

Codificación de módulos del software según requerimientos del proyecto

Codificación de módulos del software según requerimientos del proyecto GA7-

220501096-AA2-EV01

Aprendiz:

Juan Sebastián Blanco Acevedo

Tecnólogo en análisis y desarrollo de Software

Instructora:

Astrid Maritza Calvache Chicangana

Servicio Nacional de Aprendizaje – SENA

Análisis y desarrollo de software ADSO

2025

Codificación de módulos del software según requerimientos del proyecto

Introducción

En este proyecto se implementó la conexión entre Java (usando NetBeans IDE) y MySQL Workbench, con el objetivo de crear, gestionar y manipular datos de una base de datos relacionada con la tienda virtual *PC Innovation*.

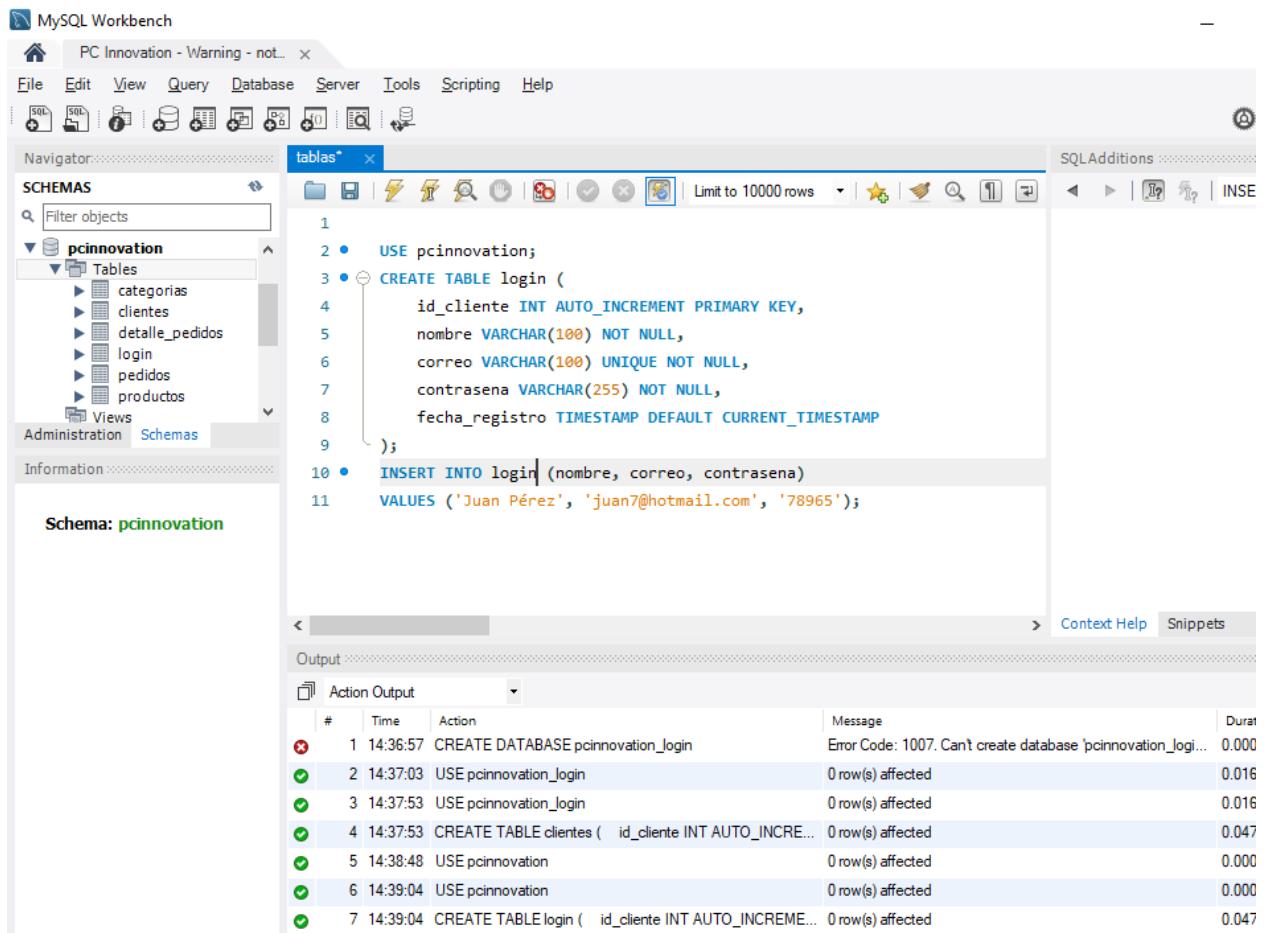
Durante el proceso se diseñó una base de datos con sus respectivas tablas, en las cuales se almacenan los datos de los usuarios y clientes. Posteriormente, se desarrolló un programa en Java que permite establecer conexión con el servidor de MySQL mediante el uso del Driver JDBC, ejecutar sentencias SQL de tipo INSERT y SELECT, y visualizar la información registrada directamente desde la aplicación.

