I.OBJETIVOS

- Que el alumno sea capaz de conocer las clases específicas que posee Java para el manejo de base de datos y el manejo de las API de JDBC.
- Que al alumno sea capaz de aprender el uso de las API JDBC para la creación de consultas.

II. INTRODUCCIÓN

Conexión a Bases de Datos con NetBeans IDE 8 o superior

JDBC (Java DataBase Connectivity) es la tecnología Java que permite a las aplicaciones interactuar directamente con motores de base de datos relacionales.

La API JDBC es una parte integral de la plataforma Java, por lo tanto no es necesario descargar ningún paquete adicional para usarla. JDBC es una interface única que independiza a las aplicaciones del motor de la base de datos.

Un driver JDBC es usado por la JVM para introducir las invocaciones JDBC en invocaciones que la base de datos entiende.



Conectar a la base de datos

Para que una aplicación en Java se comunique con una base de datos usando la API JDBC, se requiere de un conector que realice la conexión de la aplicación a la Base de Datos.

Ese conector es específico para el manejador de base de datos y viene en la forma de un archivo ".jar" o ".zip".

Por ejemplo:

MySQL, está en el archivo: mysql-connector-java5.1.6.zip (mysql-connector-java-5.1.6-bin.jar)

SQL Server, está en el archivo: sqljdbc_1.2.2828.100_esn.zip (sqljdbc.jar)

NetBeans nos permite realizar algunas tareas relacionadas con bases de datos:

- 1. Conectar una aplicación a una base de datos.
- 2. Conectar a NetBeans directamente a una base de datos para crear, eliminar, modificar tablas, agregar, eliminar, modificar registros y hacer consultas de datos.
- 3. Generar aplicaciones a partir de ingeniería inversa.
- 4. Crear persistencia con la Base de Datos.
- 5. Otras innumerables.

Conexión de una aplicación a una base de datos

Para conectar a una aplicación a una base de datos, se requiere:

- 1. Agregar a NetBeans el conector como una biblioteca o librería. Esto permite que el conector esté disponible para los proyectos.
- 2. Agregar a un proyecto el conector. Esto permite que la aplicación se pueda conectar a la base de datos.

Agregar a NetBeans el conector para utilizar una base de datos en MySQL.

En las versiones más recientes de NetBeans, se cuenta ya con las librerías para la mayoría de gestores de bases de datos, o al menos, los más importantes.



Esto no es una regla exclusiva, puede tener una versión un poco antigua de Netbeans (por lo que se le recomienda actualizar), pueden no funcionar los drivers que posee o simplemente no estar incluidos. Certifique que lo posee, revise en su grupo de ventanas en **Services->Drivers** que posee al menos el conector MySQL (Para esta práctica)

¿No se encuentras las librerías?

Nota: Siga este procedimiento solo si no se encuentran o no funciona la librería de MySQL.

Agregaremos a Netbeans el conector de MySQL como ejemplo. El procedimiento es el siguiente:

1. Seleccionar la opción **Tools/Libraries** de la barra de menú de **NetBeans IDE 8.0 o superior** como se muestra en la siguiente figura:



Figura 1

 Aparece un cuadro de diálogo que nos permite administrar las bibliotecas de NetBeans IDE 8.0.

Library Manager
Libraries location: Global Libraries
Libraries:
Absolute Layout ActiveSync. Java Wrapper Library ANT Contrib Beans Binding CDC Build System Ant Utilities CopyLibs Task Exception Handler I dentity Ant Tasks J2ME Ant Extension Jakarta Silde Ant WebDAV Java ME CDC AGUI Free Layout Java ME CDC Personal Profile Fre Java Tree API JAX:WS 2.1 JAX:8 2.1 DBC Design-Time Support - Depr Mere Library Remove Move Depr Move Library Remove Move Depr Move Library Remove Move Depr Move Depr Move Construction Autor of the support - Depr Mere Library Remove Move Depr Mere Library Remove Move Depr Mere Library Remove Move Depr Mere Library Remove Move Depr Move Depr Move Depr Move Depr

Figura 2

Del lado izquierdo del cuadro de dialogo, aparece una ventana con las bibliotecas agregadas a NetBeans. Del lado derecho aparece el nombre de la biblioteca y la trayectoria del archivo con la biblioteca. Para agregar el conector de **MySQL** a NetBeans, hacer clic en el botón **New Library...** y aparece un cuadro de dialogo como el siguiente:

