

## Actividad | #2 | Conexión con la BD y

### Creación de Tablas

#### Lenguajes de Programación II

Ingeniería en Desarrollo de Software



TUTOR: Félix Acosta Hernández

ALUMNO: Erick Alfredo Quiroz Figueroa

FECHA: 29/11/2024

## Contenido

<b>Introducción .....</b>	<b>3</b>
<b>Descripción .....</b>	<b>4</b>
<b>Justificación .....</b>	<b>5</b>
<b>Desarrollo.....</b>	<b>6</b>
<b>Conexión.....</b>	<b>6</b>
<b>Tablas .....</b>	<b>7</b>
<b>Código.....</b>	<b>8</b>
<b>Conclusión .....</b>	<b>13</b>
<b>Bibliografía .....</b>	<b>14</b>

## **Introducción**

En esta actividad conoceremos acerca de la conexión de bases de datos y creación de tablas en un entorno visual con el cual se está programando dicha Base de Datos.

Uno de los puntos importantes en la realización de dicha base es el lenguaje con el cual se está programando y con cual lenguaje es que se quiere realizar dicha conexión.

Como sabemos para poder crear una base de datos es necesario conocer el lenguaje SQL que por sus siglas en ingles se llama *Lenguaje de Consulta Estructurada*, y este, a su vez tiene una sintaxis sencilla de aprender para la creación de las tablas y consultas que se necesiten realizar.

Ahora una vez creadas dichas tablas las conectaremos con un lenguaje de programación, en este caso usaremos el lenguaje C++, el cual es un lenguaje de alto nivel sencillo de manejar y con una sintaxis amigable que permite al programador poder interactuar con él.

### **Descripción**

Como pudimos apreciar en la actividad anterior conocimos como es que se crean las tablas en el lenguaje SQL en un entorno grafico especializado para la creación de tablas.

Ahora esas mismas tablas se usarán para realizar una conexión con una IDE que nos permite programar en el lenguaje C++ y hacer conexiones de SQL para ver y conocer como es que este trabaja.

Para ello, es importante conocer el lenguaje y el código exacto para tener una conexión fiable y sin errores o con la cantidad mínima de errores que se puedan tener al momento de hacer una conexión entre dos lenguajes para la realización de las tablas en una base de datos.

Repasando, el lenguaje C++ es un lenguaje de programación de alto nivel que nos ayuda a entender de manera mas concreta la Programación Orientada a Objetos, que comprende todo lo que tiene que ver con métodos, clases, herencia, polimorfismo, entre otros.

### **Justificación**

El propósito de esta actividad es adentrarnos en el mundo de las bases de datos y en la programación orientada a objetos.

Saber y conocer que estos dos conceptos se pueden combinar y que están presentes día con día en el mundo laboral y digital.

Es importante conocer muy bien el lenguaje en el que se vaya a realizar la base de datos que en este caso es un lenguaje de consulta y también a su vez conocer un poco del lenguaje en el que se pretende conectar para crear de manera grafica las tablas antes mencionadas del mismo sistema o de la base como tal.

