

UNIVERSIDAD MARIANO GÁLVEZ DE GUATEMALA

CENTRO UNIVERSITARIO PETÉN



INGENIERÍA EN SISTEMAS

LUIS GARCÍA

PROGRAMACION I

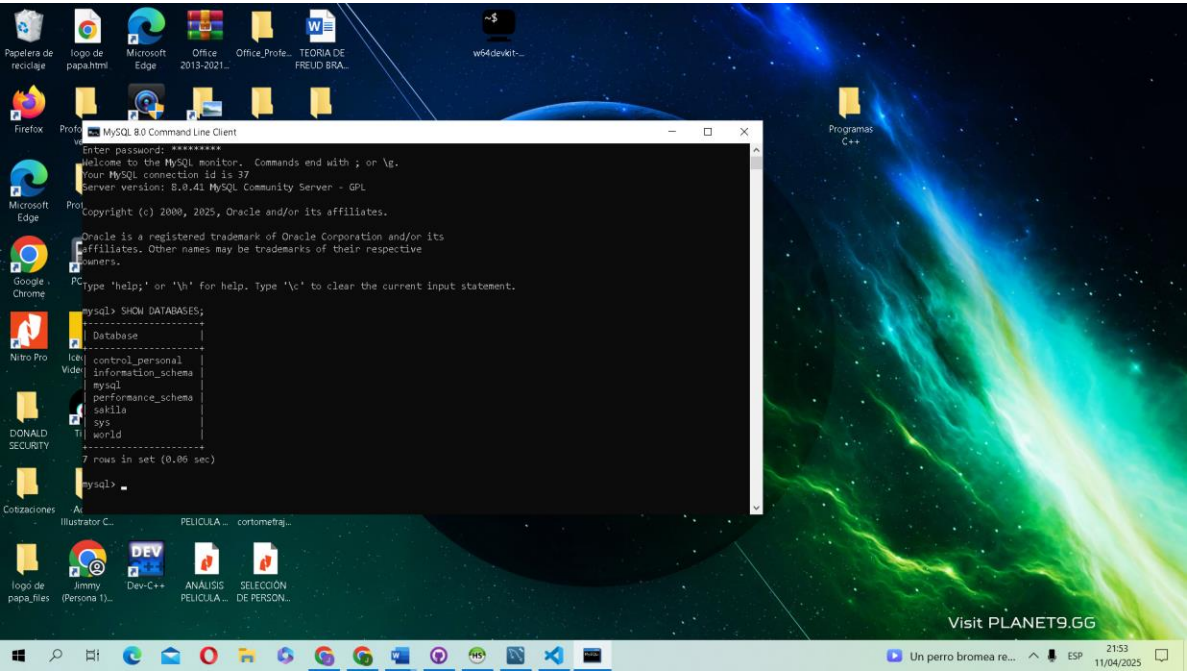
TERCER SEMESTRE

AUTOBIOGRAFÍA

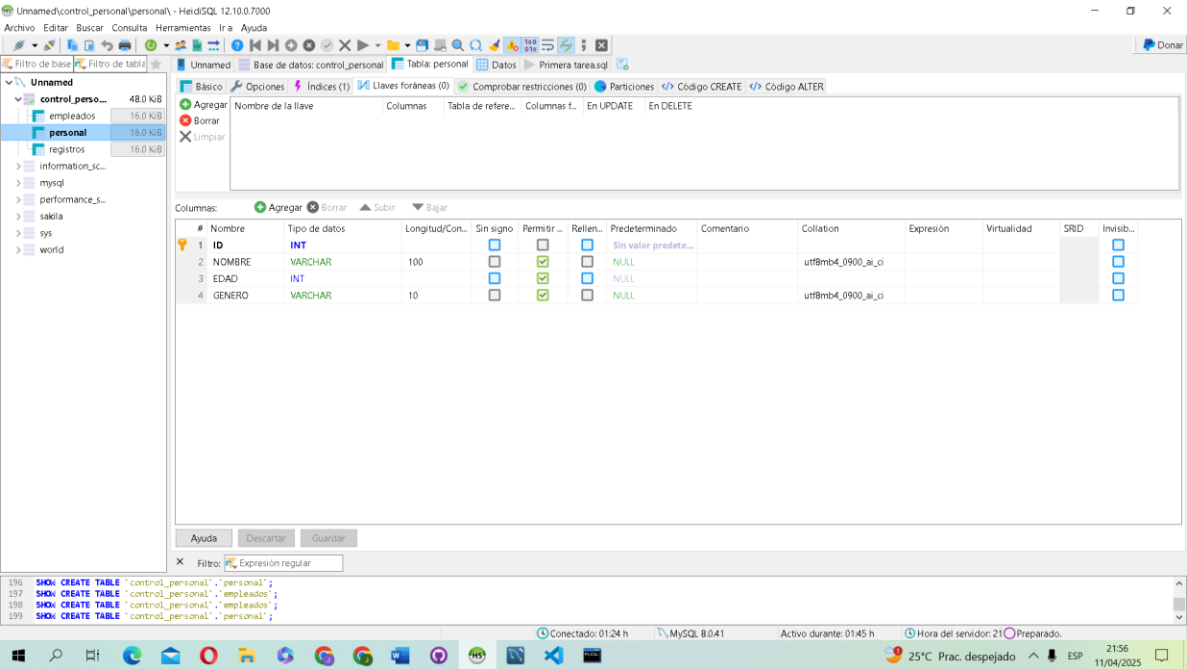
JIMMY ROBERTO ALVARADO ARROYO

CARNÉ: 1690-24-24937

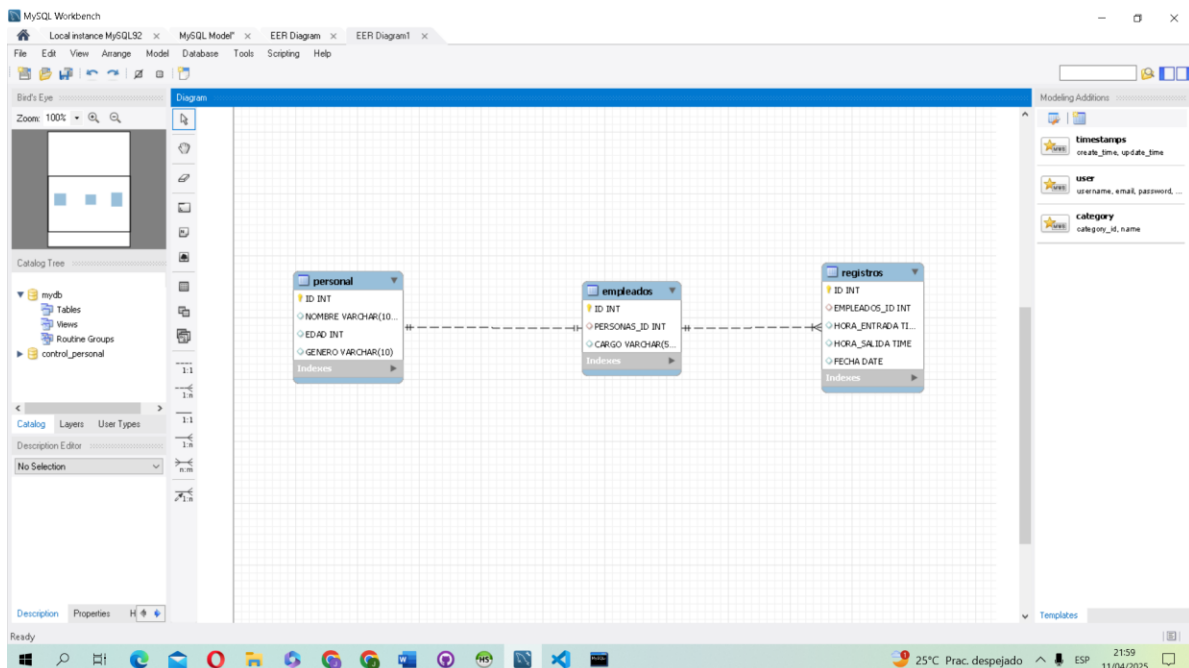
Primer paso: Crear la base de datos utilizando el cliente de línea de comandos (Command Line Client).



Accedemos a HeidiSQL y procedemos a definir las llaves foráneas para establecer las relaciones entre las tablas.



Accedemos a MySQL Workbench para diseñar los diagramas de la base de datos.

