25),
    FECHA_NACIMIENTO VARCHAR(25),
    SIGNO VARCHAR(25))

select * from DATOS
  
```

Results

Messages

	CODIGO	CEDULA	NOMBRE	FECHA_NACIMIENTO	SIGNO
--	--------	--------	--------	------------------	-------

Datos

CORRECCION PRUEBA ALAN PEREZ

CODIGO

BUSCAR POR CODIGO

CEDULA

NOMBRE

BUSCAR POR NOMBRE

FECHA DE NACIMIENTO

SIGNO DEL

Aries

BUSCAR POR SIGNO

BORRAR REGISTRO

ACTUALIZAR REGISTRO

INGRESAR REGISTRO

LIMPIAR FORMULARIO

Se eliminó la fila correctamente