Library Name: MySQL Library Type: Class Libraries	🕽 New Libra	y 🔀
Library Type: Class Libraries	Library <u>N</u> ame:	MySQL
	Library <u>T</u> ype:	Class Libraries
OK Cancel		

Figura 3

3. En este cuadro de dialogo se establece el nombre que tendrá el conector, digitar **MySQL** y hacer clic en el botón **OK.** Y se regresara a la ventana del administrador de bibliotecas. Hacer clic en el botón **Add JAR/Fólder...**

raries location: Global Libraries	
raries:	
Java DB Driver	
Java ME CDC AGUI Free Lay	
	Add JAR/Folder
	Remove
	Move Up
JDBC Run-Time Support - Dej	Move Down
📲 Jersey (JSR-311 RI) Libraries	
JMUnit4CLDC10	
JMUnit4CLDC11	
MySQL	
New Library	
	OK Cancel

4. Hay que seleccionar el archivo que contiene el conector de MySQL: mysql-connector-java-5.0.8-bin.jar (Puede descargalo en el sitio de oracle), después de haberlo seleccionado, hacer clic en Add JAR/Fólder.

	ql-connector-java-5.1.6	2 10 10 10 10 10 10 10 10 10 10 10 10 10	📂 💷 📰
🚞 docs			
🛅 src	inun E 1 6 hin		
mysqi-connector-	Java-5.1.6-Din		

Figura 5

5. Regresamos al administrador de bibliotecas en donde aparece el conector agregado, hacer clic en el botón **OK**.

Cibrary Manager	×
Libraries location: Global Libraries	~
Libraries:	
Java DB Driver Java ME CDC ASUL Free Lay Java ME CDC ASUL Free Lay Java ME CDC Personal Profile Java Tree API JAX-WS 2.1 JDBC Design-Time Support - I JDBC Run-Time Support - I Move Lay Move Lay Move Down	
New Library Remove	
OK Cance	1

Figura 6

Nota:

Los pasos anteriores debe seguirse para hacer la conexión de una aplicación a una Base de Datos de SQL Server también tomar en cuenta los siguientes pasos:

- Seleccionar la opción **Tools/ Libraries** de la barra de menú como se muestra en la figura 1.
- Mostrará un cuadro de dialogo que permite administrar las bibliotecas de **NetBeans IDE 8** como se muestra en la figura 2.
- Agregar una nueva librería (**New Library**), la cual se llamara **MSSqlServer**, el nombre lo colocara en la opción **Library name** así como se muestra en la figura 3.
- Seleccionar el archivo que contiene el conector a SQL Server:

Buscar en: 🛅 esn		👻 🤌	📂 💷 📰
🛅 auth			
🛅 help			
🚞 xa			
📓 sqljdbc			
📓 sqljdbc			
🔊 sqljdbc			
📓 sqljdbc			
sqljdbc	solidbo, iar		Add JAR/Folder

Figura 7

• Regresar al administrador de bibliotecas en donde aparece el conector agregado, así como se muestra en la figura 6, y hacer clic en el botón **OK**

Importante:

Para hacer uso de las conexiones de las bases de datos que se instalaron anteriormente, debe copiar los archivos .jar en la siguiente dirección:

C:\Archivos de programa\Java\jre1.6.0_05\lib\ext Ó

```
C:\Program Files\Java\jdk1.6.0_25\jre\lib\ext
```

Así como se muestra en la siguiente figura:

🚔 ext	
Archivo Edición Ver Favoritos Herramientas Ayuda	
🔇 Atrás 🝷 🌍 🗧 🏂 🔎 Búsqueda 🎼 Carpetas 📰 -	
Dirección 🛅 C:\Archivos de programa\Java\jre1.6.0_05\lib\ext	🗸 🔁 Ir
Tareas de archivo y carpeta the file of the second	
Crear nueva carpeta sunjce_provider Sunjce_provider Executable Jar File Compartir esta carpeta Image: Sunjce_provider Image: Sunjce_provider Executable Jar File 167 KB Image: Sunjce_provider Sunjce_provider Sunjce_provider Sunjce_provider Sunscapi Executable Jar File Sunscapi Sunscapi Sunscapi Sunsca	
Otros sitios Support Sition Support Sition Support Sition Support Sition Support Sition Support Sition Sitien Sition Sitien Siti	
Detalles	
ext Carpeta de archivos	

Figura 8

III. PROCEDIMIENTO

En esta guía trabajaremos la conexión a **MySQL** desde una aplicación de Java creada en NetBeans 8 Ejercicio 1. Crear un proyecto en Netbeans con el nombre Guia4 y realizar todos los pasos que están descritos en la introducción de esta guía.

