

## Prueba del segundo bimestre

Desarrollar una aplicación en Java que permita gestionar calificaciones de estudiantes mediante una interfaz gráfica. El sistema debe incluir un login y una funcionalidad para ingresar las calificaciones de 5 materias para un estudiante específico, identificado por su número de cédula. Debe además calcular el promedio. La información debe ser almacenada en una base de datos MySQL.

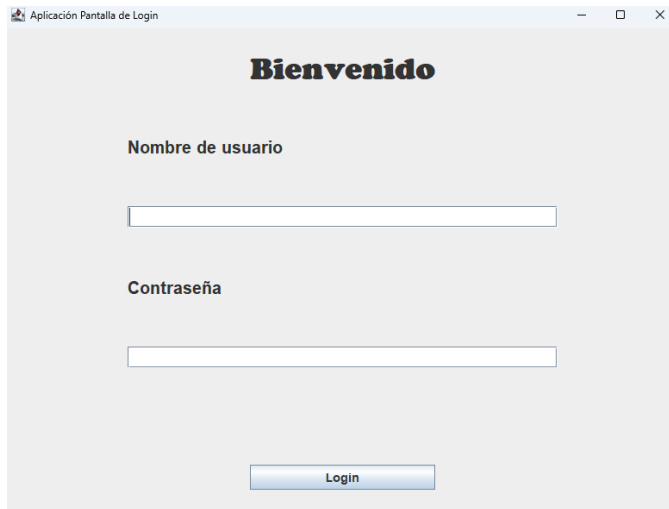
## Enlace gitHub

[https://github.com/CristianTambaco/Prueba3\\_Gestion\\_Calificaciones.git](https://github.com/CristianTambaco/Prueba3_Gestion_Calificaciones.git)

### Requerimientos Funcionales

#### 1. Pantalla de Login:

- Permitir al usuario ingresar un nombre de usuario y una contraseña.
  - Validar las credenciales almacenadas en una tabla de usuarios en MySQL.
  - Si las credenciales son incorrectas, mostrar un mensaje de error.
  - Si las credenciales son correctas, permitir el acceso a la pantalla de gestión de calificaciones.
- 
- Se muestra la pantalla de login



Aplicación Pantalla de Login

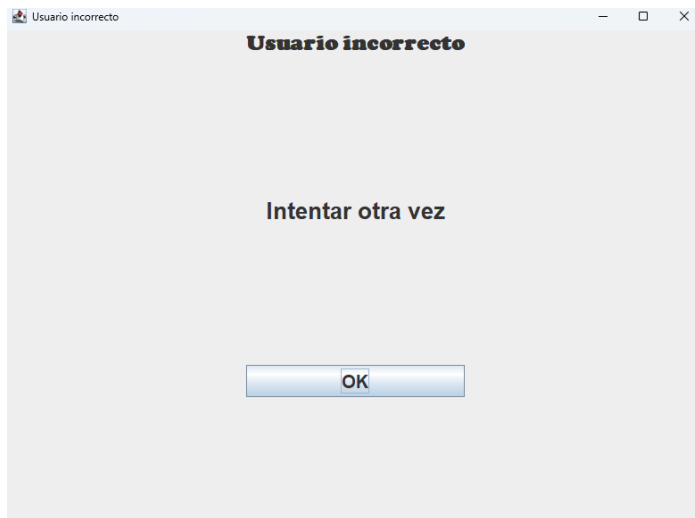
**Bienvenido**

Nombre de usuario

Contraseña

Login

- Si las credenciales no son correctas, se muestra una ventana de “usuario incorrecto”



