

## Creando el repositorio en Github Desktop

The screenshot displays the GitHub Desktop application interface. The top bar includes menus for File, Edit, View, Repository, Branch, and Help. Below this, the 'Current repository' is set to 'control\_empleados\_cpp' and the 'Current branch' is 'main'. A 'Publish repository' button is visible, with the text 'Publish this repository to GitHub'.

The main area is divided into two sections. On the left, the 'History' tab is active, showing a list of commits. The first commit, 'Integración de mySQL con C++', is highlighted in blue and was made by Ariel Orellana 22 hours ago. Below this, a list of files changed in the commit is shown, including .vscode\c\_cpp\_properties.json, .vscode\launch.json, .vscode\tasks.json, base\_datos\EloquentORM.h, base\_datos\MySQLConexion.h, base\_datos\MySQLModel.h, base\_datos\comandos.sql, base\_datos\control\_horario\_base\_datos.sql, base\_datos\libmysql.dll, base\_datos\main.cpp, base\_datos\main.exe, base\_datos\others\README.md, base\_datos\others\io.png, and base\_datos\others\output\hello.exe.

On the right, the 'Integración de mySQL con C++' commit is selected, showing the configuration for Visual Studio Code. The file .vscode\c\_cpp\_properties.json is open, displaying the following JSON configuration:

```
@@ -0,0 +1,21 @@
1 + {
2 +   "configurations": [
3 +     {
4 +       "name": "Win32",
5 +       "includePath": [
6 +         "${workspaceFolder}/**"
7 +       ],
8 +       "defines": [
9 +         "_DEBUG",
10 +         "UNICODE",
11 +         "_UNICODE"
12 +       ],
13 +       "windowsSdkVersion": "10.0.22621.0",
14 +       "compilerPath": "cl.exe",
15 +       "cStandard": "c17",
16 +       "cppStandard": "c++17",
17 +       "intelliSenseMode": "windows-msvc-x64"
18 +     }
19 +   ],
20 +   "version": 4
21 + }
```

## Uso de HeidiSQL para ingresar registros y exportar la base de datos

Unamed\control\_horario\empleados\ - HeidiSQL 12.10.0.7000

Archivo Editar Buscar Consulta Herramientas Ira Ayuda

Filtro de base Filtro de tabla

Unamed Base de datos: control\_horario Tabla: empleados Datos comandos comandline.sql control\_horario\_base\_datos.sql