Codificación de módulos del software según requerimientos del proyecto

Desarrollo de la Actividad

Una vez instalado el software iniciare con la creación del a base datos que voy a usar para la actividad, en este caso voy a crear dentro de la base de datos de mi tienda virtual, una tabla llamada “login, que es desde donde se van a almacenar los usuarios y contraseñas de mis clientes.

Para esto desde Workbench inicio con la creación de la tabla:



The screenshot shows the MySQL Workbench interface. In the top-left pane, the 'SCHEMAS' section is open, showing the 'pcinnovation' schema with its tables: categorias, clientes, detalle_pedidos, login, pedidos, and productos. The 'Tables' section under 'pcinnovation' is also visible. The main workspace shows a SQL editor with the following code:

```
1
2 • USE pcinnovation;
3 • CREATE TABLE login (
4     id_cliente INT AUTO_INCREMENT PRIMARY KEY,
5     nombre VARCHAR(100) NOT NULL,
6     correo VARCHAR(100) UNIQUE NOT NULL,
7     contrasena VARCHAR(255) NOT NULL,
8     fecha_registro TIMESTAMP DEFAULT CURRENT_TIMESTAMP
9 );
10 • INSERT INTO login (nombre, correo, contrasena)
11     VALUES ('Juan Pérez', 'juan@hotmail.com', '78965');
```

The 'Output' pane at the bottom displays the execution log:

#	Time	Action	Message	Durat
1	14:36:57	CREATE DATABASE pcinnovation_login	Error Code: 1007. Can't create database 'pcinnovation_login'	0.000
2	14:37:03	USE pcinnovation_login	0 row(s) affected	0.016
3	14:37:53	USE pcinnovation_login	0 row(s) affected	0.016
4	14:37:53	CREATE TABLE clientes (id_cliente INT AUTO_INCRE...	0 row(s) affected	0.047
5	14:38:48	USE pcinnovation	0 row(s) affected	0.000
6	14:39:04	USE pcinnovation	0 row(s) affected	0.000
7	14:39:04	CREATE TABLE login (id_cliente INT AUTO_INCRE...	0 row(s) affected	0.047

Codificación de módulos del software según requerimientos del proyecto

Ahora procedo a ingresar los valores de los datos de mis clientes a la tabla

The screenshot shows the MySQL Workbench interface. In the top left, the 'Navigator' pane displays the database schema with 'pedidos', 'productos', and 'usuarios' tables. The 'usuarios' table is expanded, showing columns: 'id_cliente', 'correo', and 'contrasena'. The 'SQL' tab contains a query window with the following SQL code:

```
1 • select * from usuarios;
2 • INSERT INTO usuarios (correo,contraseña)
3 VALUES ('overkill57@hotmail.com','15277');
4 • INSERT INTO usuarios (correo,contraseña)
5 VALUES ('pepito7@gmail.com','987423');
6 • INSERT INTO usuarios (correo,contraseña)
```

The 'Result Grid' pane shows the data being inserted into the 'usuarios' table:

	id_cliente	correo	contrasena
1		Steve27@gmail.com	152777
2		overkill57@hotmail.com	111247
3		pepito7@gmail.com	987423
4		juabsz@gmail.com	781473
5		alejandra1@gmail.com	715157
*	NULL	NULL	NULL