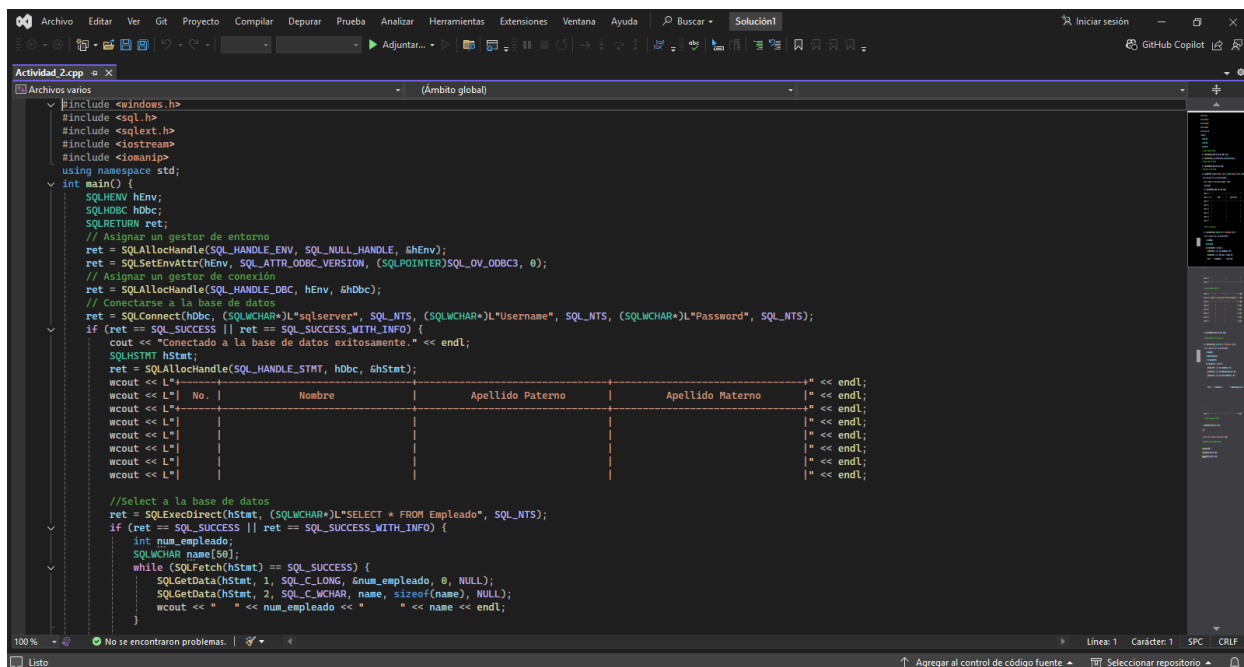
Centrándonos en lo que estamos por realizar, podemos decir que aprenderemos mucho realizando este ejercicio y que si nos lo permiten las circunstancias, realizarlo a menudo o dedicarnos a esto de lleno.

Una vez dicho esto pasamos a la demostración del ejercicio o el proyecto como tal para ver como es que va quedando la conexión entre los lenguajes de programación utilizados.