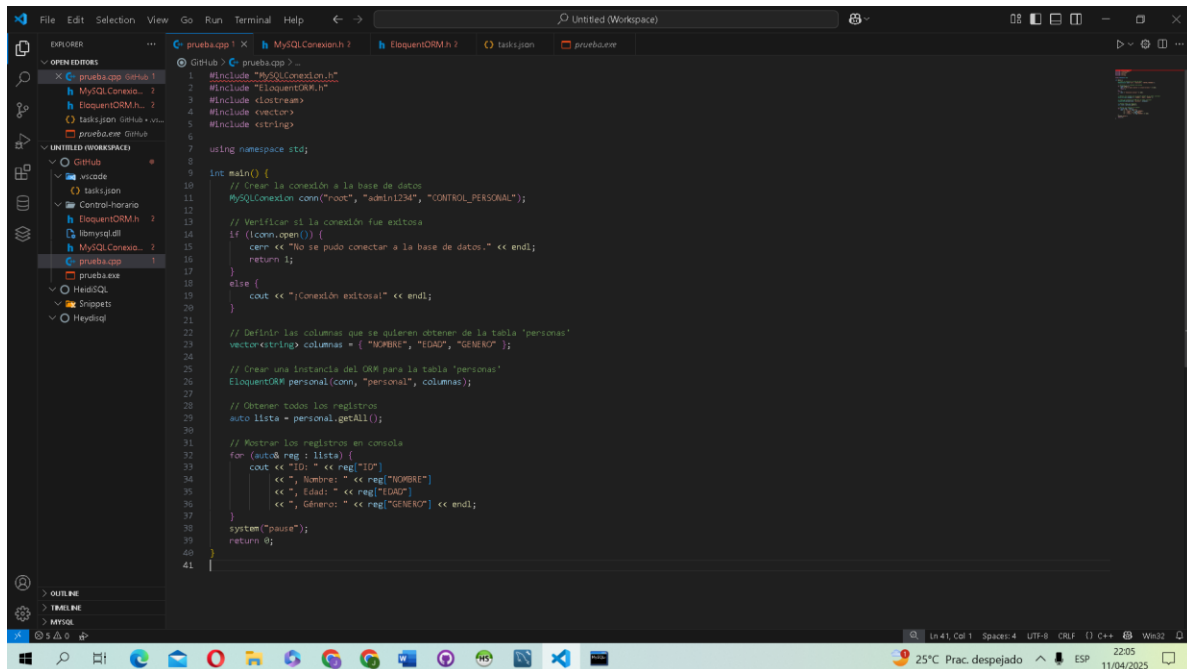


Abrimos Visual Studio Code y configuramos las rutas de los archivos necesarias para establecer la conexión entre el proyecto y la base de datos.

The screenshot shows the Visual Studio Code editor with a **tasks.json** file open. The file is located in the **tasks** folder of the workspace. The configuration is as follows:

```
1 {
2   "tasks": [
3     {
4       "type": "cppbuild",
5       "label": "C/C++: g++ build active file",
6       "command": "C:\\Program Files\\winlibs-x86_64-posix-seh-gcc-14.2.0-11vm-19.1.7-mingw-w64ucrt-12.0.0-r3\\mingw64\\bin\\g++.exe",
7       "args": [
8         "-fdiagnostics-color=always",
9         "-g",
10        "${file}",
11        "-I",
12        "C:/Users/USUARIO/Downloads/mysql-8.0.41-win64/mysql-8.0.41-win64/include",
13        "-L",
14        "C:/Users/USUARIO/Downloads/mysql-8.0.41-win64/mysql-8.0.41-win64/lib",
15        "-llibmysql",
16        "-o",
17        "${fileDirname}\\${fileBasenameNoExtension}.exe"
18      ],
19      "options": {
20        "cwd": "${fileDirname}"
21      },
22      "problemMatcher": [
23        "$gcc"
24      ],
25      "group": {
26        "kind": "build",
27        "isDefault": true
28      },
29      "detail": "Task generated by Debugger."
30    }
31  ],
32  "version": "2.0.0"
33 }
```

Probar y validar la conexión del programa con la base de datos ejecutando el archivo principal desde Visual Studio Code. Si la conexión es exitosa, se visualizarán los datos almacenados en la tabla 'personal' directamente en la consola del sistema.



