



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 conexión 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

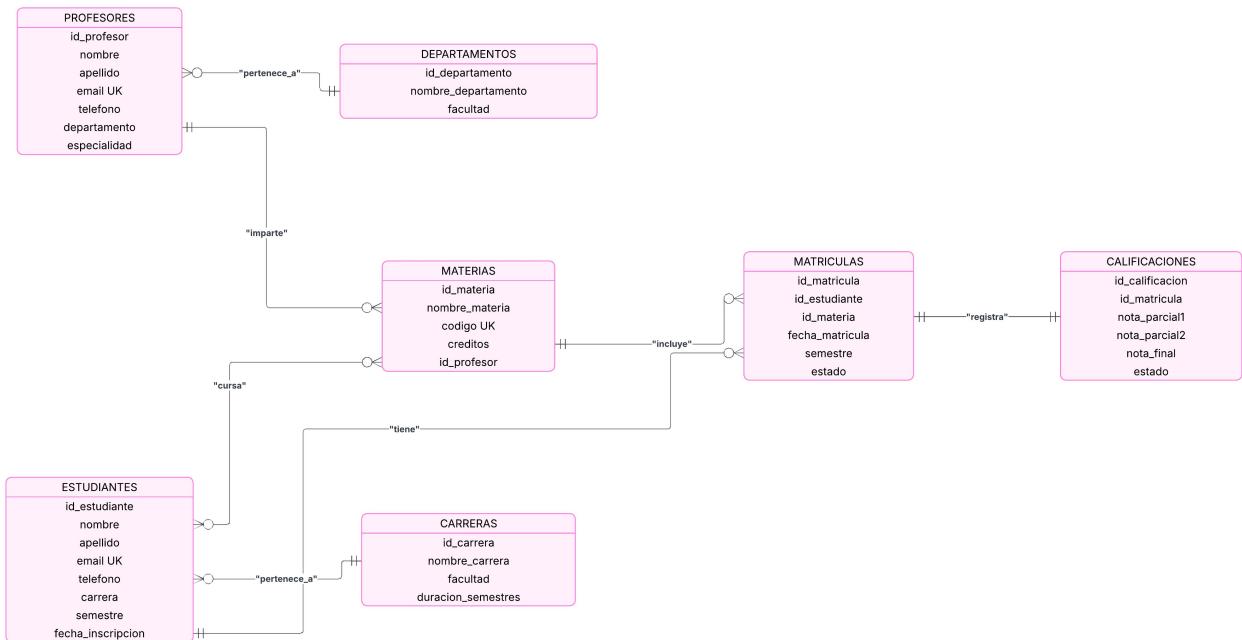
Copyright (c) 1982, 2014, Oracle. All rights reserved.

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:

```

1 CREATE TABLE students (
2     student_id NUMBER PRIMARY KEY,
3     first_name VARCHAR2(100) NOT NULL,
4     last_name VARCHAR2(100) NOT NULL,
5     email VARCHAR2(150) UNIQUE NOT NULL,
6     phone VARCHAR2(15),
7     career VARCHAR2(100) NOT NULL,
8     semester NUMBER NOT NULL,
9     enrollment_date DATE DEFAULT SYSDATE
10 );

```

PROBLEMS OUTPUT DEBUG CONSOLE QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR TERMINAL PORTS

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	Inginería de Sistemas	3	30/10/
2	2	Maria	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:

The screenshot shows a SQL interface with the following details:

- Code Area:**

```
1 CREATE TABLE professors (
2     professor_id NUMBER PRIMARY KEY,
3     first_name VARCHAR2(100) NOT NULL,
4     last_name VARCHAR2(100) NOT NULL,
5     email VARCHAR2(150) UNIQUE NOT NULL,
6     phone VARCHAR2(15),
7     department VARCHAR2(100) NOT NULL,
8     specialty VARCHAR2(100) NOT NULL
9 );
```
- Toolbar:** PROBLEMS, OUTPUT, DEBUG CONSOLE, **QUERY RESULT**, SCRIPT OUTPUT, SQL HISTORY, TASK MONITOR, TERMINAL, PORTS.
- Message Bar:** All rows fetched: 2 in 0.010 seconds.
- Data Grid:** Shows two rows of data from the professors table.