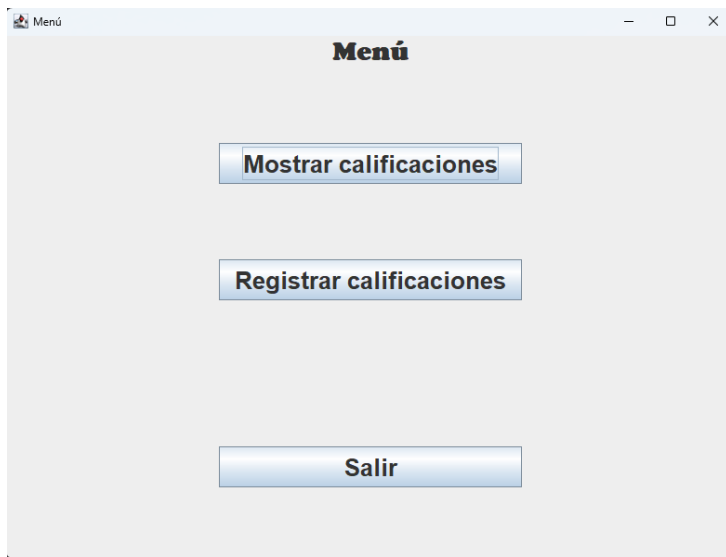
Usuario incorrecto

**Intentar otra vez**

OK

2. Pantalla de Gestión de Calificaciones:

- Permitir ingresar el número de cédula de un estudiante.
  - Registrar las calificaciones de cinco materias.
  - Validar las calificaciones para asegurarse de que sean números entre 0 y 20.
  - Guardar los datos del estudiante y sus calificaciones en la base de datos.
  - Mostrar un mensaje de confirmación al guardar correctamente.
- 
- Si se ingresa con las credenciales correcta muestra el menú de opciones



- Al hacer clic en Mostrar calificaciones, muestra la pantalla para la visualización de calificaciones.



- Se cargan los datos que están registrados en My SQLWorkbench

Ver calificaciones

## Calificaciones

COD: 1  
Cedula: 184839543  
Nombre: Adrian  
Estudiante1: 6.8  
Estudiante2: 8.0  
Estudiante3: 9.0  
Estudiante4: 7.0  
Estudiante5: 7.0

COD: 2  
Cedula: 777777777  
Nombre: Carlos  
Estudiante1: 4.0  
Estudiante2: 10.0  
Estudiante3: 9.1  
Estudiante4: 10.0  
Estudiante5: 7.9

COD: 3  
Cedula: 454545454  
Nombre: Vicente  
Estudiante1: 8.0  
Estudiante2: 6.0  
Estudiante3: 7.0  
Estudiante4: 9.8  
Estudiante5: 7.1