Ejercicio 1. Creando una Base de datos en MySQL 5.0

¿Tiene instalado en su Equipo MySQL Server?

Realizar los siguientes pasos:

 Hacer clic en el botón Inicio, seleccionar Todos los programas, después seleccionar la opción MySQL, seleccionar la opción MySQL 5.0 y por ultimo dar clic en MySQLCommand Line Client

📑 Yah	00!	•	
🛅 Net	🛅 NetBeans 🔸		
🛅 Cut	ePDF	►	
💼 Му	MySQL Command Line Client		💼 MySQL Server 5.0 🔸
	MySQL Manual MySQL Service Technology Configuration		nd Line Client

- Aparece una ventana en la cual debe digitar la contraseña del usuario root de MySQL, en la opción Enterpassword digitar admin (o el password asignado, pregunte a su instructor), presionar la tecla Enter.
- 3. Digitar create database Guia4;(Toda línea termina con punto y coma), presionar la tecla Enter

MySQL Command Line Client
Enter password: ***** Welcome to the MySQL monitor. Commands end with ; or \g. Your MySQL connection id is 9 to server version: 5.0.27-community-nt
Type 'help;' or '\h' for help. Type '\c' to clear the buffer.
mysql> create database Guia4_LPIII; Query OK, 1 row affected <0.36 sec>
mysq1>

En este paso hemos creado una Base de datos llamada Guia4, para hacer uso de la base de datos debe digitar: **use Guia4;**

4. En este paso vamos a crear una tabla en la BD creada en el paso 4, vamos a crear Tabla Empleados con la siguiente estructura:

Nombre_columna	Tipo_dato
Codigo	int (llave primaria)
Nombres	varchar(25)

Apellidos	varchar(25)
Telefono	varchar(9)

create table Empleados
(Codigo int primary key,
Nombre varchar(25),
Apellidos varchar(25),
Telefonovarchar(9)
)

Nota: Para ver las tablas creadas en la base de datos digitar: show tables;

🛤 MySQL Command Line Client		- 🗆 X
mysql> create table Emplead lidos varchar(25), Telefond Query OK, Ø rows affected (dos (Codigo int primary key, Nombres varchar(25), o varchar(9)); (0.09 sec)	Ape 1
mysql> show tables;		
++ Tables_in_guia4_lpiii		
tt l empleados l		
++ 1 row in set (0.01 sec)		
mysql>		

5. Para ver la estructura de la tabla debe digitar: describe empleados;

sql> descr	ibe empleados;	•	•		··
Field	Туре	Null	Кеу	Default	Extra
Codigo Nombres Apellidos Telefono	int(11) varchar(25) varchar(25) varchar(25) varchar(9)	I NO VES VES VES VES	PRI	NULL NULL NULL	

6. Agregar los siguientes datos:

1	Roberto Mario	Rodríguez	2589-8585
2	Maria Gabriela	Carranza	7895-7858
3	José Fernando	Martínez	2698-4576

Para agregar los datos debemos utilizar transacciones SQL, en este caso se debe usar la cláusula **insert**, así como se muestra en la siguientefigura:



7. Para seleccionar los datos de la tabla debe realizar una consulta select. En este caso se quiere seleccionar toda la información de la tabla, se debe digitar: select * from empleados;

sql> se	lect * from emplea	ados;		
Codigo	Nombres	Apellidos	Telefono	
1 2 3	Roberto Mario Maria Gabriela Jose Fernando	Rodriguez Carranza Martinez	2589-8585 7895-7858 2698-4576	

¿Tiene otro MySQLi nstalado en su equipo en otro servicio como XAMP?

XAMP es un aplicativo que incluye otros servicios como Apache,MySQL y PHP. La administración de MySQL puede realizarse desde una interfase de aplicación PHP incluida (<u>http://localhost/phpmyadmin/</u>)



Luego continuar desde el paso 3 de este ejercicio, debe examinar las opciones de la aplicación para lograr crear la base de datos y las tablas.

¿Se conectará a un servidor MySQL instalado en otro equipo?