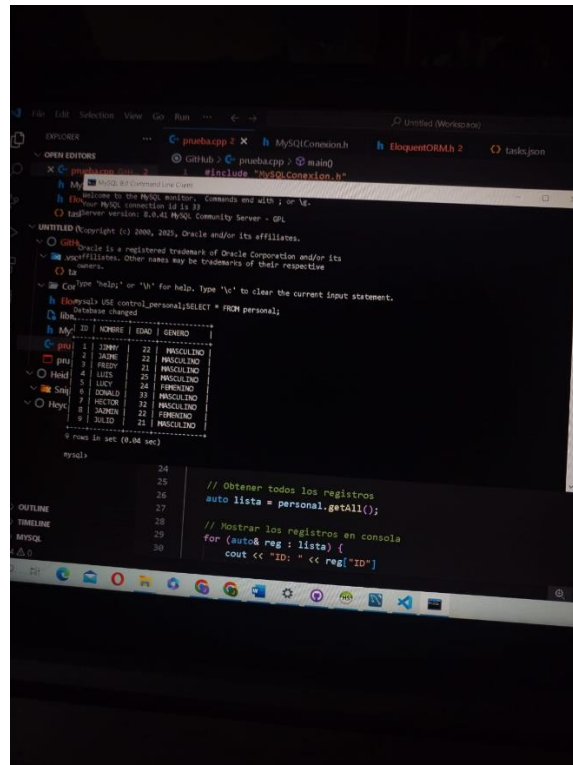
The screenshot shows the Visual Studio Code interface with a C++ file named `prueba.cpp` open. The code is as follows:

```
1 #include "MySQLConexion.h"
2 #include "EloquentORM.h"
3 #include <iostream>
4 #include <vector>
5 #include <string>
6
7 using namespace std;
8
9 int main() {
10     // Crear la conexión a la base de datos
11     MySQLConexion conn("root", "saahin1234", "CONTROL_PERSONAL");
12
13     // Verificar si la conexión fue exitosa
14     if (!conn.open()) {
15         cerr << "No se pudo conectar a la base de datos." << endl;
16         return 1;
17     }
18     else {
19         cout << "¡Conexión exitosa!" << endl;
20     }
21
22     // Definir las columnas que se quieren obtener de la tabla 'personas'
23     vector<string> columnas = { "NOMBRE", "EDAD", "GENERO" };
24
25     // Crear una instancia del ORM para la tabla 'personas'
26     EloquentORM personal(conn, "personal", columnas);
27
28     // Obtener todos los registros
29     auto lista = personal.getAll();
30
31     // Mostrar los registros en consola
32     for (auto& reg : lista) {
33         cout << "ID: " << reg["ID"]
34              << ", Nombre: " << reg["NOMBRE"]
35              << ", Edad: " << reg["EDAD"]
36              << ", Género: " << reg["GENERO"] << endl;
37     }
38     system("pause");
39     return 0;
40 }
41
```

The interface also shows a file explorer on the left with a project structure including folders like `taskijon`, `Control-horario`, `EloquentORM.h`, `MySQLConexion.h`, `prueba.cpp`, `pruebas`, `HeadSQL`, `Snippets`, and `MySQL`. The status bar at the bottom indicates the file is at line 41, column 1, with UTF-8 encoding and CRLF line endings.

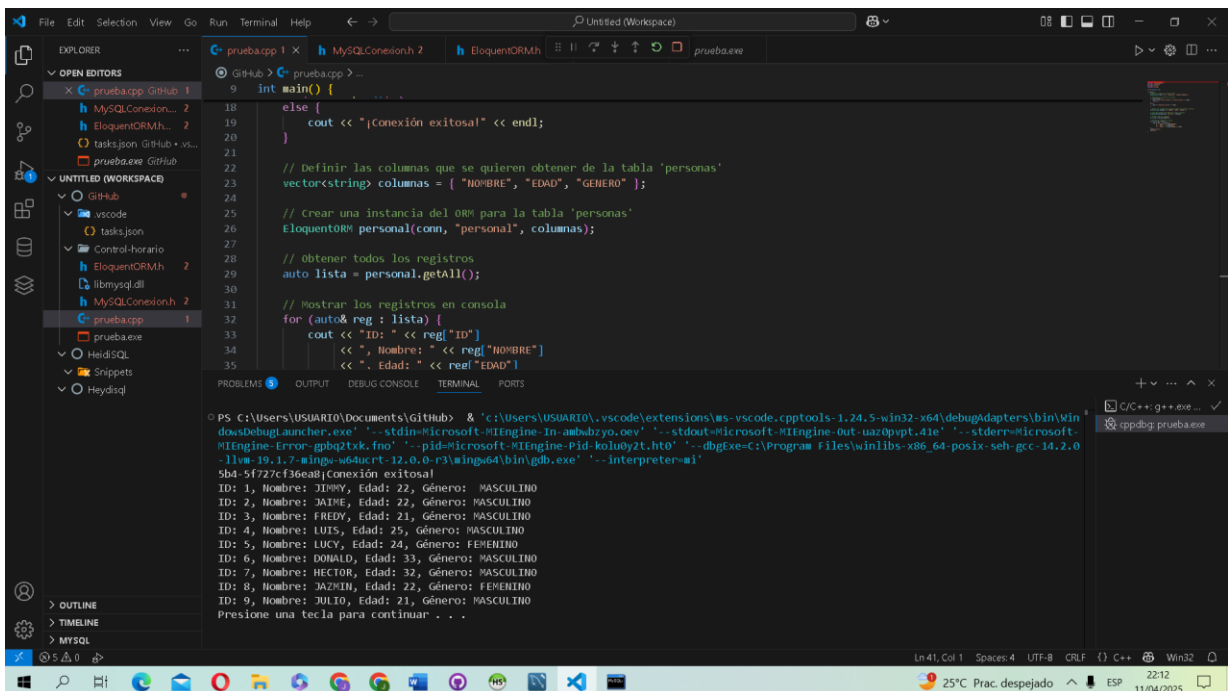
Antes de ejecutar el programa en Visual Studio Code, abrimos el *MySQL Command Line Client* para verificar que la base de datos y los datos estaban correctamente cargados. Ejecutamos los siguientes comandos: USE control_personal;

