Departamento de
Tecnología Digital

Tecsup TECNOLOGÍA CON SENTIDO

PROYECTO FINAL

Base de Datos de Sistema Administrativo Universitario

DOCENTE:

Suárez García, Gonzalo

CURSO:

Base de Datos
2 - C24 - Secciones A-B

Integrantes :

- Suarez Lavado, Pedro Junior
- Chuquirima Alanya, Augusto Benjamin
- Prieto Huiza, Alexis Stephano

Entrega 1

1) Instalación y Configuración de Oracle 11g

Paso 1: Descargar Oracle 11g desde el sitio oficial de Oracle

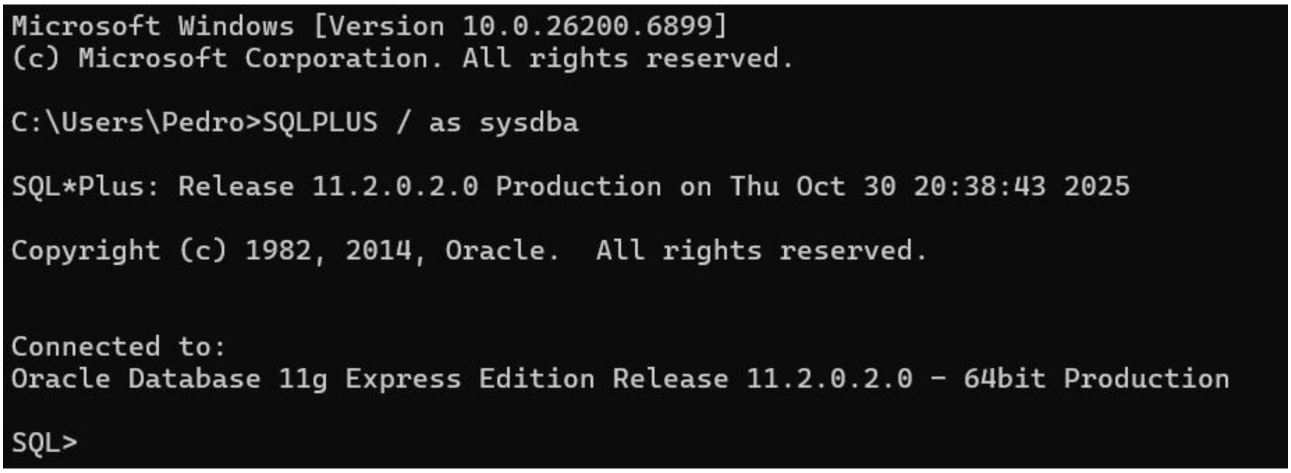
Paso 2: Ejecutar el instalador como administrador

Paso 3: Configuración inicial durante la instalación

1. Seleccionar "Instalar software de base de datos"
2. Elegir "Instalación básica"
3. Especificar directorio de Oracle
4. Configurar contraseña para usuarios SYS y SYSTEM