Debe conocer la IP y el puerto del servidor MySQL.Este será el caso más habitual en un entorno de trabajo real. Consulte a su instructor para más detalles.

Luego continuar desde el paso 3 de este ejercicio, utilizará la consola de comandos SQL de NetBeans para este caso.

Ejercicio 2. Conexión de NetBeans IDE a una base de datos

Para conectar una aplicación a una base de datos, se requiere:

- Instalar en NetBeans IDE el conector de la base de datos SOLO SI NO LO TIENE INSTALADO
- Establecer la conexión entre NetBeans IDE y la base de datos

Instalación en NetBeans IDE del conector de la Base de Datos (MySQL)

El procedimiento a seguir es el siguiente:

1. Hacer clic en la opción **Services** del proyecto y expanda los nodos **Databases** y **Drivers** para los conectores disponibles y las conexiones de las Base de Datos.



2. Para agregar el conector de MySQL, hacer clic derecho en el nodo **Drivers** y seleccione la opción **New Driver** del menú contextual.



3. Aparece un cuadro de dialogo para agregar un conector, hacer en el botón Add

New JDBC	Driver 🔀
<u>D</u> river File(s):	Add
Driver ⊆lass: Name:	Eind
	OK Cancel

4. Seleccionar el conector, seleccionar el conector y hacer clic en el botón Abrir.

Select Driver				X		
Buscar en: 🛅 myso	ql-connector-java-5.1.6	~	ø 🕫			
a docs						
C src mysgl-connector-java-5.1.6-bin						
<u>N</u> ombre de archivo:	mysql-connector-java-5.1.6-bin.jar			Abrir		
Archivos de <u>t</u> ipo:	Archive Files (*.jar, *.zip)		*	Cancelar		

5. Aparece la información del conector seleccionado. Para confirmar hacer clic en el botón OK.

Driver File(s):	C:\Documents and Settings\Compaq_Propietario\Mis
	Zeinve
	< <u>></u>
Driver <u>⊂</u> lass:	com.mysql.jdbc.Driver 💌 Eind
<u>N</u> ame:	MySQL (Connector/J driver) (1)

6. Se agrega un nuevo nodo para el conector de MySQL en la opción Services



Establecer la conexión entre NetBeans IDE y la base de datos (ESTOS PASOS SÍ LOS REALIZARÁ)

1. Hacer clic derecho sobre el nodo del conector a MySQL



- 2. Seleccionar la opción **ConnectUsing...**, del menú contextual, aparece un cuadro de dialogo para establecer una nueva conexión, aparece una ventana como se muestra a continuación: Los datos que deben estar en esta ventana son:
 - a. En la opción Name: MySQL (Connector/J driver)
 - b. En la opción Driver: com.mysql.jdbc.Driver
 - c. Establecer en la opcionDatabase URL el siguiente formato:

jdbc:mysql://servidor:puerto/baseDatos Por ejemplo: jdbc:mysql://localhost:3306/mysql

Donde **servidor** es la dirección IP (o nombre de dominio del servidor), en caso que el servidor este en la misma computadora que NetBeans utiliza el nombre localhost, **puerto**, es el puerto 3306 del servidor, si el servidor utiliza el puerto predefinido se puede omitir, **baseDatos**, es la base de datos a la que se desea conectar. Establecer el nombre del usuario (Username) y la contraseña (Password), para acceder a la base de datos, presionar el botón **OK**.

New Conne	ection Wizard X
Customize Con	nection
Driver Name:	MySQL (Connector/J driver)
Host:	localhost Port: 3306
Database:	Guia4
User Name:	root
Password:	
	Remember password
	Connection Properties Test Connection
JDBC URL:	jdbc:mysql://localhost:3306/Guia4?zeroDateTimeBehavior=convertToNull

3. Aparece un cuadro de dialogo confirmando que se estableció la conexión. Para aceptar hacer clic en el botón **OK**.

🗊 New Database	Connection		
Basic setting Advance	:ed		
Select a database so	hema to use.		
∑elect schema: <\\	o schema>	<u> </u>	Get Schemas
Connection establish	ied.		
		OK Cancel	

4. Para confirmar los datos, aparece un nuevo nodo con la conexión a la base de datos.

Projects	Files	Serv	40 ×		
🗸 🗐 Databases					
📓 Java DB					
ᅇ Java DB (Embedded)					
🥮 Java DB (Network)					
🥮 JDBC-ODBC Bridge					
🌮 Mys	QL (Connector	/J driver)			
MySQL_CONECT					
PostgreSQL					
🔀 jdbc:d	erby://localhost	::1527/san	٦۴		
Þ 🖳 jdbc:m	nysql://localhos	t:3306/Gui	a 🛛		

Acceso a la Base de Datos desde NetBeans IDE

NetBeans nos permite realizar operaciones sobre la base de datos como crear y borrar tablas, agregar, eliminar, actualizar y seleccionar registros que están almacenados en las tablas, para hacer estas operaciones, debe expandir el nodo con la conexión a la base de datos para que aparezcan los objetos creados en la BD (tablas, vistas y procedimientos).