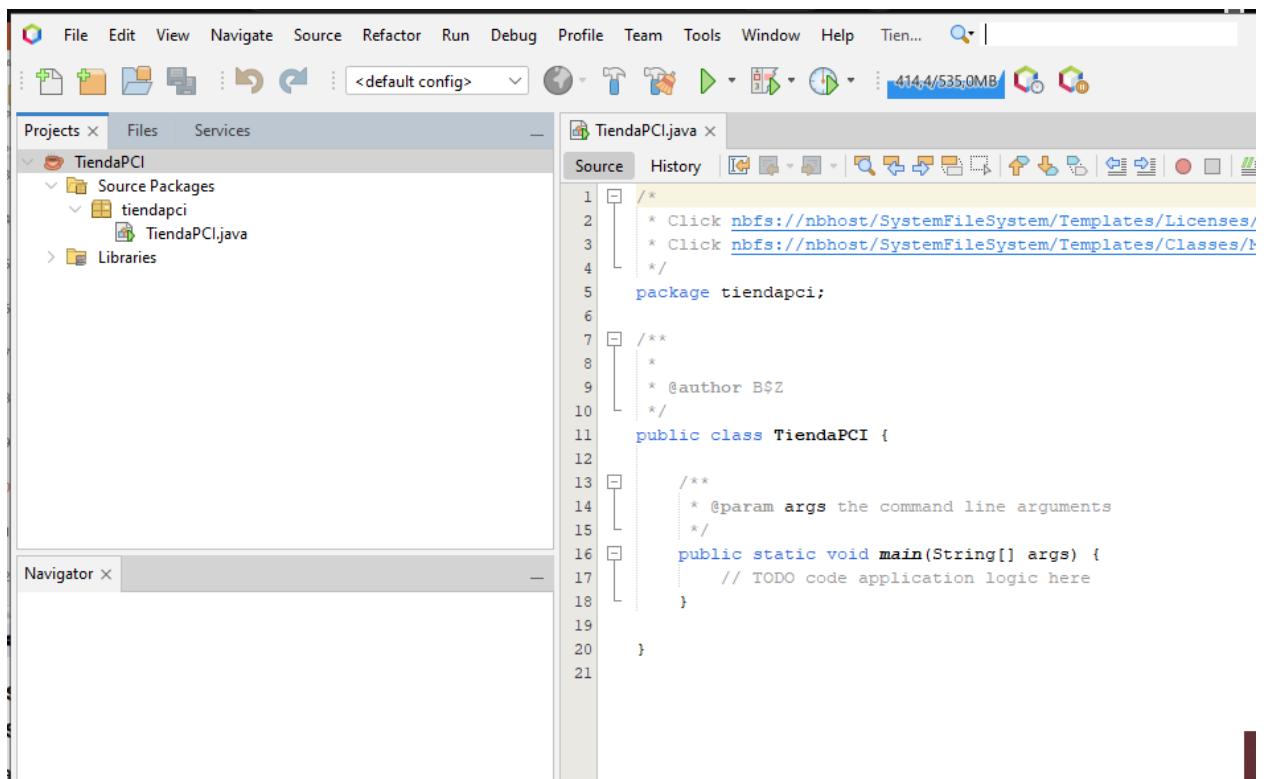
The 'Output' pane at the bottom shows the execution log:

#	Time	Action	Message	Duration / I
31	15:01:13	insert into usuarios (correo,contraseña) VALUES (pepito...)	1 row(s) affected	0.031 sec
32	15:01:36	insert into usuarios (correo,contraseña) VALUES (juabsz...)	1 row(s) affected	0.062 sec
33	15:02:46	insert into usuarios (correo,contraseña) VALUES (alejan...)	1 row(s) affected	0.015 sec
34	15:03:02	select * from usuarios LIMIT 0, 10000	5 row(s) returned	0.000 sec /
35	15:06:01	select * from usuarios LIMIT 0, 10000	5 row(s) returned	0.000 sec /
36	15:06:15	select * from usuarios LIMIT 0, 10000	5 row(s) returned	0.000 sec /
37	15:06:44	select * from usuarios LIMIT 0, 10000	5 row(s) returned	0.016 sec /

Como podemos observar ya tengo mi tabla con las 3 columnas diligenciadas con algunos de los clientes y sus datos de ingreso a la tienda virtual.