SELECT * FROM personal; Como resultado, se mostraron todos los registros de la tabla personal, lo que confirma que los datos están almacenados correctamente en la base de datos. Esta verificación previa es importante para asegurarse de que el programa en C++ podrá acceder a la información de forma exitosa.



Al ejecutar el programa desde Visual Studio Code, se establece correctamente la conexión con la base de datos 'CONTROL_PERSONAL'. En la terminal se muestra un mensaje de confirmación 'Conexión exitosa', seguido de la visualización de los registros almacenados en la tabla 'personal'.

Cada registro incluye el ID, nombre, edad y género del personal, lo que demuestra que la consulta a la base de datos fue exitosa y los datos se recuperaron correctamente. Este paso confirma que el sistema puede comunicarse eficazmente con MySQL y mostrar información relevante mediante el uso del ORM y la configuración previamente realizada.



The screenshot shows the Visual Studio Code interface with a C++ file named `prueba.cpp` open. The code uses EloquentORM to establish a connection to a MySQL database named 'CONTROL_PERSONAL' and retrieve all records from the 'personal' table. The terminal output shows the successful connection and the list of records.

```
int main() {  
    // ...  
    else {  
        cout << "¡conexión exitosa!" << endl;  
    }  
  
    // Definir las columnas que se quieren obtener de la tabla 'personas'  
    vector<string> columnas = { "NOMBRE", "EDAD", "GENERO" };  
  
    // Crear una instancia del ORM para la tabla 'personas'  
    EloquentORM personal(conn, "personal", columnas);  
  
    // Obtener todos los registros  
    auto lista = personal.getAll();  
  
    // Mostrar los registros en consola  
    for (auto& reg : lista) {  
        cout << "ID: " << reg["ID"]  
        << ", Nombre: " << reg["NOMBRE"]  
        << ", Edad: " << reg["EDAD"]  
    }  
}
```

Terminal Output:

```
PS C:\Users\USUARIO\Documents\Github> & 'c:\Users\USUARIO\.vscode\extensions\ms-vscode.cpptools-1.24.5-win32-x64\debugAdapters\bin\Win  
docsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-ambabzyo.oev' '--stdout=Microsoft-MIEngine-out-uaz0pvt.41e' '--stderr=Microsoft-  
MIEngine-Error-gpbq2tk.fno' '--pid=Microsoft-MIEngine-Pid-kolu0y2t.ht0' '--dbgExec=c:\Program Files\winlibs-x86_64-posix-seh-gcc-14.2.0  
-11va-19.1.7-mingw-w64ucrt-12.0.0-r3\mingw64\bin\gdb.exe' '--interpreter=i  
5b4-5f727cf36ea8jConexión exitosa!  
ID: 1, Nombre: JIMMY, Edad: 22, Género: MASCULINO  
ID: 2, Nombre: JATHE, Edad: 22, Género: MASCULINO  
ID: 3, Nombre: FREDY, Edad: 21, Género: MASCULINO  
ID: 4, Nombre: LUIS, Edad: 25, Género: MASCULINO  
ID: 5, Nombre: LUCY, Edad: 24, Género: FEMENINO  
ID: 6, Nombre: DONALD, Edad: 33, Género: MASCULINO  
ID: 7, Nombre: HECTOR, Edad: 32, Género: MASCULINO  
ID: 8, Nombre: JAZMIN, Edad: 22, Género: FEMENINO  
ID: 9, Nombre: JULIO, Edad: 21, Género: MASCULINO  
Presione una tecla para continuar . . .
```