Para ver las tablas creadas en una BD, hacer clic sobre el nodo de la opción **Tables** y para ver las columnas o campos de una tabla hacer clic en (+) en el nombre de cada tabla.

Si quiere crear una tabla nueva en la BD, haga clic derecho sobre el nombre **Table** y seleccionar la opción **CreateTable...**

Services		× B					
🖃 🗐 Databases							
🕀 🌆 MySQL Server at localhost:3306 [root]							
🗎 🕼 Java DB							
🕀 🕖 Drivers							
🗄 🔢 jdbc:derby://loca	lhost:1527/sample [app on	APP]					
📄 🗄 🔂 jdbc:mysql://loca	lhost:3306/bdalumnos?zer	oDateTimeBehavio					
jdbc:mysql://loca	lhost:3306/blog [root on De	efault schema]					
jdbc:mysql://loca	lhost:3306/controldeprodue	ctos?zeroDateTim					
jdbc:mysql://loca	lhost:3306/empleados?zer	oDateTimeBehavio					
jdbc:mysql://loca	lhost:3306/guia4?zeroDate	TimeBehavior=co					
📄 🗐 guia4							
📄 🖨 🔁 Tables		1					
🕀 🖽 en	Create Table						
🕀 🗀 Views	Recreate Table						
🕀 📄 Proce							
🕀 🗐 Other da	Execute Command						
idbc:mysql:/		DateTimeBehavio					
jdbc:mysql:/	Refresh	TimeBehavior=co					
jdbc:mysql:/,,,,,,		ones?zeroDateTir					

Se abre una nueva ventana donde se escribe los nombres de las columnas de la tabla, el tipo de dato, si tiene llave primaria, para agregar mas columnas hacer clic en **Addcolumn**, si quiere eliminar una columna hacer clic en **Remove.** Al terminar hacer clic en el botón **OK**.

Cr	Create Table							
Iable name: Alumno Owner:								
Key	Index	Null	Unique	Column name	Data type	Size	A <u>d</u> d column	
	V		~	Cod_alumno	INT	0	Remove	
		~		Apellidos	VARCHAR	15		
		 Image: A set of the set of the		Nombres	VARCHAR	15		
<						>		
						ОК	Cancel	

Para ejecutar un comando ya sea de inserción, eliminación, actualización y selección de información, seleccionar la opción **ExecuteCommand...**

Se abrirá en Netbeans el área de edición de comandos o instrucciones SQL, donde digitamos los comandos. En la parte superior de la ventana se encuentran las transacciones y en la parte inferior muestra el resultado de la transacción. Y finalmente podemos manipular la información de una tabla



Para ejecutar la transacción y ver los resultados, se hace clic en el icono Run SQL.

NetBeans IDE 8.2	-		×
File Edit View Navigate Source Refactor Run Debug Profile Team Tools Window Help	Q Search (Ctrl+I)		
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1			
🕫 Services × Projects - 🐻 Output - SQL 4 execution × Start Page × 🗃 SQL 4 [jdbc:mysql://localhost:33] ×			-
Databases MySQL Server at localhost: 3306 [root] Image: Server at localhost: 1527/sample [appe: Server at localhost: 3306/bdlumnos: Image: Server at localhost: 3306/bdlumnos: <p< td=""><td></td><td><mark>, - Q</mark></td><td>→</td></p<>		<mark>, - Q</mark>	→
Procedures Select * from empleados ×			
K Max, rows: 100 Fetched Rows: 12	Matching Rows:		
Navigator X		<u> </u>	
Members <pre> <empty> < </empty></pre> # id nombres apellidos	1	telefono	E₽
Conexion 1 1Rafael Torres	2	2292-6599	^
dose(ResultSet rs)	2	2292-3180	
dose(PrenaredStatement stmt)	2	2292-6588	

También podemos ver los datos de una tabla seleccionando la opción **View Data**, la cual la seleccionamos haciendo clic derecho sobre el nombre de la tabla la cual queremos ver los datos.