	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:

```
1 CREATE TABLE subjects (
2   subject_id NUMBER PRIMARY KEY,
3   subject_name VARCHAR2(150) NOT NULL,
4   code VARCHAR2(20) UNIQUE NOT NULL,
5   credits NUMBER NOT NULL,
6   professor_id NUMBER,
7   CONSTRAINT fk_subject_professor
8     FOREIGN KEY (professor_id)
9       REFERENCES professors(professor_id)
10      ON DELETE SET NULL
11 );
```

PROBLEMS OUTPUT DEBUG CONSOLE QUERY RESULT SCRIPT OUTPUT SQL HISTORY TASK MONITOR TERMINAL PORTS

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)

```
8  public class DatabaseConnection { 15 usages
9      private static final String URL = "jdbc:oracle:thin:@localhost:1521/XE"; 3 usages
10     private static final String USER = "NUEVO_ADMIN"; 3 usages
11     private static final String PASSWORD = "AdminPass123"; 3 usages
12     private static Connection connection; 12 usages
13
14     public static Connection getConnection() { 11 usages
15         try {
16
17             if (connection == null || connection.isClosed()) {
18                 System.out.println("Creando nueva conexión...");
19                 Class.forName( className: "oracle.jdbc.driver.OracleDriver");
20                 connection = DriverManager.getConnection(URL, USER, PASSWORD);
21
22                 connection.setAutoCommit(true);
23                 System.out.println("Conexión creada exitosamente");
24             } else {
25                 System.out.println("Reutilizando conexión existente");
26             }
27
28             if (!connection.isValid( timeout: 2)) {
29                 System.out.println("Conexión no válida, recreando...");
30                 closeConnection();
31                 Class.forName( className: "oracle.jdbc.driver.OracleDriver");
32                 connection = DriverManager.getConnection(URL, USER, PASSWORD);
33             }
34         } catch (ClassNotFoundException | SQLException e) {
35             System.out.println("Error en la conexión: " + e.getMessage());
36             connection = null;
37             JOptionPane.showMessageDialog( parentComponent: null,
38                                         message: "Error de conexión a Oracle:\n" + e.getMessage(),
39                                         title: "Error de Conexión",
40                                         JOptionPane.ERROR_MESSAGE);
41         }
42         return connection;
43     }
}
```

6) consulta simple para mostrar datos

The screenshot shows the IntelliJ IDEA interface with the following details:

- Project Structure:** The left sidebar shows the project structure under "University Administrative System". It includes a .idea folder, an out folder, a src folder containing controller, model, services, views, ConsultaSimple, Main, and .gitignore; an External Libraries section with openjdk-24; and a Scratches and Consoles section.
- Code Editor:** The main window displays Java code for a "ConsultaSimple" class. The code connects to a database, executes a query to retrieve student information, and prints the results to the console.
- Run Tab:** The bottom tab bar shows the "Run" tab is active, with "ConsultaSimple" selected as the run configuration.
- Console Output:** The bottom pane shows the output of the Java application running. It prints a header, creates a new connection, and then lists student data from the database.

```
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");
                System.out.println(id + " | " + nombre + " | " + apellido + " | " + career + " | " + semester);
            }
        }
    }
}
```

```
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.encoding=UTF-8
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 | Chuguiruma | redes | 5
```

SCRIPT DE LA CONSULTA SIMPLE EN 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");
                String semestre = rs.getString("semester");
                System.out.println(id + " | " + nombre + " | " + apellido + " | " + carrera + " | " + semestre);
            }
        }
    }
}
```

```
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("X Error: " + e.getMessage());
}
}
```

Link de Github:

<https://github.com/IICK1ILLERII/University-Administrative-System.git>