

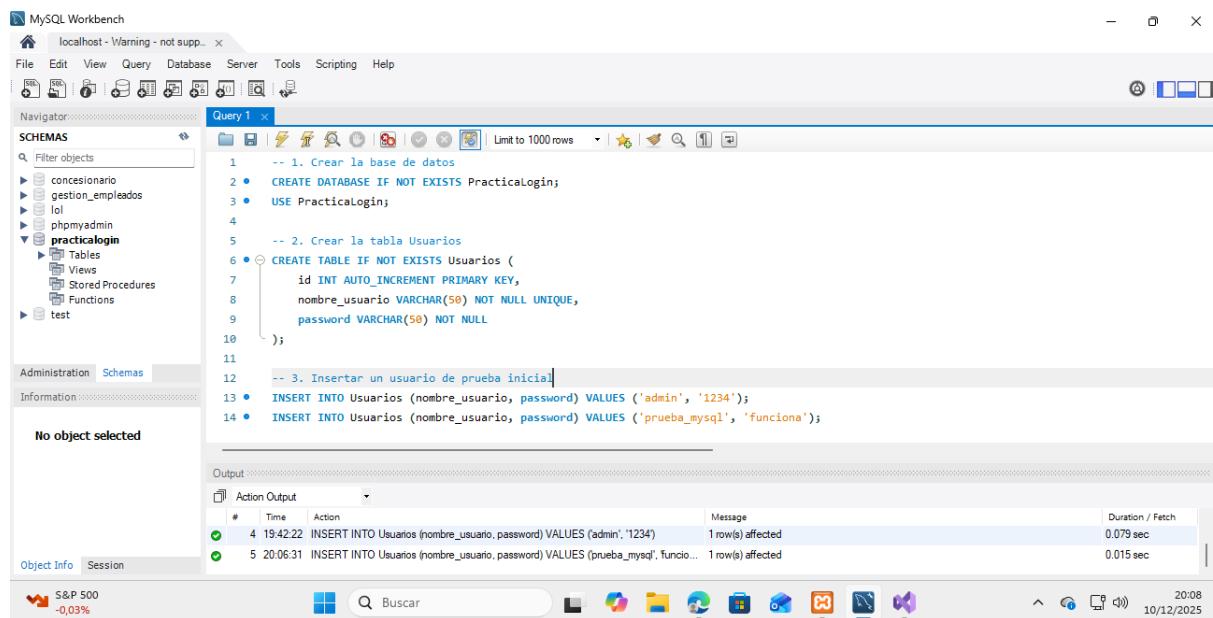
Documentación de Pruebas: Práctica 3 - Conexión y Autenticación MySQL

Práctica: Práctica 3. Conexión y Autenticación contra MySQL **Objetivo:** Migrar la lógica de autenticación para verificar credenciales contra una base de datos real (MySQL) y permitir el registro de usuarios.

1. Configuración de la Base de Datos

Descripción: Se ha creado la base de datos PracticaLogin y la tabla Usuarios con los campos requeridos (`id`, `nombre_usuario`, `password`).

Evidencia (Entregable 2): Se muestra a continuación una captura del gestor de base de datos (Workbench/phpMyAdmin) listando los usuarios registrados, confirmando que la estructura es correcta.



The screenshot shows the MySQL Workbench interface. In the left sidebar, under the 'SCHEMAS' tab, the 'practicalogin' database is selected. The 'Tables' section shows the 'Usuarios' table. The main area displays a SQL query window titled 'Query 1' containing the following code:

```
1 -- 1. Crear la base de datos
2 • CREATE DATABASE IF NOT EXISTS Practicalogin;
3 • USE Practicalogin;
4
5 -- 2. Crear la tabla Usuarios
6 • CREATE TABLE IF NOT EXISTS Usuarios (
7     id INT AUTO_INCREMENT PRIMARY KEY,
8     nombre_usuario VARCHAR(50) NOT NULL UNIQUE,
9     password VARCHAR(50) NOT NULL
10 );
11
12 -- 3. Insertar un usuario de prueba inicial
13 • INSERT INTO Usuarios (nombre_usuario, password) VALUES ('admin', '1234');
14 • INSERT INTO Usuarios (nombre_usuario, password) VALUES ('prueba_mysql', 'funciona');
```