Paso 4: Conexión a Oracle usando SQL*Plus

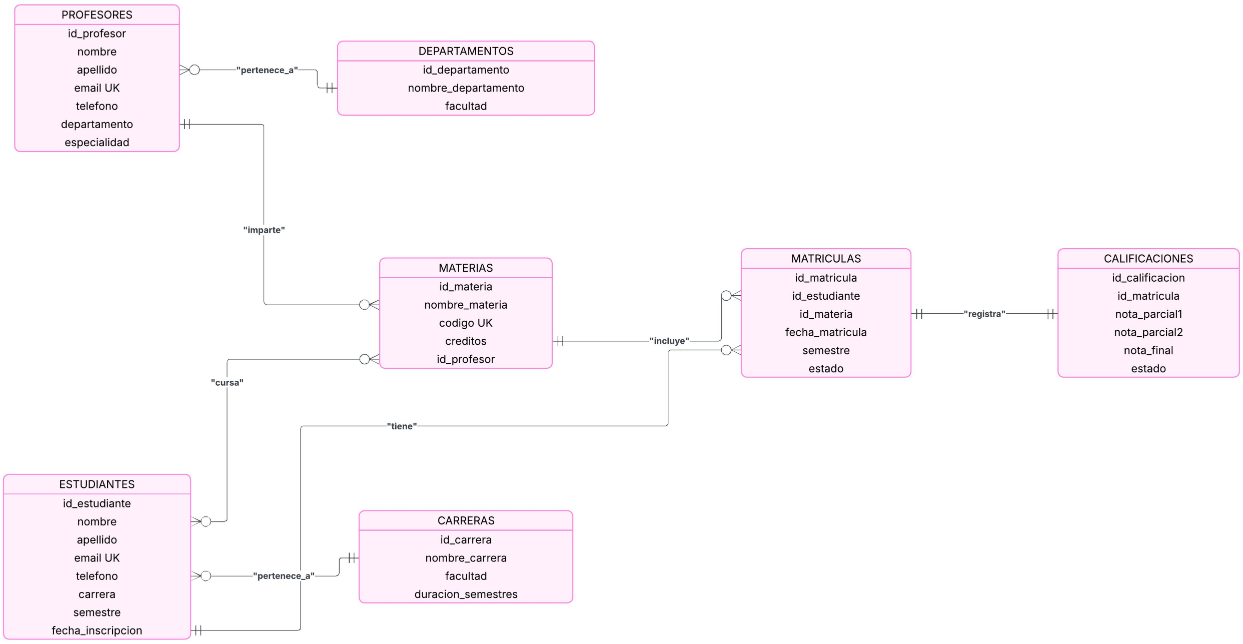
Se usa el comando SQLPLUS / as sysdba para hacer la conexion con Oracle mediante el comando.



```
Microsoft Windows [Version 10.0.26200.6899]
(c) Microsoft Corporation. All rights reserved.

C:\Users\Pedro>SQLPLUS / as sysdba

SQL*Plus: Release 11.2.0.2.0 Production on Thu Oct 30 20:38:43 2025

Copyright (c) 1982, 2014, Oracle.  All rights reserved.


Connected to:
Oracle Database 11g Express Edition Release 11.2.0.2.0 - 64bit Production

SQL>
```

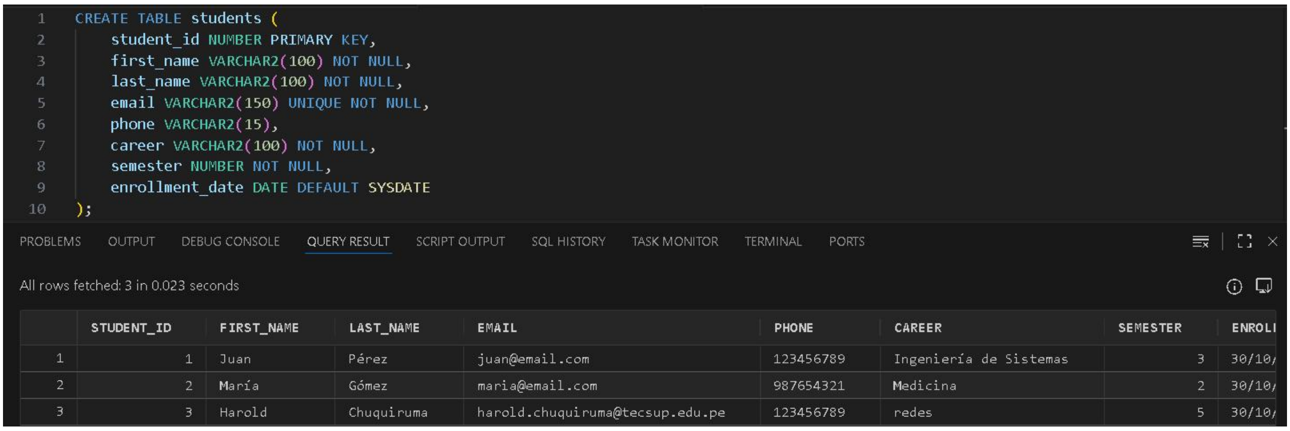
2) El tema escogido sera una Base de Datos de un Sistema Administrativo Universitario.

3) Diseño de esquema básico



4) Creación de tablas

Tabla Students:



```
CREATE TABLE students (
    student_id NUMBER PRIMARY KEY,
    first_name VARCHAR2(100) NOT NULL,
    last_name VARCHAR2(100) NOT NULL,
    email VARCHAR2(150) UNIQUE NOT NULL,
    phone VARCHAR2(15),
    career VARCHAR2(100) NOT NULL,
    semester NUMBER NOT NULL,
    enrollment_date DATE DEFAULT SYSDATE
);
```