Cargar Datos

- En la opción registrar calificaciones se muestra los campos para el registro

Registro

## Registro

Cedula

Nombre

Nota1

Nota2

Nota3

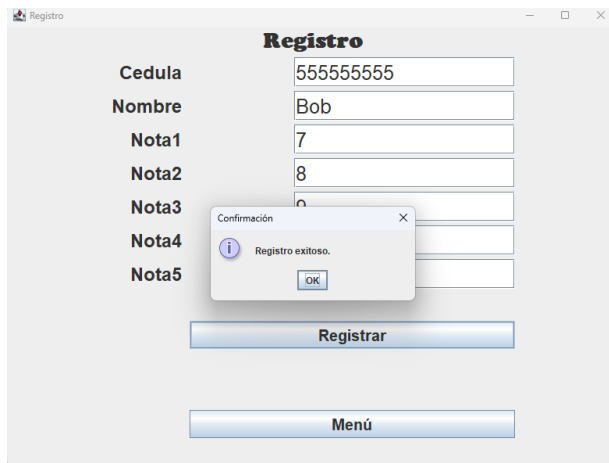
Nota4

Nota5

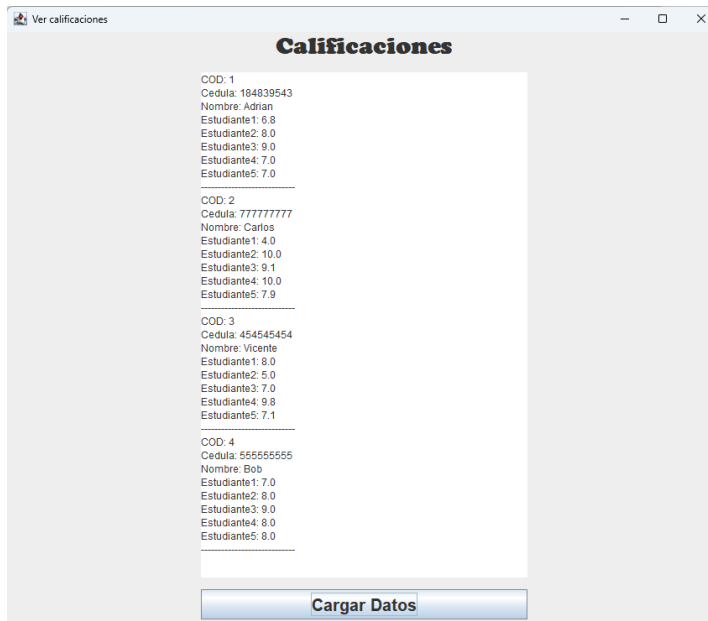
Registrar

Menú

- Se registra un estudiante



- Se visualiza la el registro



### 3. Conexión a MySQL:

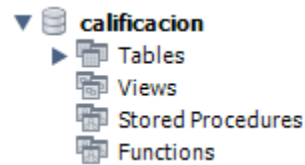
- La base de datos debe contener dos tablas:
- users: Para las credenciales del login.
- students: Para almacenar la cédula, nombre y calificaciones de cada estudiante.
- Utilizar una conexión establecida con el controlador JDBC para interactuar con MySQL.

- Se crea la base de datos Calificacion en mysql workbench

### -- Estructura de la Base de Datos

**CREATE DATABASE Calificacion;**

## USE Calificacion;



- Se crea la tabla usuarios y la tabla estudiante, una vez creado las tablas se inserta registros a cada tabla.

### CREATE TABLE usuarios (

id INT AUTO\_INCREMENT PRIMARY KEY,

username VARCHAR(50) NOT NULL UNIQUE,

pass VARCHAR(255) NOT NULL

);

-- Tabla estudiantes:

### CREATE TABLE estudiantes (

id INT AUTO\_INCREMENT PRIMARY KEY,

cedula VARCHAR(10) NOT NULL UNIQUE,

nombre VARCHAR(100) NOT NULL,

estudiante1 DOUBLE NOT NULL,

estudiante2 DOUBLE NOT NULL,

estudiante3 DOUBLE NOT NULL,

estudiante4 DOUBLE NOT NULL,

estudiante5 DOUBLE NOT NULL

);

```
INSERT INTO usuarios(username, pass)values
```

```
("usuario", "usuario123");
```

```
INSERT INTO estudiantes(cedula, nombre, estudiante1, estudiante2, estudiante3, estudiante4,  
estudiante5)values
```

```
("184839543", "Adrian", 6.8, 8, 9, 7,7),
```

```
("777777777", "Carlos", 4, 10, 9.1, 10,7.9),
```

```
("454545454", "Vicente", 8, 5, 7, 9.8,7.1);
```

Table: **estudiantes**

Columns:

<b>id</b>	int AI PK
<b>cedula</b>	varchar(10)
<b>nombre</b>	varchar(100)
estudiante1	double
estudiante2	double
estudiante3	double
estudiante4	double
estudiante5	double

	id	cedula	nombre	estudiante1	estudiante2	estudiante3	estudiante4	estudiante5
▶	1	184839543	Adrian	6.8	8	9	7	7
	2	777777777	Carlos	4	10	9.1	10	7.9
	3	454545454	Vicente	8	5	7	9.8	7.1
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

