

Práctica de Laboratorio 1

Conexión Java – SQL Server con Maven Base de Datos: VidaSana

Objetivo

En esta práctica usted creará un proyecto Java de consola usando el framework Maven en Apache NetBeans y realizará operaciones básicas hacia SQL Server usando la API JDBC. Se realizarán inserciones, consultas y la invocación de un stored procedure sobre las tablas Pacientes, Medicos y Citas. Requisitos previos: Crear la base de datos vida_sana con tablas, stores procedure y logins que haya hecho en prácticas anteriores.

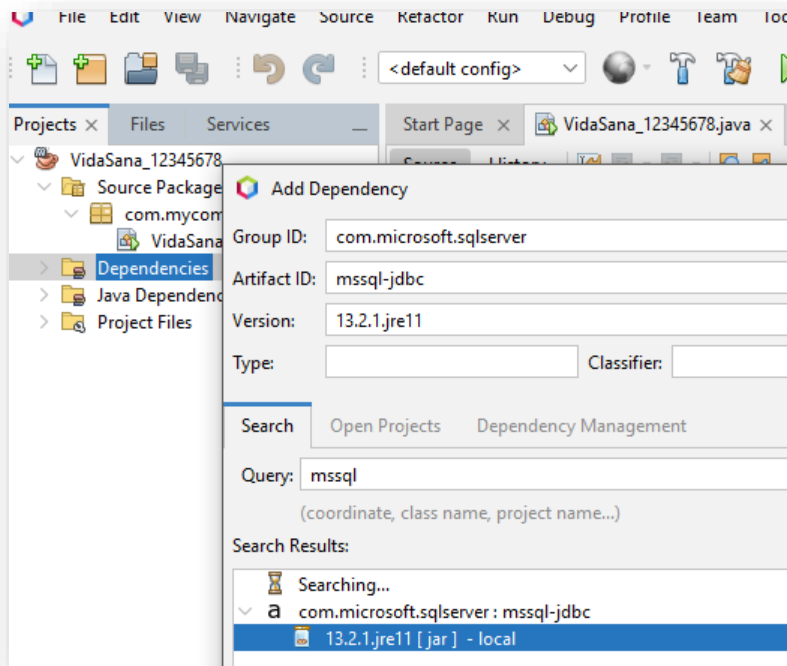
Parte 1 – Crear proyecto Maven en NetBeans

1. Abrir Apache NetBeans
2. File → New Project
3. Seleccionar: Java with Maven → Java Application
4. Nombre del proyecto: VidaSana_suNumeroDeControl
5. Presionar botón “Finish”.

Parte 2 – Agregar dependencia de SQL Server

Aquí está a punto de descargar el JAR desde el repositorio Maven. Este JAR contiene la API responsable de conectarse desde Java hacia SQL Server. Hay dos maneras de hacerlo:

- 2.A) Expanda la estructura de su proyecto, haga click derecho en “Dependencies” y seleccione “Add Dependency”. Busque *mssql*, expanda el resultado, expándalo, seleccione el jar y finalmente agréguelo a su proyecto.



2.B) Alternativamente, puede editar el archivo pom.xml que se encuentra en Project Files.

Abrir el archivo pom.xml y agregar la siguiente dependencia dentro de la sección <dependencies>:

```
<dependency>
  <groupId>com.microsoft.sqlserver</groupId>
  <artifactId>mssql-jdbc</artifactId>
  <version>13.2.1.jre11</version>
</dependency>
```

Parte 3 – Conexión a SQL Server

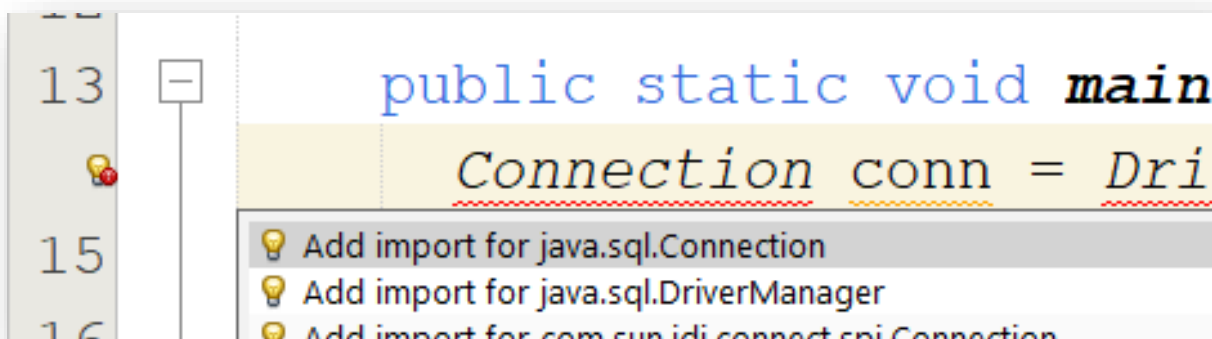
Dentro del método main de su archivo Vida_Sana_XXXXXXX agregue el siguiente código. Ejemplo usando autenticación de SQL Server (login y contraseña) Recuerde adecuar los valores que haya creado:

```
Connection conn = DriverManager.getConnection(
"jdbc:sqlserver://nombreServidor;databaseName=vida_sana;encrypt=true;trustServerCertificate=true",
"loginMedico",
"medico123"
);
```