## Desarrollo

### Conexión

En este apartado se muestra la conexión de la base de datos del código en C++ con SQL.



```

#include <windows.h>
#include <sql.h>
#include <sql.h>
#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    SQLHENV hEnv;
    SQLHDBC hDbc;
    SQLRETURN ret;

    // Asignar un gestor de entorno
    ret = SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &hEnv);
    ret = SQLSetEnvAttr(hEnv, SQL_ATTR_ODBC_VERSION, (SQLPOINTER)SQL_OV_ODBC3, 0);

    // Asignar un gestor de conexión
    ret = SQLAllocHandle(SQL_HANDLE_DBC, hEnv, &hDbc);

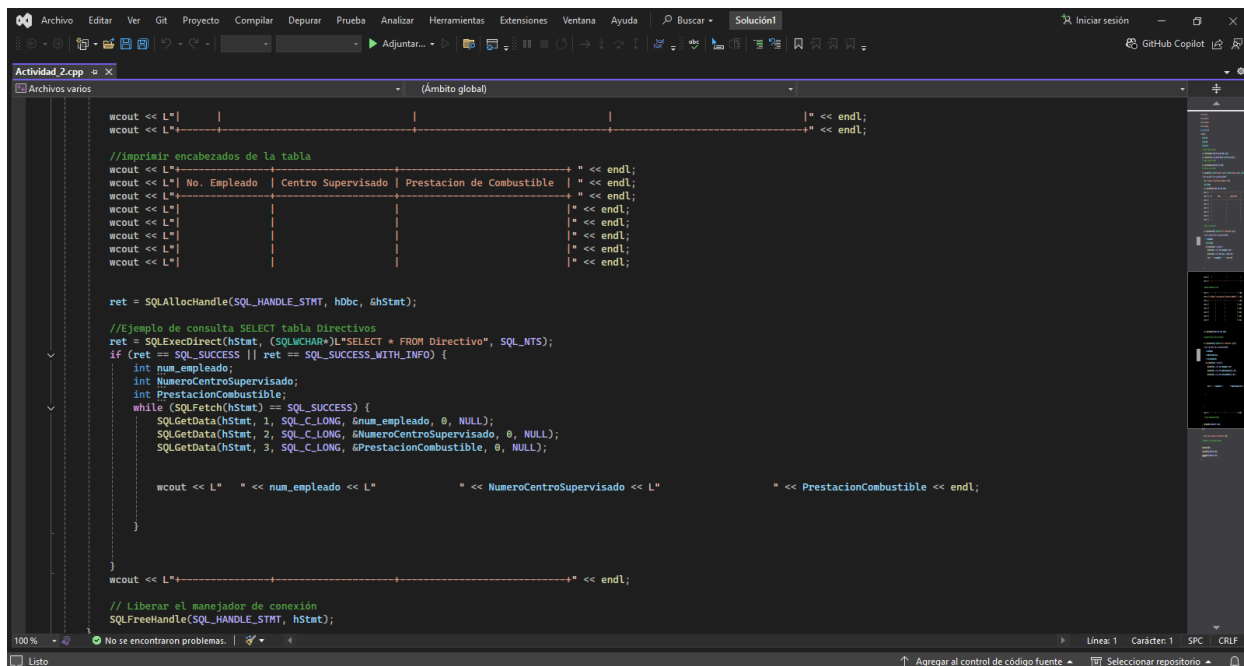
    // Conectarse a la base de datos
    ret = SQLConnect(hDbc, (SQLWCHAR*)L"sqlserver", SQL_NTS, (SQLWCHAR*)L"Username", SQL_NTS, (SQLWCHAR*)L"Password", SQL_NTS);
    if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
        cout << "Conectado a la base de datos exitosamente." << endl;

        SQLHSTMT hStmt;
        ret = SQLAllocHandle(SQL_HANDLE_STMT, hDbc, &hStmt);

        wcout << L" | " << endl;
        wcout << L" | " << endl;
        wcout << L" | " << endl;
        wcout << L" | " << endl;
        wcout << L" | " << endl;
        wcout << L" | " << endl;
        wcout << L" | " << endl;

        //Select a la base de datos
        ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT * FROM Empleado", SQL_NTS);
        if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
            int num_empleado;
            SQLWCHAR name[50];
            while (SQLFetch(hStmt) == SQL_SUCCESS) {
                SQLGetData(hStmt, 1, SQL_C_LONG, &num_empleado, 0, NULL);
                SQLGetData(hStmt, 2, SQL_C_WCHAR, name, sizeof(name), NULL);
                wcout << L" | " << num_empleado << L" | " << name << endl;
            }
        }
    }
}

```



```

wcout << L" | " << endl;
wcout << L" | " << endl;

//Imprimir encabezados de la tabla
wcout << L" | " << endl;
wcout << L" | " << endl;
wcout << L" | " << endl;
wcout << L" | " << endl;
wcout << L" | " << endl;
wcout << L" | " << endl;
wcout << L" | " << endl;

ret = SQLAllocHandle(SQL_HANDLE_STMT, hDbc, &hStmt);

//Ejemplo de consulta SELECT tabla Directivos
ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT * FROM Directivo", SQL_NTS);
if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
    int num_empleado;
    int NumeroCentroSupervisado;
    int PrestacionCombustible;
    while (SQLFetch(hStmt) == SQL_SUCCESS) {
        SQLGetData(hStmt, 1, SQL_C_LONG, &num_empleado, 0, NULL);
        SQLGetData(hStmt, 2, SQL_C_LONG, &NumeroCentroSupervisado, 0, NULL);
        SQLGetData(hStmt, 3, SQL_C_LONG, &PrestacionCombustible, 0, NULL);

        wcout << L" | " << num_empleado << L" | " << NumeroCentroSupervisado << L" | " << PrestacionCombustible << endl;
    }
    wcout << L" | " << endl;

    // Liberar el manejador de conexión
    SQLFreeHandle(SQL_HANDLE_STMT, hStmt);
}