🚊 🖳 jdbc:mysql://localhost:3306/guia4?zeroDateTimeBehavior=convertToNull [
🖨 🗐 guia4				
🖨 🧰 Tables				
🗄 ··· 🛄 empleados			1	
🕀 💼 Views	View Data			
😥 🧰 Procedures	Execute Command			
🗄 🗐 Other database:				
🗄 🛐 jdbc:mysql://localho	Add Column		=convertTo	
🗄 🛐 jdbc:mysql://localho	Refresh		/ertToNull	
🗄 🛐 jdbc:mysql://localho	Delete	Suprimir	Behavior =	
🖶 🕵 jdbc:mysql://localho			:Behavior =	
🗄 🔣 jdbc:mysql://localho	Grab Structure		rtToNull [rc	
idbc:mysql://localho	Recreate Table		=convertT	
🗄 🗐 🧟 Web Services				
Servers	Properties			

Acceder a los datos de una BD desde una aplicación Java

Agregue la librería de MySQL a las rutas antes mencionadas

C:\Archivos de programa\Java\jre1.6.0_05\lib\ext Ó C:\Program Files\Java\jdk1.6.0_25\jre\lib\ext



También puede agregar la librería directamente al proyecto

- 1. En el proyecto Guia4 debe crear una clase y colocar el nombre VerEmpleado, dentro del paquete "sv.edu.udb.empleado".
- 2. Digitar el siguiente código:

```
import java.sql.*;
public class VerEmpleado {
public VerEmpleado()
    {
    // Se utiliza un try por los posibles errores de MySQL
try
    {
    // Se utiliza un try por los posibles errores de MySQL
try
    {
    // obtenemos el driver de para mysql
    Class.forName("com.mysql.jdbc.Driver");
    // Se obtiene una conexión con la base de datos.
Connection conexion = DriverManager.getConnection (
    "jdbc:mysql://localhost/Guia4","root", "rafael");
    // IMPORTANTE: EL CAMPO PASSWORD POR DEFECTO DEBE IR EN BLANCO,
    // DEBE ASIGNAR EL PASSWORD CORRECTO
    // Permite ejecutar sentencias SQL sin parámetros
```

```
Statement s = conexion.createStatement();
// Contiene la tabla resultado de la pregunta SQL que se haya realizado
ResultSet rs = s.executeQuery ("select * from Empleados");
//Se recorre el ResultSet, mostrando por pantalla los resultados.
while (rs.next())
       {
/Podemos mostrar los datos de otra forma ver mas abajo e la guia.
System.out.println ("ID: "+rs.getInt(1)+
"\nNombre: "+rs.getString(2)+
"\nApellidos: "+rs.getString(3)+
"\nTelefono: "+rs.getString(4));
}
// Se cierra la conexión con la base de datos. NUNCA OLVIDE CERRARLA
conexion.close();
    }
catch (ClassNotFoundException e1) {
//Error si no puedo leer el driver de MySQL
                 System.out.println("ERROR:No encuentro el driver de la BD:
"+e1.getMessage());
    }
catch (SQLException e2) {
//Error SQL: login/passwdó sentencia sqlerronea
                 System.out.println("ERROR:Fallo en SQL: "+e2.getMessage());
    }
  }
/**
  * Método principal, instancia una clase PruebaMySQL
  * @paramargs the command line arguments
public static void main(String[] args)
{
newVerEmpleado();
  }
  }
```

3. Ejecutar la aplicación y obtendremos los siguientes resultados:

```
Output - Clase4 (run)
\square
    run:
    ID: 1
\square
    Nombre: Rafael
    Apellidos: Torres
    Telefono: 2292-6599
22
    *******
    TD- 2
    Nombre: Manuel
    Apellidos: Orellana
    Telefono: 2292-3180
    ******
    ID: 3
    Nombre: Glenda
    Apellidos: Martinez
    Telefono: 2292-6588
    ********
```

4. Forma alternativa de mostrar los datos.

Para obtener los distintos atributos de la base se utilizarán, normalmente, los métodos getString(atributo), cuyo parámetro es una cadena con el nombre del atributo que queremos recuperar.