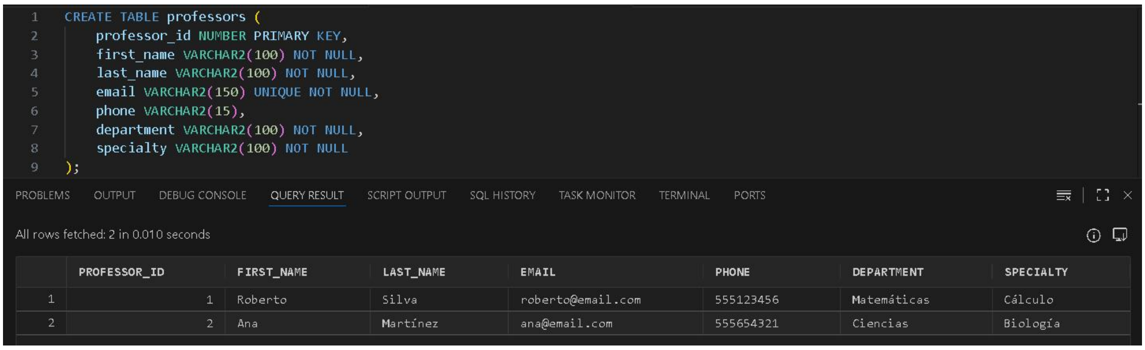
All rows fetched: 3 in 0.023 seconds

	STUDENT_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	CAREER	SEMESTER	ENROLI
1	1	Juan	Pérez	juan@email.com	123456789	Ingeniería de Sistemas	3	30/10/
2	2	María	Gómez	maria@email.com	987654321	Medicina	2	30/10/
3	3	Harold	Chuquiruma	harold.chuquiruma@tecsup.edu.pe	123456789	redes	5	30/10/

Script Tabla Students:

```
CREATE TABLE students (
    student_id NUMBER PRIMARY KEY,
    first_name VARCHAR2(100) NOT NULL,
    last_name VARCHAR2(100) NOT NULL,
    email VARCHAR2(150) UNIQUE NOT NULL,
    phone VARCHAR2(15),
    career VARCHAR2(100) NOT NULL,
    semester NUMBER NOT NULL,
    enrollment_date DATE DEFAULT SYSDATE
);
```

Tabla Professors:



```
CREATE TABLE professors (
    professor_id NUMBER PRIMARY KEY,
    first_name VARCHAR2(100) NOT NULL,
    last_name VARCHAR2(100) NOT NULL,
    email VARCHAR2(150) UNIQUE NOT NULL,
    phone VARCHAR2(15),
    department VARCHAR2(100) NOT NULL,
    specialty VARCHAR2(100) NOT NULL
);
```

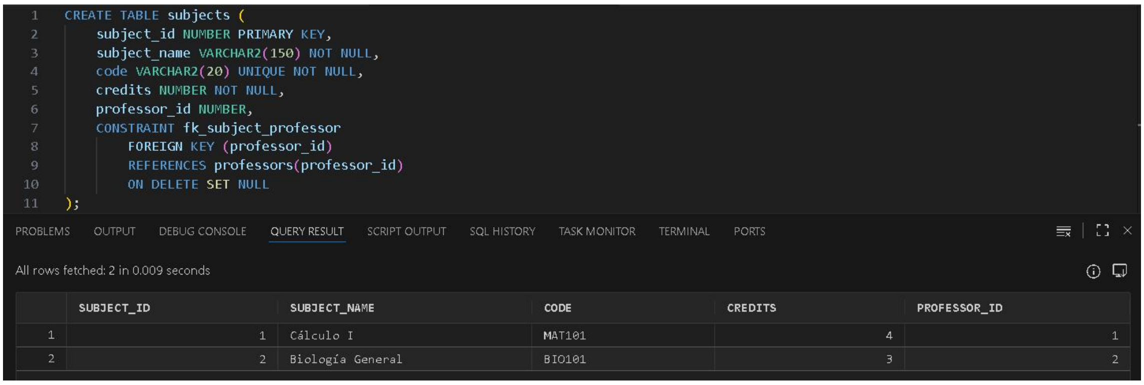
All rows fetched: 2 in 0.010 seconds

	PROFESSOR_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE	DEPARTMENT	SPECIALTY
1	1	Roberto	Silva	roberto@email.com	555123456	Matemáticas	Cálculo
2	2	Ana	Martínez	ana@email.com	555654321	Ciencias	Biología