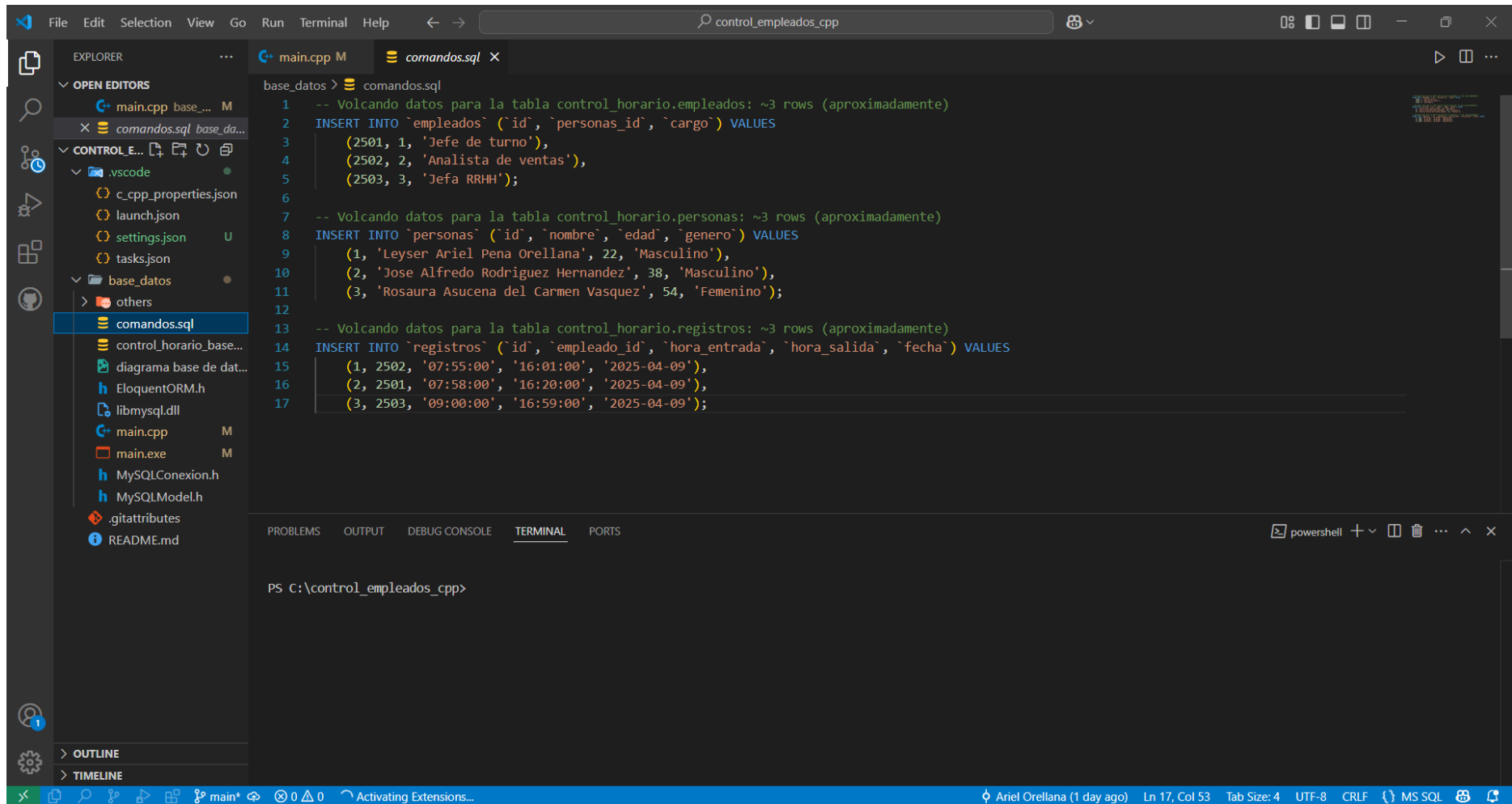
Nombre	Filas	Tamaño	Creado	Actualizado	Motor	Comentario	Tipo
empleados	3	32.0 KiB	2025-04-08 22:27:29	2025-04-09 00:03:50	InnoDB		Table
personas	3	16.0 KiB	2025-04-05 10:02:20	2025-04-08 23:58:17	InnoDB		Table
registros	3	32.0 KiB	2025-04-08 22:35:52	2025-04-09 00:10:37	InnoDB		Table

```
8 /* Conectado. ID de Hilo: 83 */
9 /* Reading function definitions from C:\Program Files\HeidiSQL\functions-mysql8.ini */
10 SHOW TABLES FROM `information_schema`;
11 SHOW DATABASES;
12 SHOW OPEN TABLES FROM control_horario WHERE `in_use`!=0;
13 USE `control_horario`;
14 /* Entrando a la sesión "Unamed" */
15 SELECT `DEFAULT_COLLATION_NAME` FROM `information_schema`.`SCHEMATA` WHERE `SCHEMA_NAME`='control_horario';
16 SHOW TABLE STATUS FROM `control_horario`;
17 SHOW FUNCTION STATUS WHERE `Db`='control_horario';
18 SHOW PROCEDURE STATUS WHERE `Db`='control_horario';
19 SHOW TRIGGERS FROM `control_horario`;
20 SELECT *, EVENT_SCHEMA AS `Db`, EVENT_NAME AS `Name` FROM information_schema.`EVENTS` WHERE `EVENT_SCHEMA`='control_horario';
21 /* Cargando archivo "C:\Users\lp24k\Documents\HeidiSQL\projects\05_04\comandos comandline.sql" (403 B) en pestaña de consulta #1 */
22 /* Cargando archivo "C:\control_empleados_cpp\base_datos\control_horario_base_datos.sql" (3.4 KiB) en pestaña de consulta #2 */
23 /* Escalando controles a DPI de pantalla: 125% */
24 SELECT * FROM `information_schema`.`COLUMNS` WHERE TABLE_SCHEMA='control_horario' AND TABLE_NAME='empleados' ORDER BY ORDINAL_POSITION;
25 SHOW INDEXES FROM `empleados` FROM `control_horario`;
26 SELECT * FROM information_schema.REFERENTIAL_CONSTRAINTS WHERE CONSTRAINT_SCHEMA='control_horario' AND TABLE_NAME='empleados' AND REFERENCED_TABLE_NAME IS NOT NULL;
27 SELECT * FROM information_schema.KEY_COLUMN_USAGE WHERE TABLE_SCHEMA='control_horario' AND TABLE_NAME='empleados' AND REFERENCED_TABLE_NAME IS NOT NULL;
28 SHOW COLLATION;
29 SHOW ENGINES;
30 SHOW CREATE TABLE `control_horario`.`empleados`;
31 SELECT tc.CONSTRAINT_NAME, cc.CHECK_CLAUSE FROM `information_schema`.`CHECK_CONSTRAINTS` AS cc, `information_schema`.`TABLE_CONSTRAINTS` AS tc WHERE tc.CONSTRAINT_SCHEMA='control_horario' AND tc.TABLE_NAME='empleados' AND tc.CONST
```