Codificación de módulos del software según requerimientos del proyecto

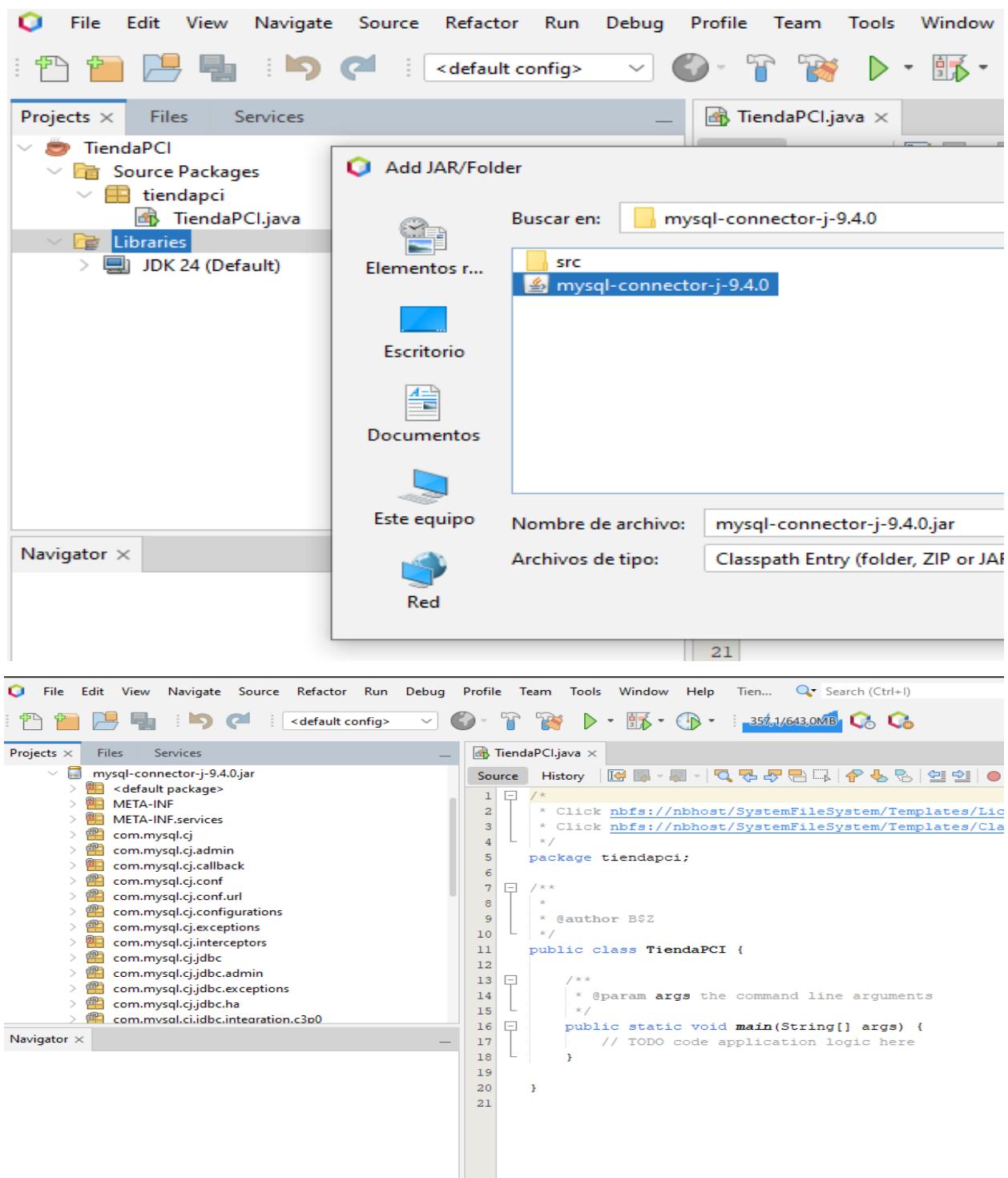
Ahora procedo a abrir la herramienta Netbeans para iniciar con la creación de mi nuevo proyecto el cual nombre: TiendaPCI, una vez creado el proyecto voy a realizar el cargue del driver de JDBC para la creación de las librerías



```
1  /*
2  * Click nbfs://nbhost/SystemFileSystem/Templates/Licenses/
3  * Click nbfs://nbhost/SystemFileSystem/Templates/Classes/-
4  */
5  package tiendapci;
6
7  /**
8  *
9  * @author B$Z
10 */
11 public class TiendaPCI {
12
13     /**
14      * @param args the command line arguments
15     */
16     public static void main(String[] args) {
17         // TODO code application logic here
18     }
19
20 }
21
```