Below the query window, the 'Output' pane shows the results of the executed queries:

#	Time	Action	Message	Duration / Fetch
4	19:42:22	INSERT INTO Usuarios (nombre_usuario, password) VALUES ('admin', '1234')	1 row(s) affected	0.079 sec
5	20:06:31	INSERT INTO Usuarios (nombre_usuario, password) VALUES ('prueba_mysql', 'funciona')	1 row(s) affected	0.015 sec

2. Evidencias de Ejecución: Autenticación (Login)

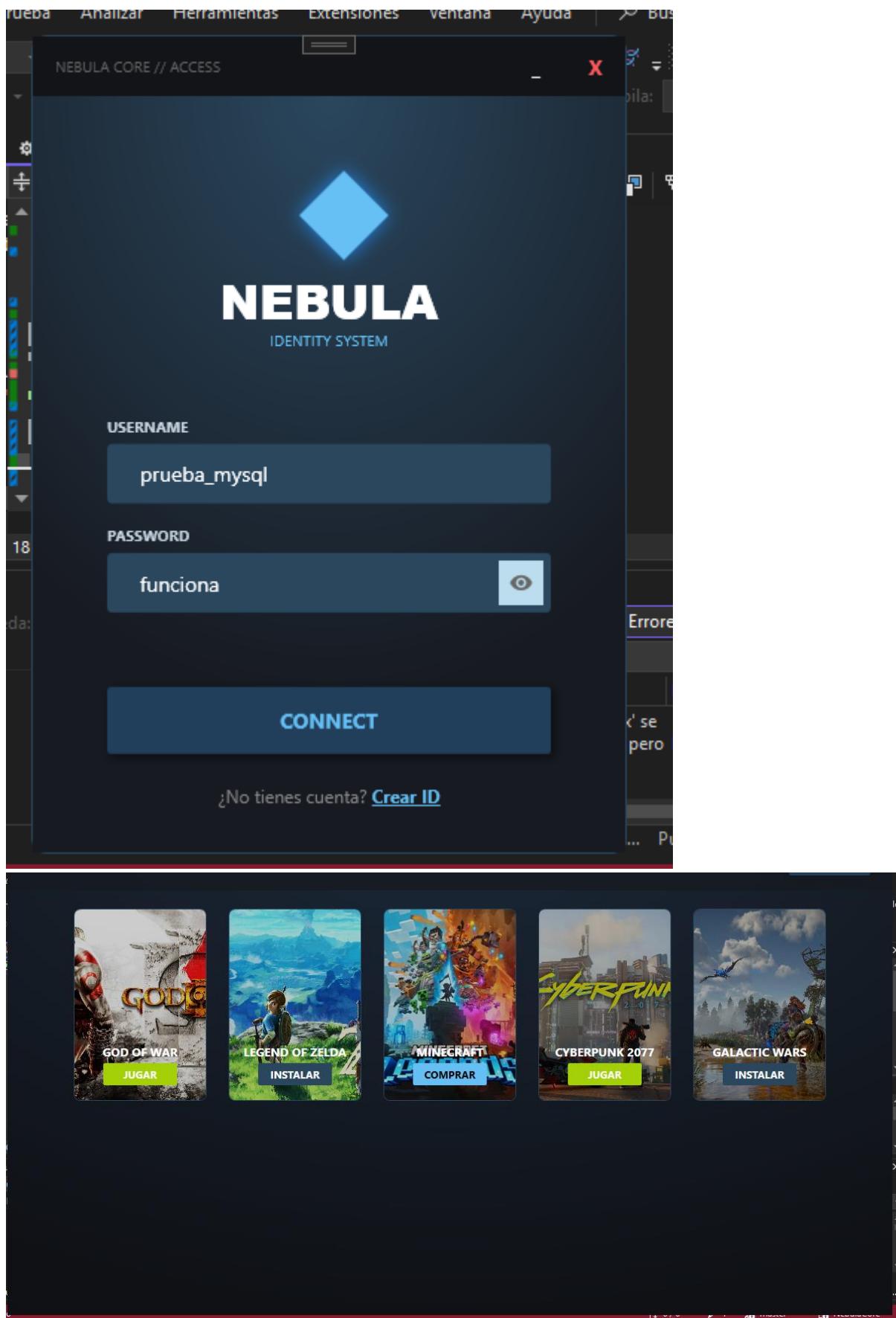
Se valida que la aplicación se conecta correctamente mediante la clase ConexionBD y ejecuta consultas SELECT.