control\_horario: 3 tables

Conectado: 00:00 h MySQL 8.0.41 Activo durante: 5 días, 13:21 h Hora del servidor: 22 Preparado.

## Comandos de ingreso de registros en SQL

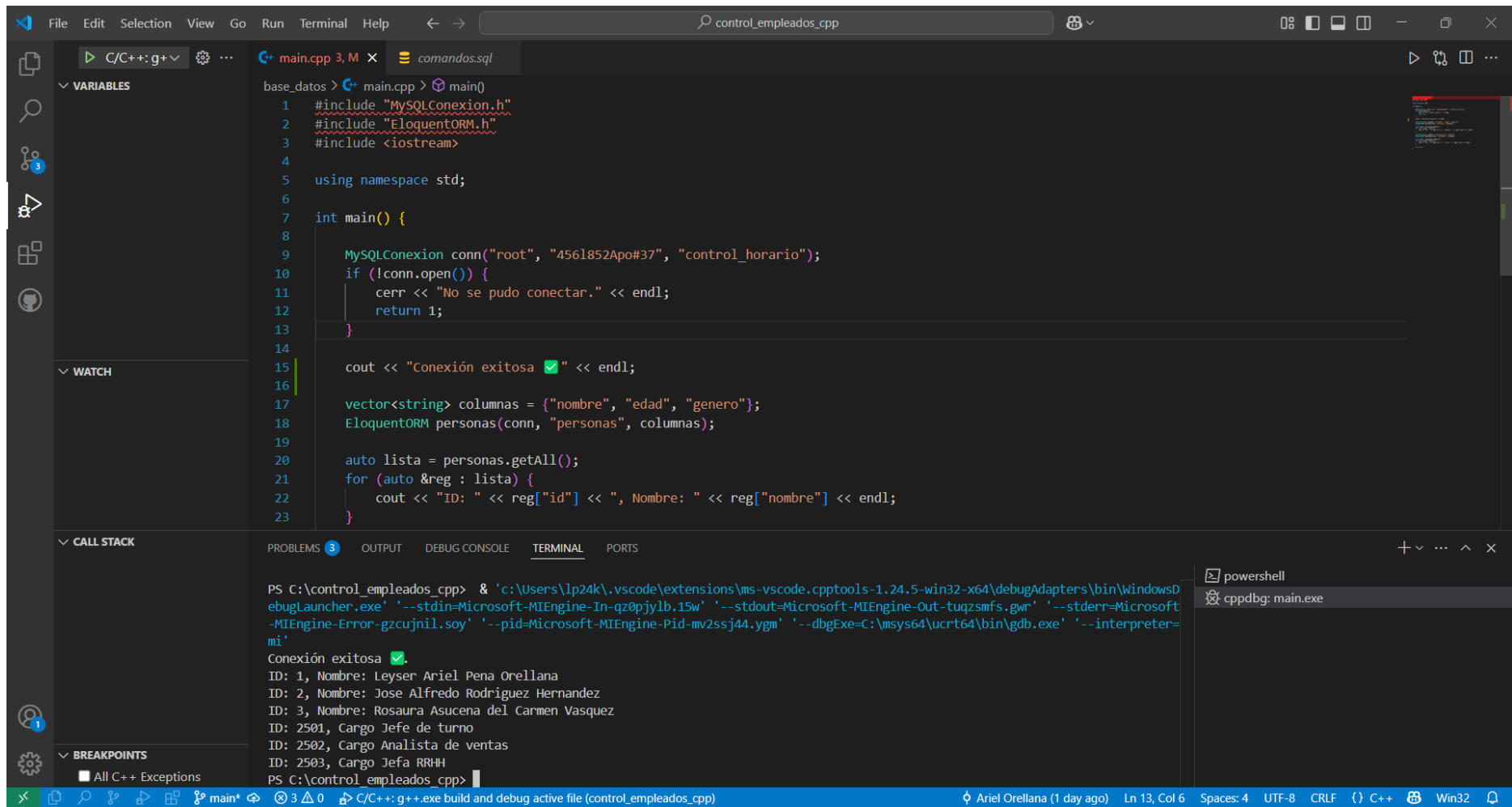


The image shows a Visual Studio Code editor window with a dark theme. The Explorer sidebar on the left displays a project structure with folders like 'base\_datos' and 'others', and files such as 'comandos.sql', 'main.cpp', and 'main.exe'. The 'comandos.sql' file is open in the main editor, showing three SQL INSERT statements. The first statement inserts data into the 'empleados' table, the second into the 'personas' table, and the third into the 'registros' table. The bottom panel shows the 'TERMINAL' tab with a PowerShell prompt at 'PS C:\control\_empleados\_cpp>'. The status bar at the bottom indicates the active file is 'main\*', the cursor is at line 17, column 53, and the encoding is UTF-8.

```
base_datos > comandos.sql
1  -- Volcando datos para la tabla control_horario.empleados: ~3 rows (aproximadamente)
2  INSERT INTO `empleados` (`id`, `personas_id`, `cargo`) VALUES
3      (2501, 1, 'Jefe de turno'),
4      (2502, 2, 'Analista de ventas'),
5      (2503, 3, 'Jefa RRHH');
6
7  -- Volcando datos para la tabla control_horario.personas: ~3 rows (aproximadamente)
8  INSERT INTO `personas` (`id`, `nombre`, `edad`, `genero`) VALUES
9      (1, 'Leyser Ariel Pena Orellana', 22, 'Masculino'),
10     (2, 'Jose Alfredo Rodriguez Hernandez', 38, 'Masculino'),
11     (3, 'Rosaura Asucena del Carmen Vasquez', 54, 'Femenino');
12
13 -- Volcando datos para la tabla control_horario.registros: ~3 rows (aproximadamente)
14 INSERT INTO `registros` (`id`, `empleado_id`, `hora_entrada`, `hora_salida`, `fecha`) VALUES
15     (1, 2502, '07:55:00', '16:01:00', '2025-04-09'),
16     (2, 2501, '07:58:00', '16:20:00', '2025-04-09'),