Parámetros importantes:

- encrypt=true: Indica que la conexión entre Java y SQL Server se cifra.
- trustServerCertificate=true: Permite confiar en el certificado del servidor incluso si no está firmado por una autoridad certificadora.

Nota para su código en NetBeans: Donde le marque error en el código, recuerde que NetBeans (como cualquier otro IDE) lo asiste en importar los paquetes si presiona Ctrl+Shift+i, o hace click en el foco amarillo de la línea con error y seleccione las sugerencias.



Parte 4 – Insertar datos desde Java

```
Statement stmt = conn.createStatement();
```

```
String sqlPaciente = "INSERT INTO Pacientes(IdPaciente, Nombre, FechaNacimiento, Telefono) VALUES  
(1,'Juan Perez','2000-01-01','5551234567)";  
stmt.executeUpdate(sqlPaciente);
```

```
String sqlCita = "INSERT INTO Citas(idCita, Fecha, Hora, Estado, idPaciente, idMedico) VALUES (10,'2026-04-  
01','10:00','Programada',1,1)";  
stmt.executeUpdate(sqlCita);
```

```
System.out.println("Registros insertados correctamente.");
```

Ajuste valores, nombres y tipos de dato de sus columnas según requiera. Observe los resultados en su base de datos y capture pantalla de evidencia. Intente ejecutar de nuevo el query y observe la excepción que Java arroja acerca de que está intentando insertar nuevamente una llave primaria duplicada en la tabla Pacientes. Tome captura de pantalla de dicha excepción para su evidencia.

Parte 5 – Consultar datos

```
String consulta = "SELECT idPaciente, nombre, telefono FROM Pacientes";
```

```
ResultSet rs = stmt.executeQuery(consulta);
```

```
while(rs.next()){
```

```
    int id = rs.getInt(1); // por índice
```

```
    String nombre = rs.getString("nombre"); // por nombre de columna
```

```
    String telefono = rs.getString("telefono");
```

```
    System.out.println(id + " - " + nombre + " - " + telefono);
```

```
}
```

Observación:

- Las columnas pueden accederse por índice (1,2,3...) o por nombre de columna.

Query # 2. Diseñe un query en SQL Server con el diseñador de consultas de tal manera que se despliegue todas las citas de todos los pacientes con su: nombre, teléfono, fecha de cita, nombre de médico y su especialidad. Imprima la lista en Java usando el mismo while. Tome captura pantalla de este resultado.

Parte 6 – Llamar un Stored Procedure

(Requisito previo: debe tener un sp llamado sp_RegistrarPaciente que reciba los parámetros para insertarlo en la tabla Pacientes. Cree el sp si no lo tiene)

```
CallableStatement sp = conn.prepareCall("{call sp_RegistrarPaciente(?,?,?,?)}");

sp.setString(1, "1");
sp.setInt(1, 10);
sp.setString(2, "Sandra Jimenez");
sp.setDate(3, Date.valueOf(LocalDate.of(2000,1,1)) );
sp.setString(4,"6622122112");
sp.execute();

System.out.println("Stored Procedure ejecutado.");
```

Note como hay un setXXXXX con su respectivo tipo de dato en Java que tiene que coincidir en SQL Server. Por ejemplo, si vamos a pasar letras y números, en Java usamos setString y en SQL Server ese valor se tendría que recibir como VARCHAR.

También note como la fecha se hace uso de la clase Date y el método valueOf. Es una manera muy segura de pasar los parámetros tipo DATE.

Parte 7 – Cerrar conexión

```
rs.close();
stmt.close();
conn.close();
```

Preguntas de reflexión

1. ¿Cuál es la diferencia entre executeQuery() y executeUpdate()?
2. ¿Qué ventaja tiene usar stored procedures?
3. ¿Por qué no es recomendable escribir queries concatenando texto directamente?
4. ¿Qué diferencia observaste entre acceder a columnas por índice y por nombre?

Entregables

-Archivo ZIP con el mismo nombre de su proyecto en NetBeans

-Documento Word o PDF con las preguntas de reflexión y sus respuestas, junto con las capturas de pantalla de evidencia.