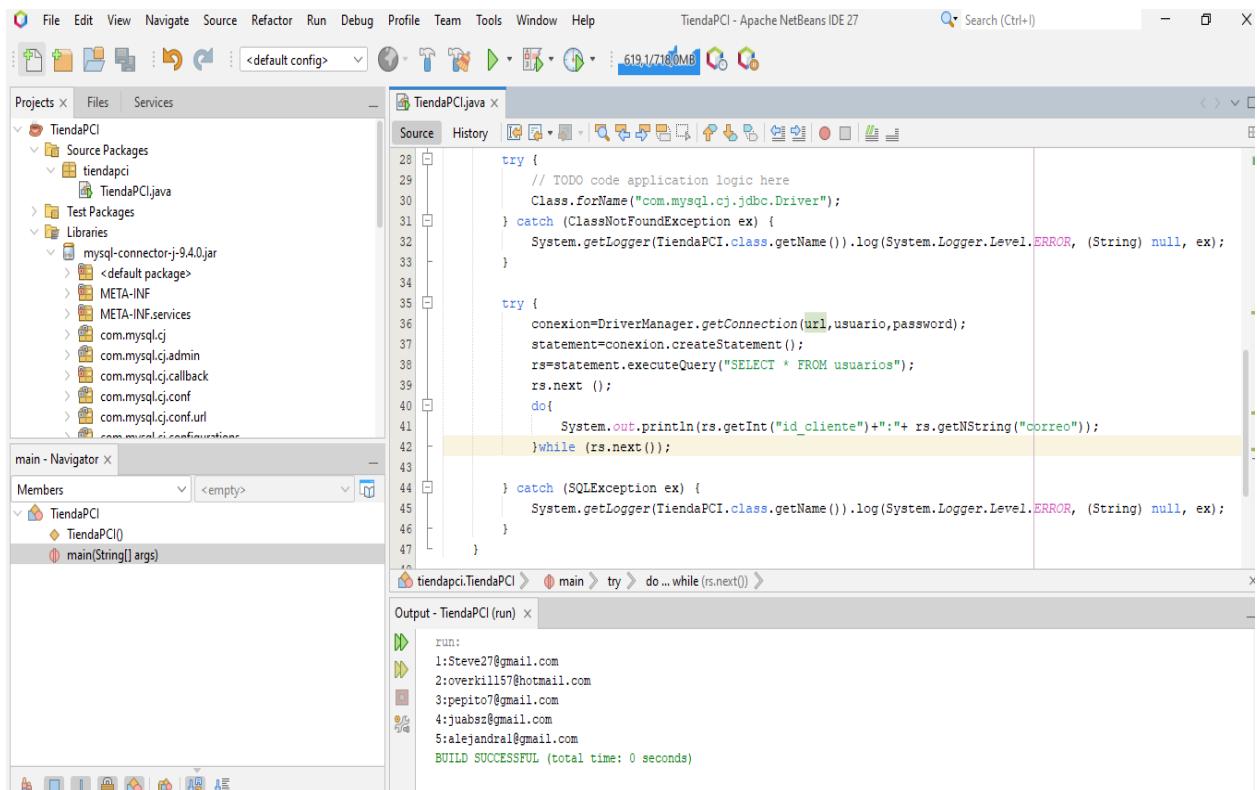
Codificación de módulos del software según requerimientos del proyecto

En la siguiente imagen se puede observar como se selecciono el conector y me genera todo el listado de las librerías



Codificación de módulos del software según requerimientos del proyecto

Luego procedo a configurar la conexión entre el Netbeans y mi base de datos como se observa en la siguiente evidencia:



The screenshot shows the Apache NetBeans IDE 27 interface. The top menu bar includes File, Edit, View, Navigate, Source, Refactor, Run, Debug, Profile, Team, Tools, Window, Help, and a search bar. The title bar says "TiendaPCI - Apache NetBeans IDE 27". The left sidebar has "Projects X", "Files", and "Services" tabs, with "TiendaPCI" selected. Under "Source Packages", there is a "tiendapci" package containing a "TiendaPCI.java" file. Under "Libraries", there is a "mysql-connector-j-9.4.0.jar" entry. The main workspace shows the "TiendaPCI.java" code editor with the following Java code:

```
28     try {
29         // TODO code application logic here
30         Class.forName("com.mysql.cj.jdbc.Driver");
31     } catch (ClassNotFoundException ex) {
32         System.getLogger(TiendaPCI.class.getName()).log(System.Logger.Level.ERROR, (String) null, ex);
33     }
34
35     try {
36         conexion=DriverManager.getConnection(url,usuario,password);
37         statement=conexion.createStatement();
38         rs=statement.executeQuery("SELECT * FROM usuarios");
39         rs.next ();
40         do{
41             System.out.println(rs.getInt("id_cliente")+"-"+ rs.getString("correo"));
42         }while (rs.next());
43
44     } catch (SQLException ex) {
45         System.getLogger(TiendaPCI.class.getName()).log(System.Logger.Level.ERROR, (String) null, ex);
46     }
47 }
```

The code uses JDBC to connect to a MySQL database and print the results of a query to the console. The output window shows the printed results:

```
run:
1:Steve27@gmail.com
2:overkill157@hotmail.com
3:pepit07@gmail.com
4:juabs8@gmail.com
5:alejandral@gmail.com
BUILD SUCCESSFUL (total time: 0 seconds)
```

Ingreso mis credenciales y datos de conexión, puerto, contraseña de mi base de datos , para que se logre abrir el Driver Manager. Como observamos la conexión se logra correctamente y podemos ver los batos de mi tabla en My Sql Workbench como también en Netbeans.

Codificación de módulos del software según requerimientos del proyecto

Insertar nuevo dato en la tabla: INSERT

The screenshot shows an IDE interface with two main panes. The top pane displays the code for a Java class named `TiendaPCI.java`. The code includes a `try` block that performs a database query to select all users from the `usuarios` table, prints their `id_cliente` and `correo` to the console, and then inserts a new user with `id_cliente = 6`, `correo = 'ingridl77@hotmail.com'`, and `telefono = '152424'`. The bottom pane shows the `Output - TiendaPCI (run)` window, which displays the printed output:

```
5:alejandral@gmail.com
1:Steve27@gmail.com
2:overkill157@hotmail.com
3:pepito7@gmail.com
4:juabsz@gmail.com
5:alejandral@gmail.com
6:ingridl77@hotmail.com
BUILD SUCCESSFUL (total time: 1 second)
```

Codificación de módulos del software según requerimientos del proyecto

MySQL Workbench

PC Innovation - Warning - not... ×

File Edit View Query Database Server Tools Scripting Help

Navigator: tablas* ×

SCHEMAS

Filter objects

- libreta
- pc
- pci
- pcinnovation
 - Tables
 - categorias
 - clientes

Administration Schemas

Information

Connection Details

Name: PC Innovation
Host: localhost
Port: 3306
Login User: root
Current User:root@localhost
SSL cipher: SSL not used

Server

Product: mariadb.org binary
Version: 10.4.32-MariaDB

Connector

Version: C++ 9.4.0

tablas* ×

1 • select * from usuarios

2

3

Result Grid | Filter Rows: Edit:

	id_cliente	correo	contrasena
1	Steve27@gmail.com	152777	
2	overkill57@hotmail.com	111247	
3	pepito7@gmail.com	987423	
4	juabsz@gmail.com	781473	
5	alejandra1@gmail.com	715157	
6	ingrid177@hotmail.com	152424	
NULL	NULL	NULL	

usuarios 3 ×

Action Output

#	Time	Action
1	15:43:52	select * from usuarios LIMIT 0, 10000
2	16:17:01	select * from usuarios LIMIT 0, 10000
3	16:24:29	select * from usuarios LIMIT 0, 10000

Codificación de módulos del software según requerimientos del proyecto

Actualización de un dato existente con UPDATE

The screenshot shows an IDE interface with a code editor and a run output window.