Script Tabla Professors:

```
CREATE TABLE professors (
    professor_id NUMBER PRIMARY KEY,
    first_name VARCHAR2(100) NOT NULL,
    last_name VARCHAR2(100) NOT NULL,
    email VARCHAR2(150) UNIQUE NOT NULL,
    phone VARCHAR2(15),
    department VARCHAR2(100) NOT NULL,
    specialty VARCHAR2(100) NOT NULL
);
```

Tabla Subjects:



```
CREATE TABLE subjects (
    subject_id NUMBER PRIMARY KEY,
    subject_name VARCHAR2(150) NOT NULL,
    code VARCHAR2(20) UNIQUE NOT NULL,
    credits NUMBER NOT NULL,
    professor_id NUMBER,
    CONSTRAINT fk_subject_professor
        FOREIGN KEY (professor_id)
        REFERENCES professors(professor_id)
        ON DELETE SET NULL
);
```

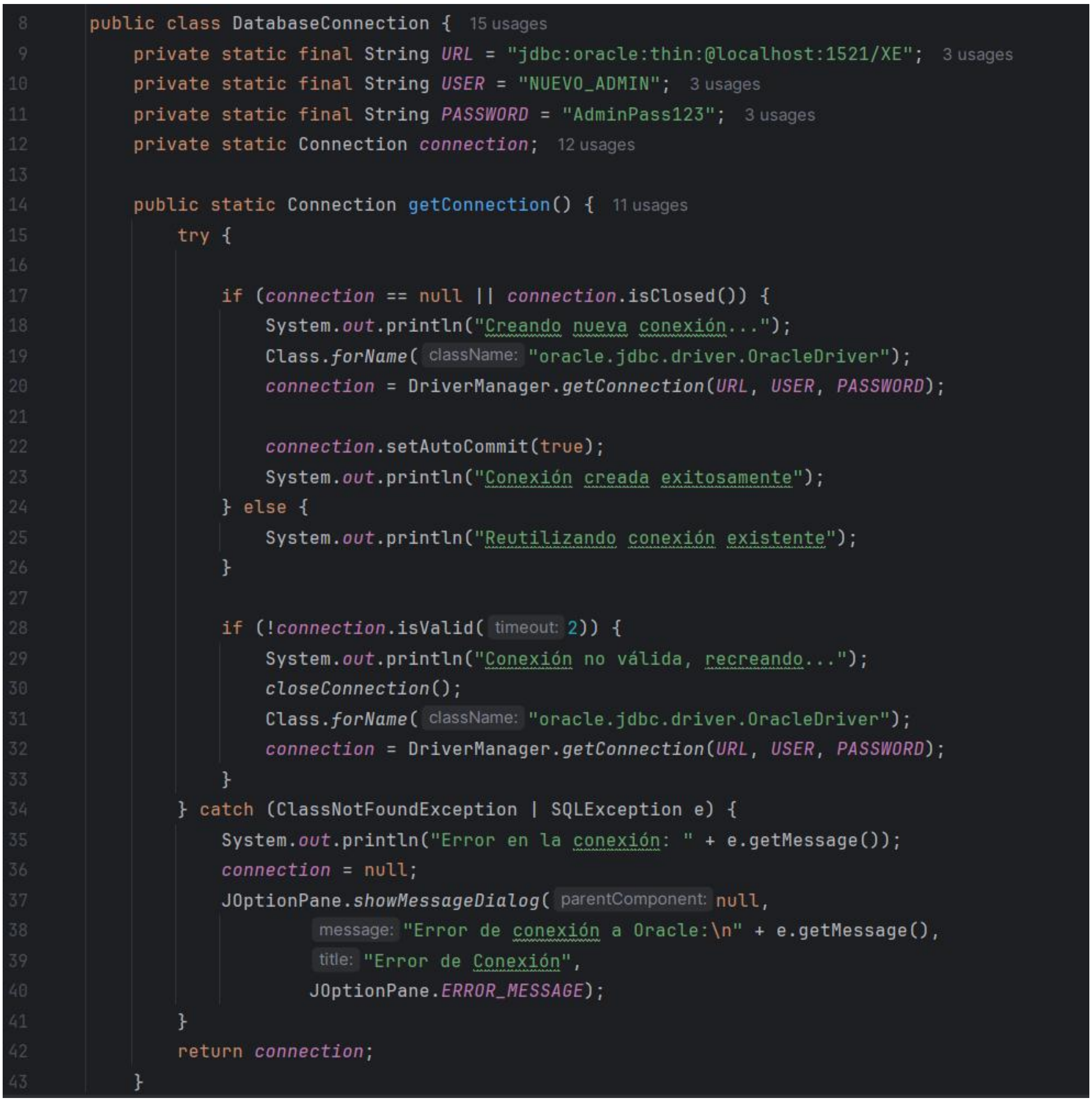
All rows fetched: 2 in 0.009 seconds

	SUBJECT_ID	SUBJECT_NAME	CODE	CREDITS	PROFESSOR_ID
1	1	Cálculo I	MAT101	4	1
2	2	Biología General	BIO101	3	2