17     (3, 2503, '09:00:00', '16:59:00', '2025-04-09');
```

PS C:\control\_empleados\_cpp>

## Integración de SQL con C++



The screenshot displays the Visual Studio Code editor with a C++ project named 'control\_empleados\_cpp'. The editor shows the 'main.cpp' file with the following code:

```
base_datos > C++ main.cpp > main()
1  #include "MySQLConexion.h"
2  #include "EloquentORM.h"
3  #include <iostream>
4
5  using namespace std;
6
7  int main() {
8
9      MySQLConexion conn("root", "4561852Apo#37", "control_horario");
10     if (!conn.open()) {
11         cerr << "No se pudo conectar." << endl;
12         return 1;
13     }
14
15     cout << "Conexión exitosa ✓" << endl;
16
17     vector<string> columnas = {"nombre", "edad", "genero"};
18     EloquentORM personas(conn, "personas", columnas);
19
20     auto lista = personas.getAll();
21     for (auto &reg : lista) {
22         cout << "ID: " << reg["id"] << ", Nombre: " << reg["nombre"] << endl;
23     }
24 }
```

The left sidebar shows the 'VARIABLES' and 'WATCH' panels. The bottom panel shows the 'TERMINAL' output:

```
PS C:\control_empleados_cpp> & 'c:\Users\lp24k\.vscode\extensions\ms-vscode.cpptools-1.24.5-win32-x64\debugAdapters\bin\WindowsD
ebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-qz0pjylb.15w' '--stdout=Microsoft-MIEngine-Out-tuqzsmfs.gwn' '--stderr=Microsoft
-MIEngine-Error-gzcuynil.15w' '--pid=Microsoft-MIEngine-Pid-mv2ssj44.ygm' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=
mi'
Conexión exitosa ✓
ID: 1, Nombre: Leyser Ariel Pena Orellana
ID: 2, Nombre: Jose Alfredo Rodriguez Hernandez
ID: 3, Nombre: Rosaura Asucena del Carmen Vasquez
ID: 2501, Cargo Jefe de turno
ID: 2502, Cargo Analista de ventas
ID: 2503, Cargo Jefa RRHH
PS C:\control_empleados_cpp>
```

The status bar at the bottom indicates the file is 'main.cpp' and the build and debug active file is 'control\_empleados\_cpp'.