Table: **usuarios**

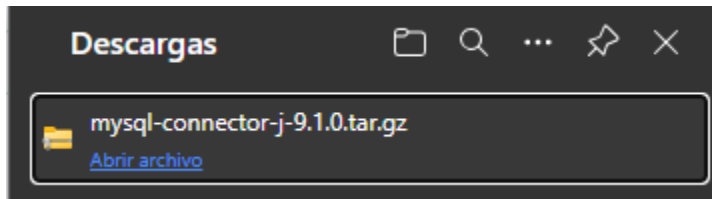
Columns:

<b>id</b>	int AI PK
<b>username</b>	varchar(50)
<b>pass</b>	varchar(255)

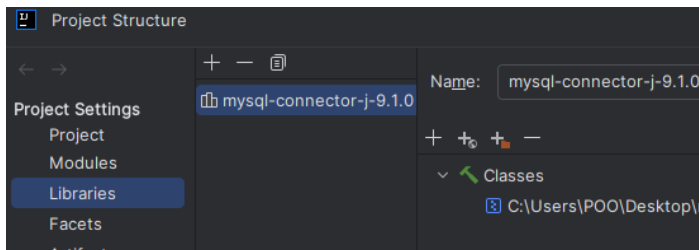
	id	username	pass
▶	1	usuario	usuario123
*	NULL	NULL	NULL



- Se descarga “mysql-connector-j” para la conexión con la base de datos Mysql Workbench



- Luego se añade al proyecto para la conexión con la base de datos



- Se verifica en la consola el estado de la conexión con la base de datos: “Conectado”, y se muestra las credenciales para el login.

```
C:\Users\P00\.jdk\openjdk-23\bin\java.exe "-javaagent:C:\Program Files
-Dfile.encoding=UTF-8 -Dsun.stdout.encoding=UTF-8 -Dsun.stderr.encoding
.0\mysql-connector-j-9.1.0.jar Main
CONECTADO
id: 1, username: usuario, password: usuario123
```

#### 4. Extras Opcionales:

- Calcular y mostrar el promedio de las calificaciones.
- Indicar si el estudiante aprueba (promedio  $\geq 60$ ) o reprueba (promedio  $< 60$ ).
- Mostrar un historial de calificaciones almacenadas.

#### Estructura de la Base de Datos

Tabla usuarios:

CREATE TABLE usuarios (

```

id INT AUTO_INCREMENT PRIMARY KEY,
username VARCHAR(50) NOT NULL UNIQUE,
password VARCHAR(255) NOT NULL
);

```

Tabla estudiante:

```

CREATE TABLE estudiantes (
  id INT AUTO_INCREMENT PRIMARY KEY,
  cedula VARCHAR(10) NOT NULL UNIQUE,
  nombre VARCHAR(100) NOT NULL,
  estudiante1 DOUBLE NOT NULL,
  estudiante2 DOUBLE NOT NULL,
  estudiante3 DOUBLE NOT NULL,
  estudiante4 DOUBLE NOT NULL,
  estudiante5 DOUBLE NOT NULL
);

```

Rúbrica:

Elementos	Puntaje	Descripción
Conexión a MySQL	15	a aplicación se conecta correctamente a MySQL y utiliza el controlador JDBC.
Login funcional	15	Valida credenciales desde la tabla users en MySQL. Pantalla de Gestión de Calificaciones
Pantalla de Gestión de Calificaciones	20	Permite ingresar cédula y calificaciones, valida los datos y los guarda en MySQL.
Validación de datos	10	Valida correctamente los rangos de calificaciones
Persistencia de datos	10	Las calificaciones ingresadas se almacenan correctamente en la tabla
Interfaz gráfica	15	La interfaz es funcional y fácil de usar, con diseño claro y botones intuitivos.
Cálculo del promedio	5	Calcula correctamente el promedio
Manejo de errores	5	Uso de try-catch
Repositorio github	5	Repositorio legible