Script Tabla Subjects:

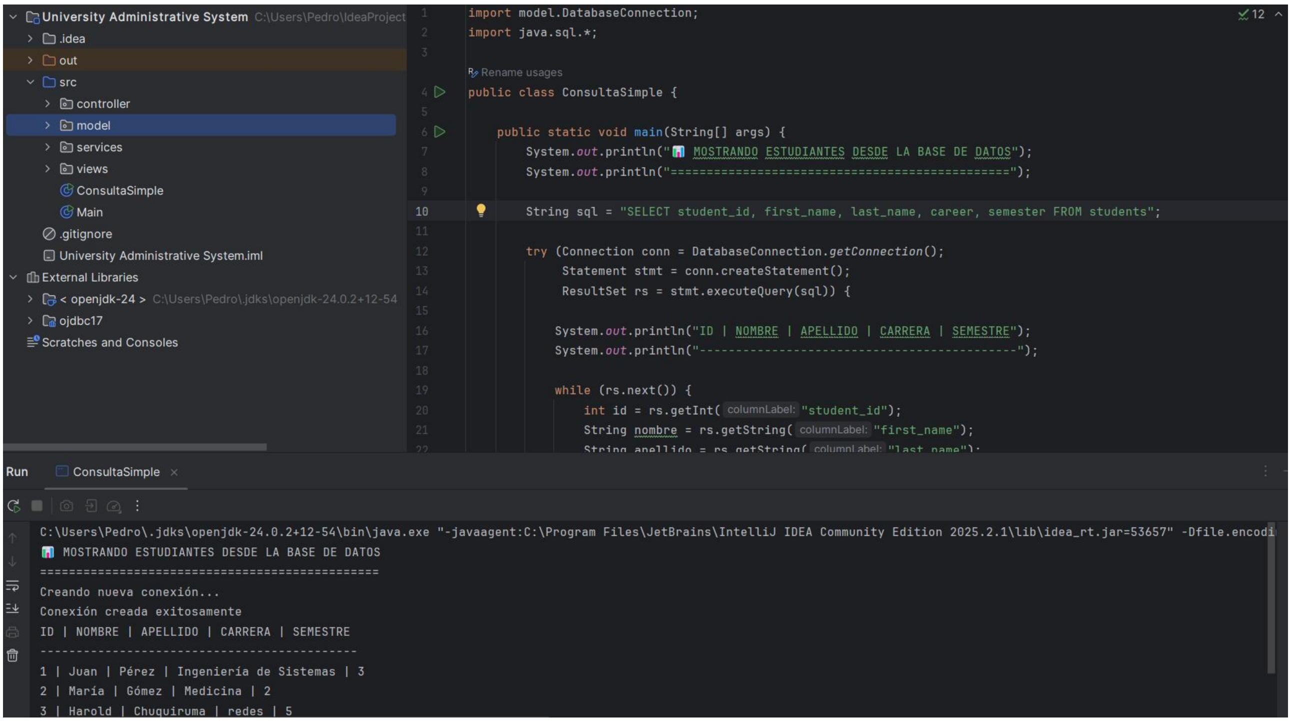
```
CREATE TABLE subjects (

    subject_id NUMBER PRIMARY KEY,

    subject_name VARCHAR2(150) NOT NULL,

    code VARCHAR2(20) UNIQUE NOT NULL,

    credits NUMBER NOT NULL,

    professor_id NUMBER,

    CONSTRAINT fk_subject_professor

        FOREIGN KEY (professor_id)

        REFERENCES professors(professor_id)

        ON DELETE SET NULL

);
```