Ejercicio 6: Ingresar datos desde una aplicación a una base de datos

- En el proyecto Guia4_POO1 debe crear una clase y colocar el nombre IngresoDatos, dentro del paquete "sv.edu.udb.empleado".
- 2. Digitar el siguiente código.

```
import java.sql.*;
import javax.swing.JOptionPane;
import sv.edu.udb.util.* ;
public class IngresoDatos {
  private int id;
  private String ids;
  private String nombre;
  private String apellido;
  private String telefono;
private Connection conexion;
```

```
private ResultSet rs;
private Statement s:
public IngresoDatos()
  ł
// Se utiliza un try por los posibles errores de MySQL
try
//obtenemos el driver de para mysql
Class.forName("com.mysql.jdbc.Driver");
// Se obtiene una conexión con la base de datos.
conexion = DriverManager.getConnection (
"jdbc:mysql://localhost/Guia4","root", "rafael");
        IMPORTANTE: EL CAMPO PASSWORD DEBE IR EN BLANCO POR DEFECTO,
             DEBE ASIGNAR EL PASSWORD CORRECTO
// Permite ejecutar sentencias SOL sin parámetros
       s = conexion.createStatement();
//Metodo para ingresar valores
       ingreso();
s.executeUpdate("Insert into Empleados
values("+id+",\""+nombre+"\",\""+apellido+"\",\""+telefono+"\")");
JOptionPane.showMessageDialog(null,"Persona Ingresada Correctamente");
catch (ClassNotFoundException e1) {
//Error si no puedo leer el driver de MySQL
        System.out.println("ERROR:No encuentro el driver de la BD: "+e1.getMessage());
System.exit(0);
     }
catch (SQLException e2) {
//Error SQL: login/passwdó sentencia sql erronea
        System.out.println("ERROR:Fallo en SQL: "+e2.getMessage());
System.exit(0);
    }
  }
public void ingreso()
  {
    ids=JOptionPane.showInputDialog("Ingrese el ID");
    id=Integer.parseInt(ids);
nombre=JOptionPane.showInputDialog("Ingrese el Nombre");
    apellido=JOptionPane.showInputDialog("Ingrese el Apellido");
telefono=JOptionPane.showInputDialog("Ingrese su Telefono");
      do {
            if (MatchTelephone.compareTelephone(telefono) == TRUE)
            {
              break:
            }else{
              JOptionPane.showMessageDialog(null, "Numero de Telefono invalido");
              telefono=JOptionPane.showInputDialog("Ingrese su Telefono");
            }
```

```
} while (true);
}
public void mostrardatos() throws SQLException{
rs = s.executeQuery ("select * from Empleados");
while (rs.next())
       Ł
JOptionPane.showMessageDialog(null,"ID: "+rs.getString("Codigo")+
"\nNombre: "+rs.getString("Nombres")+
"\nApellidos: "+rs.getString("Apellidos")+
"\nTelefono: "+rs.getString("Telefono")
          );
       }
  }
public void cierreconexion() throws SQLException{
// Se cierra la conexión con la base de datos.
           if (conexion != null){
              conexion.close();
            }
  }
public static void main(String[] args) throws SQLException
IngresoDatosing=newIngresoDatos();
ing.mostrardatos();
ing.cierreconexion();
  }
  }
```

 Crear la clase llamada MatchTelephone dentro del paquete "sv.edu.udb.util" y agregar el siguiente código, esta clase utilizara expresiones regulares para validar el formato de un número de teléfono.

package sv.edu.udb.util;

```
// compareTelephone("7123-4444");
// }
public static boolean compareTelephone (String telefono) {
    String expression="(2|7)\\d{3}-\\d{4}";
    Pattern pat = Pattern.compile(expression);
    Matcher mat = pat.matcher(telefono);
    if (mat.matches()) {
        System.out.println("SI");
        return TRUE;
    }
    return FALSE;
}//Cierre del main
```