Code Editor:

```
52
53     //UPDATE
54
55     statement.executeUpdate("UPDATE usuarios SET correo='daniela75@gmail.com' WHERE id_cliente=3");
56     System.out.println("");
57     rs=statement.executeQuery("SELECT * FROM usuarios");
58     rs.next ();
59     do{
60         System.out.println(rs.getInt("id_cliente")+": "+ rs.getString("correo"));
61     }while (rs.next());
62
63
64     catch (SQLException ex) {
65         System.getLogger(TiendaPCI.class.getName()).log(System.Logger.Level.ERROR, (String) null, ex);
66     }
67
68
69
70
```

Output Window:

```
tiendapci.TiendaPCI > main > try > do ... while (rs.next()) >
Output - TiendaPCI (run) >
1:Steve27@gmail.com
2:overkill57@hotmail.com
3:daniela75@gmail.com
4:juabsz@gmail.com
5:alejandral@gmail.com
6:ingrid177@hotmail.com
BUILD SUCCESSFUL (total time: 1 second)
```

Codificación de módulos del software según requerimientos del proyecto

The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it, a query editor window displays the following SQL code:

```
1 • select * from usuarios
2 -
3
```

Below the query editor is a result grid titled "Result Grid". It contains the following data:

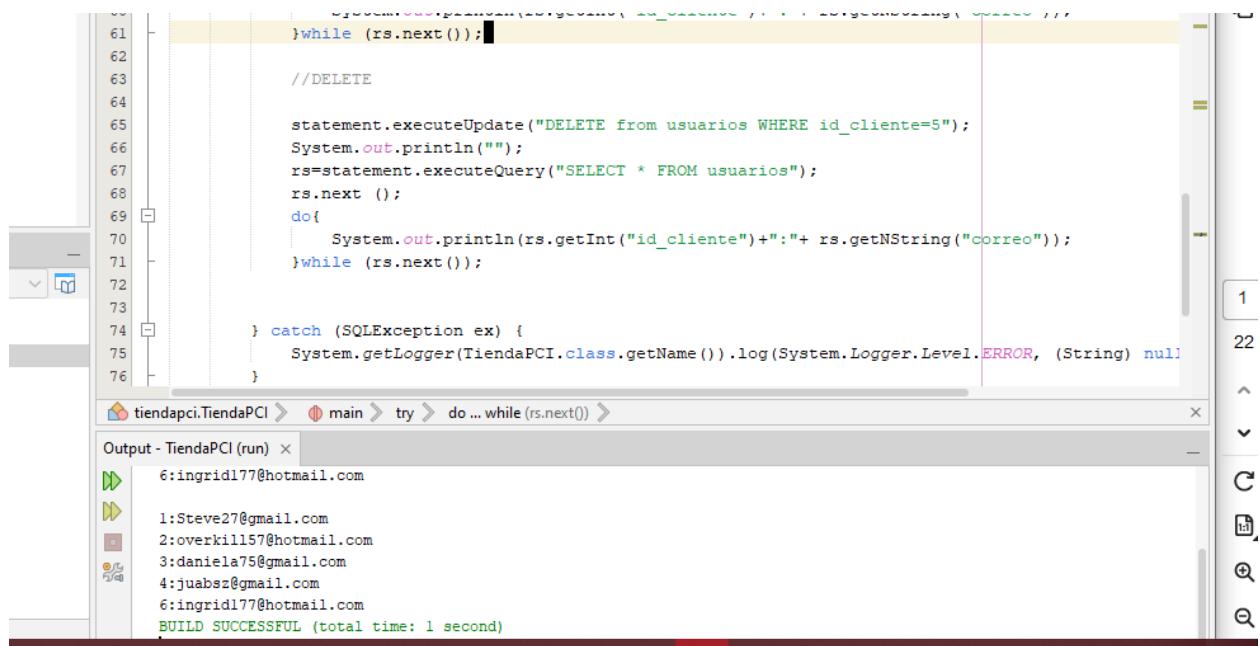
	id_cliente	correo	contrasena
▶	1	Steve27@gmail.com	152777
	2	overkill57@hotmail.com	111247
	3	daniela75@gmail.com	987423
	4	juabsz@gmail.com	781473
	5	alejandra1@gmail.com	715157
	6	ingrid177@hotmail.com	152424
	NULL	NULL	NULL

The row for id_cliente 3 has been selected. The status bar at the bottom shows "usuarios 5".

Como se puede observar el ID numero 3 que fue el seleccionado fue modificado de pepito7@gmail.com al correo daniela75@gmail.com, de esta manera se realiza actualiozacion de datos y podemos ver el cambio tanto en MySql como en NeatBeans

Codificación de módulos del software según requerimientos del proyecto

Y finalizamos con el comando DELETE



The screenshot shows an IDE interface with a code editor and an output window. The code editor contains Java code that includes a try block with a catch clause for SQLException. The catch block logs an error message. The main part of the code uses a while loop to execute an UPDATE statement on the 'usuarios' table where id_cliente=5, followed by a SELECT query to print all user records. The output window shows the results of the SELECT query, listing six users with their IDs and emails. The last user listed is 'ingrid177@hotmail.com'.