5) Conexión básica desde lenguaje elegido: (JAVA)



```
public class DatabaseConnection {  15 usages
    private static final String URL = "jdbc:oracle:thin:@localhost:1521/XE";  3 usages
    private static final String USER = "NUEVO_ADMIN";  3 usages
    private static final String PASSWORD = "AdminPass123";  3 usages
    private static Connection connection;  12 usages

    public static Connection getConnection() {  11 usages
        try {

            if (connection == null || connection.isClosed()) {
                System.out.println("Creando nueva conexión...");
                Class.forName( className: "oracle.jdbc.driver.OracleDriver");
                connection = DriverManager.getConnection(URL, USER, PASSWORD);

                connection.setAutoCommit(true);
                System.out.println("Conexión creada exitosamente");
            } else {
                System.out.println("Reutilizando conexión existente");
            }

            if (!connection.isValid( timeout: 2)) {
                System.out.println("Conexión no válida, recreando...");
                closeConnection();
                Class.forName( className: "oracle.jdbc.driver.OracleDriver");
                connection = DriverManager.getConnection(URL, USER, PASSWORD);
            }
        } catch (ClassNotFoundException | SQLException e) {
            System.out.println("Error en la conexión: " + e.getMessage());
            connection = null;
            JOptionPane.showMessageDialog( parentComponent: null,
                    message: "Error de conexión a Oracle:\n" + e.getMessage(),
                    title: "Error de Conexión",
                    JOptionPane.ERROR_MESSAGE);
        }
        return connection;
    }
```

6) consulta simple para mostrar datos


University Administrative System C:\Users\Pedro\IdeaProject
.idea
out
src
controller
model
services
views
ConsultaSimple
Main
.gitignore
University Administrative System.iml
External Libraries
< openjdk-24 > C:\Users\Pedro\.jdks\openjdk-24.0.2+12-54
ojdbc17
Scratches and Consoles
Rename usages
12
Run
ConsultaSimple

```java
import model.DatabaseConnection;
import java.sql.*;

public class ConsultaSimple {

    public static void main(String[] args) {
        System.out.println("📊 MOSTRANDO ESTUDIANTES DESDE LA BASE DE DATOS");
        System.out.println("=============================================");

        String sql = "SELECT student_id, first_name, last_name, career, semester FROM students";

        try (Connection conn = DatabaseConnection.getConnection();
             Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(sql)) {

            System.out.println("ID | NOMBRE | APELLIDO | CARRERA | SEMESTRE");
            System.out.println("------------------------------------------");

            while (rs.next()) {
                int id = rs.getInt( columnLabel: "student_id");
                String nombre = rs.getString( columnLabel: "first_name");
                String apellido = rs.getString( columnLabel: "last_name");
```

```
C:\Users\Pedro\.jdks\openjdk-24.0.2+12-54\bin\java.exe "-javaagent:C:\Program Files\JetBrains\IntelliJ IDEA Community Edition 2025.2.1\lib\idea_rt.jar=53657" -Dfile.encodi
📊 MOSTRANDO ESTUDIANTES DESDE LA BASE DE DATOS
=============================================
Creando nueva conexión...
Conexión creada exitosamente
ID | NOMBRE | APELLIDO | CARRERA | SEMESTRE
------------------------------------------
1 | Juan | Pérez | Ingeniería de Sistemas | 3
2 | María | Gómez | Medicina | 2
3 | Harold | Chuquiruma | redes | 5
```

SCRIPT DE LA CONSULTA SIMPLE EN JAVA:

```java
import model.DatabaseConnection;
import java.sql.*;

public class ConsultaSimple {

    public static void main(String[] args) {
        System.out.println("   MOSTRANDO ESTUDIANTES DESDE LA BASE DE DATOS");
        System.out.println("=============================================");

        String sql = "SELECT student_id, first_name, last_name, career, semester FROM students";

        try (Connection conn = DatabaseConnection.getConnection();
             Statement stmt = conn.createStatement();
             ResultSet rs = stmt.executeQuery(sql)) {

            System.out.println("ID | NOMBRE | APELLIDO | CARRERA | SEMESTRE");
            System.out.println("------------------------------------------");

            while (rs.next()) {
                int id = rs.getInt("student_id");
                String nombre = rs.getString("first_name");
                String apellido = rs.getString("last_name");
                String carrera = rs.getString("career");
                int semestre = rs.getInt("semester");

                System.out.println(id + " │ " + nombre + " │ " + apellido + " │ " + carrera + " │ " +
semestre);
            }

        } catch (SQLException e) {
            System.out.println("❌ Error: " + e.getMessage());
        }
    }
}
```

Link de Github:

https://github.com/IIK1ILLERII/University-Administrative-System.git