Prueba 1: Login Exitoso (Verificación SQL)

- **Objetivo:** Verificar que un usuario existente en la BBDD puede iniciar sesión.
- **Credencial Usada:** Usuario creado directamente en SQL (prueba_mysql o admin).
- **Resultado Esperado:** La aplicación consulta la BBDD, encuentra coincidencia y redirige a la Home.
- **Resultado Obtenido: Correcto.** Acceso permitido y redirección exitosa.

Evidencia:

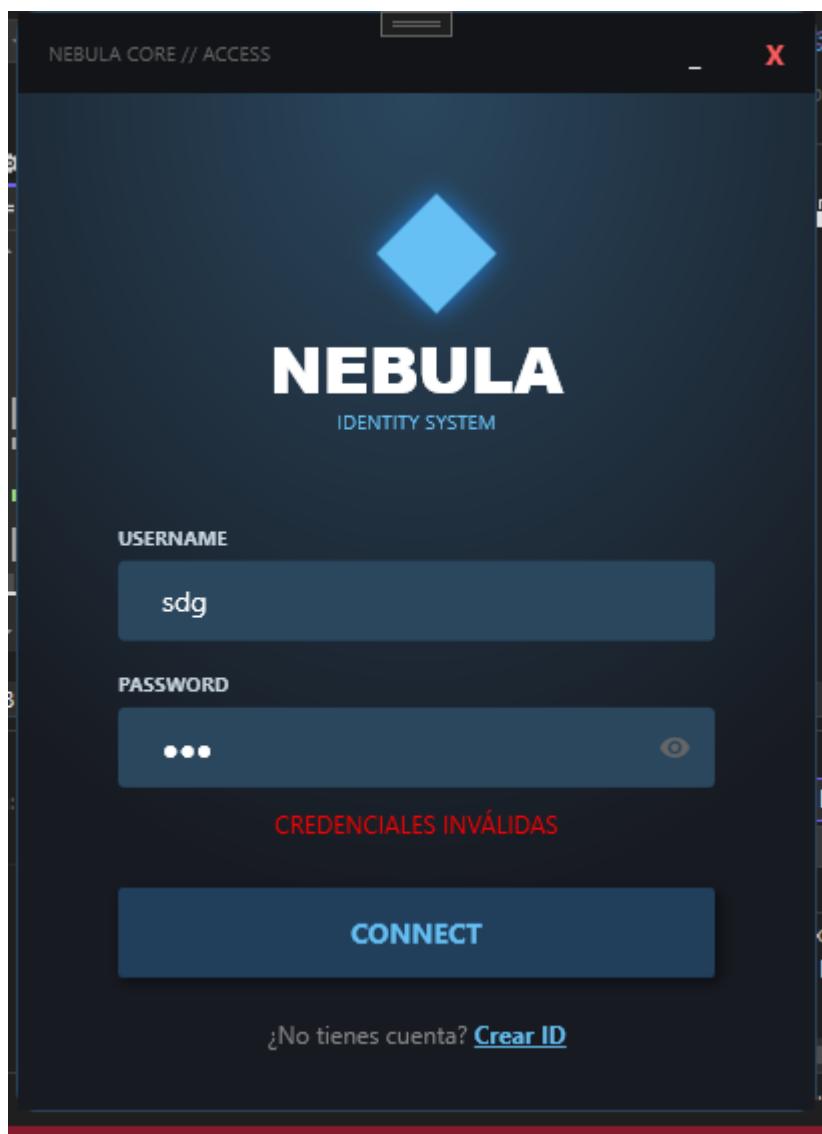
[INSERTAR CAPTURA 2 AQUÍ: Captura de la pantalla Home (HomeWindow) tras loguearse.]



Prueba 2: Login Fallido (Validación SQL)

- **Objetivo:** Verificar que la aplicación deniega el acceso si la consulta SQL no devuelve resultados.
- **Credencial Usada:** Usuario inexistente (ej: usuario_falso).
- **Resultado Esperado:** Mensaje de error rojo ("Credenciales inválidas") tras verificar que COUNT(*) == 0.
- **Resultado Obtenido: Correcto.** El sistema impidió el acceso.