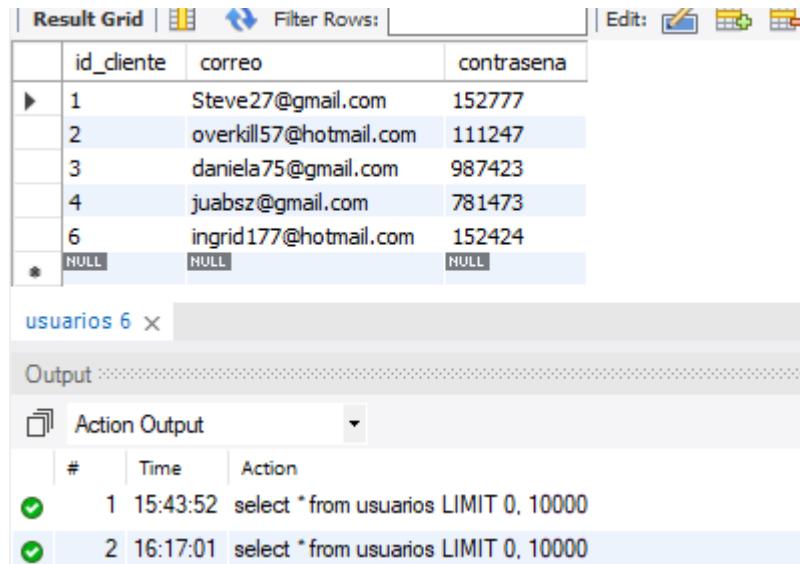
```
61 }while (rs.next());
62
63 //DELETE
64
65 statement.executeUpdate("DELETE from usuarios WHERE id_cliente=5");
66 System.out.println("");
67 rs=statement.executeQuery("SELECT * FROM usuarios");
68 rs.next();
69 do{
70     System.out.println(rs.getInt("id_cliente")+"-"+ rs.getString("correo"));
71 }while (rs.next());
72
73 } catch (SQLException ex) {
74     System.getLogger(TiendaPCI.class.getName()).log(System.Logger.Level.ERROR, (String) null);
75 }
```

Output - TiendaPCI (run) >

```
6:ingrid177@hotmail.com
1:Steve27@gmail.com
2:overkill57@hotmail.com
3:daniela75@gmail.com
4:juabsz@gmail.com
5:ingrid177@hotmail.com
6:ingrid177@hotmail.com
BUILD SUCCESSFUL (total time: 1 second)
```

Como observamos eliminamos el ID numero 5 el cual correspondía al correo

Alejandra177@hotmail.com, Y también desde MySQL queda eliminado



The screenshot shows the MySQL Workbench interface. At the top, there's a toolbar with various icons. Below it is a 'Result Grid' showing the 'usuarios' table with six rows. The columns are 'id_cliente', 'correo', and 'contrasena'. The rows contain the following data:

	id_cliente	correo	contrasena
▶	1	Steve27@gmail.com	152777
▶	2	overkill57@hotmail.com	111247
▶	3	daniela75@gmail.com	987423
▶	4	juabsz@gmail.com	781473
▶	6	ingrid177@hotmail.com	152424
*	NULL	NULL	NULL

Below the grid is a tab labeled 'usuarios 6 x'. Underneath is an 'Output' section with a dropdown menu set to 'Action Output'. The history shows two actions:

#	Time	Action
1	15:43:52	select * from usuarios LIMIT 0, 10000
2	16:17:01	select * from usuarios LIMIT 0, 10000

Codificación de módulos del software según requerimientos del proyecto

Conclusiones

A través de los ejercicios se logró establecer la conexión entre la aplicación desarrollada en Java y la base de datos MySQL, demostrando la importancia del uso del **Driver JDBC** para permitir la comunicación entre ambos entornos.

Se aplicaron correctamente las sentencias **CREATE DATABASE, USE, CREATE TABLE, INSERT** y **SELECT**, las cuales son fundamentales para la administración y manipulación de datos dentro de un sistema gestor de bases de datos.