4. Para este ejemplo se realizara una análisis que nos permita comprender que no es lo mismo manejar datos "Vacios o Nulos", en primer lugar crearemos la clase Conexión para que este en un paquete diferente "sv.edu.udb.util" y solo la invoquemos para futuros desarrollos.

```
package sv.edu.udb.util;
/**
*
* @author Rafael Torres
*/
import java.sql.*;
public class Conexion {
   private Connection conexion =null;
   private Statement s =null;
   private ResultSet rs=null;
   private String ingresoempleados="";
  //Contructor
  public Conexion() throws SQLException{
     try
     {
       //obtenemos el driver de para mysql
       Class.forName("com.mysql.jdbc.Driver");
       // Se obtiene una conexión con la base de datos.
       conexion = DriverManager.getConnection (
          "jdbc:mysql://localhost/guia4","root", "rafael");
       // Permite ejecutar sentencias SQL sin parámetros
        s = conexion.createStatement();
        System.out.println("Conexion Exitosa");
     }
    catch (ClassNotFoundException e1) {
```

```
//Error si no puedo leer el driver de MySQL
              System.out.println("ERROR:No encuentro el driver de la BD:
"+e1.getMessage());
    }
  }
  //Metodo que permite obtener los valores del resulset
  public ResultSet getRs() {
    return rs;
  }
  //Metodo que permite fijar la tabla resultado de la pregunta
  //SQL realizada
  public void setRs(String sql) {
    try {
      this.rs = s.executeQuery(sql);
    } catch (SQLException e2) {
         //Error SQL: login/passwd ó sentencia sql errónea
              System.out.println("ERROR:Fallo en SQL: "+e2.getMessage());
    }
  }
  //Metodo que recibe un sql como parametro que sea un update,insert.delete
  public void setQuery(String sql) throws SQLException {
    this.s.executeUpdate(sql);
  }
  //Metodo que cierra la conexion
  public void cerrarConexion() throws SQLException{
    conexion.close();
  }
```

5. Crear la clase llamada "InsertNulos" dentro del paquete "sv.edu.udb.nulos" y agregar el siguiente código.

```
package sv.edu.udb.nulos;

import java.sql.ResultSet;

import java.sql.SQLException;

import sv.edu.udb.util.Conexion;

/**

*

@ author Rafael Torres

*/

public class InsertNulos {
```

```
/**
   * @param args the command line arguments
   */
  public static void main(String[] args) throws SQLException {
    // TODO code application logic here
    Conexion con = new Conexion();
//
      String sql = "insert into empleados values(7,", 'Torres', null)";
//
    //con.setQuery(sql);
    String sql = "select nombres from empleados ";
    ResultSet rs;
    con.setRs(sql);
    rs= con.getRs();
    String nombre;
     while (rs.next()){
       nombre=rs.getString(1);
       if (nombre == null){
          System.out.println("Nombre 'Null': " + nombre);
       }else if(nombre.equals("")){
          System.out.println("Nombre Vacio: " + nombre);
       }else{
         System.out.println("Nombre Con Datos: " + nombre);
       }
     }
     con.cerrarConexion();
```

- 6. Ejecutar un par te sentencias update para la tabla empleados.
 - insert into empleados values(5,",'Torres',null);
 - insert into empleados values(6,null,Rodriguez,null);
- 7. Ejecutar la clase InsertNulos para ver un resultado como el de la siguiente imagen

```
Output - Clase4 (run) × HTTP Server Monitor
\supset
     run:
     Conexion Exitosa
\square
     Nombre Con Datos: Rafael
Nombre Con Datos: Manuel
     Nombre Con Datos: Glenda
82
     Nombre Con Datos: Elena
     Nombre Vacio:
     Nombre 'Null': null
     Nombre Vacio:
     BUILD SUCCESSFUL (total time: 0 seconds)
     .
```

IV. EJERCICIOS COMPLEMENTARIOS

Crear las siguientes tablas en MySql:

Tabla alumno

Nombre Columna	Tipo de dato
Cod_alumno	int primary key
Nombre	varchar(80)
Apellido	varchar(80)
Edad	int
Direccion	varchar(100)

Tabla materia

Nombre Columna	Tipo de dato
Cod_materia	int primary key
Nombre	varchar(25)
Descripción	varchar(100)

Tabla alumno_materia

Nombre Columna	Tipo de dato
Cod_alumno	int (llave foránea)
Cod_materia	int (llave foránea)

Crear el CRUD a las 3 tablas anteriores:

- Alumno
- ✤ Materia
- ✤ Alumno_materia

Reportes

- Crear una aplicación donde muestre la información de cada una de las tablas anteriores, para ello crear tres métodos diferentes que serán invocados según la necesidad.
- 2. Mostrar las materias que está cursando un alumno específico.

Tip: Usará la cláusula where en la sentencia SQL

Nota: Para cada ejercicio solicitado utilizar JoptionPane y validar los datos. Debe usar una clase pre-fabricada para validar.