Evidencia:



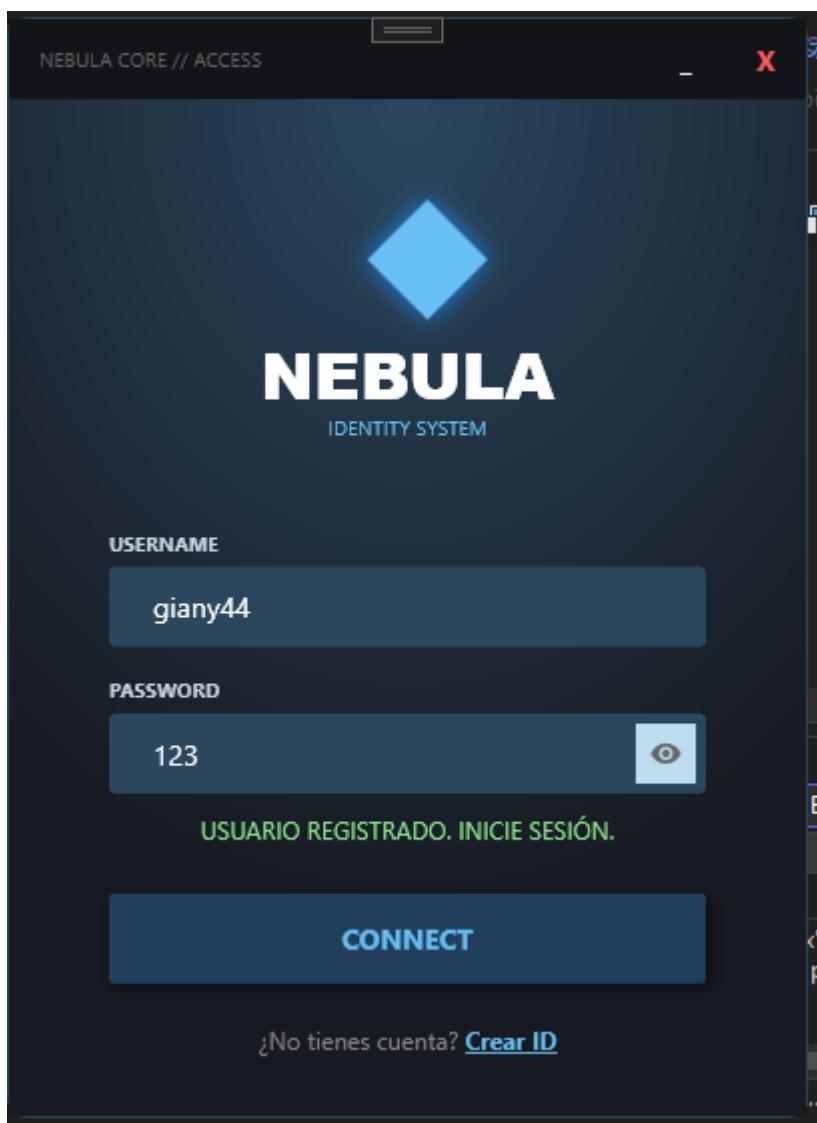
3. Evidencias de Ejecución: Registro (Nuevo Usuario)

Se valida la funcionalidad del botón "Nuevo Usuario" para ejecutar sentencias INSERT en la base de datos.

Prueba 3: Registro de Usuario Correcto

- **Objetivo:** Insertar un nuevo registro en la tabla Usuarios desde la interfaz gráfica.
- **Acción:** Se introdujeron credenciales nuevas y se pulsó "Registrar" (o botón equivalente).
- **Resultado Esperado:** Mensaje de éxito ("Usuario Registrado") y aparición del dato en la BBDD.
- **Resultado Obtenido: Correcto.** La inserción se realizó sin errores.

Evidencia de Interfaz:



4. Entrega y Código Fuente

Descripción: El proyecto integra el conector MySql.Data y utiliza una arquitectura con clase de conexión separada (ConexionBD.cs) y manejo de excepciones (try-catch) para asegurar la robustez ante fallos de conexión.

Enlace al Repositorio Git: <https://github.com/Giany44/NebulaCore.git>

Compilación: El proyecto compila y se ejecuta sin errores en Visual Studio.