```

```

wcout << L" |" << endl;

ret = SQLAllocHandle(SQL_HANDLE_STMT, hDbc, &hStmt);

//Ejemplo de consulta SELECT tabla Directivos
ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT * FROM Directivo", SQL_NTS);
if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
    int num_empleado;
    int NumeroCentroSupervisado;
    int PrestacionCombustible;
    while (SQLFetch(hStmt) == SQL_SUCCESS) {
        SQLGetData(hStmt, 1, SQL_C_LONG, &num_empleado, 0, NULL);
        SQLGetData(hStmt, 2, SQL_C_LONG, &NumeroCentroSupervisado, 0, NULL);
        SQLGetData(hStmt, 3, SQL_C_LONG, &PrestacionCombustible, 0, NULL);

        wcout << L" " << num_empleado << L" " << NumeroCentroSupervisado << L" " << PrestacionCombustible << endl;
    }

    wcout << L"-----" << endl;

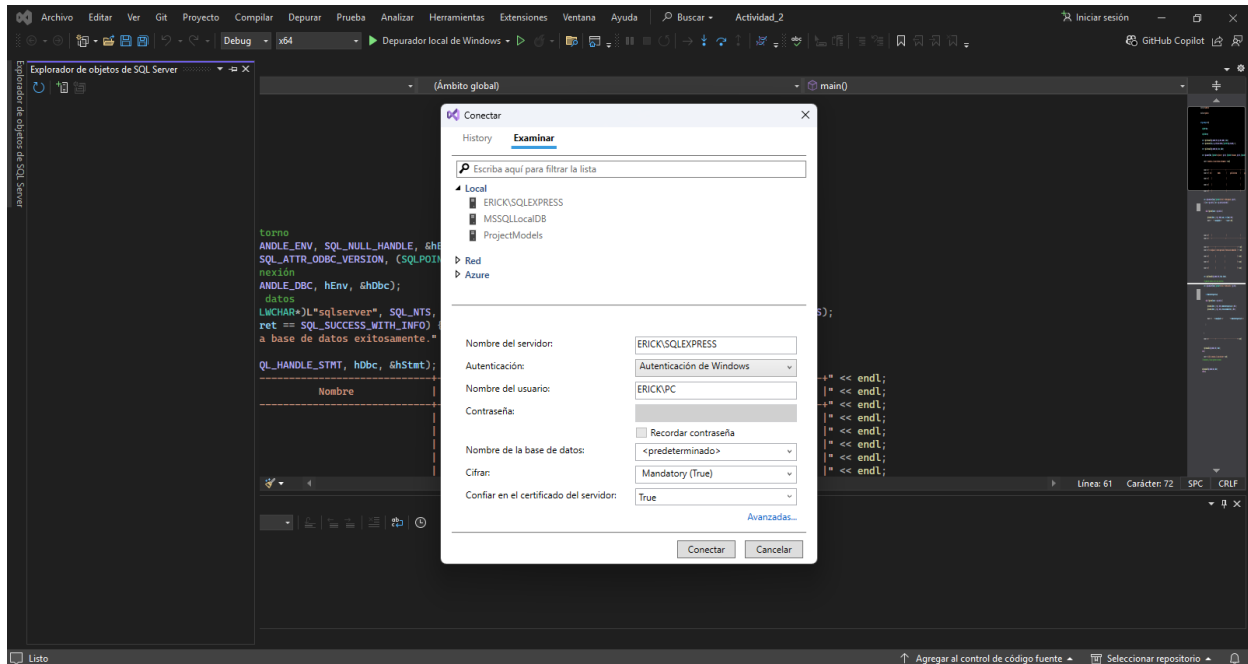
    // Liberar el manejador de conexión
    SQLFreeHandle(SQL_HANDLE_STMT, hStmt);
}
else {
    cout << "Fallo la conexión a la base de datos" << endl;
}

// Desconectar y liberar gestores de entorno
SQLDisconnect(hDbc);
SQLFreeHandle(SQL_HANDLE_DBC, hDbc);
SQLFreeHandle(SQL_HANDLE_ENV, hEnv);
return 0;
}

```

## Tablas

Como siguiente punto se demostrarán la conexión exitosa y la compilación de las tablas.



Consola de depuración de Mi

Conectado a la base de datos exitosamente.

No.	Nombre	Apellido Paterno	Apellido Materno

No. Empleado	Centro Supervisado	Prestacion de Combustible

E:\Carrera\Programacion II\Archivos\Actividad\_2\x64\Debug\Actividad\_2.exe (proceso 18164) se cerró con el código 0 (0x0)

Presione cualquier tecla para cerrar esta ventana. . .|

### Código

```
#include <windows.h>
#include <sql.h>
#include <sqlext.h>
#include <iostream>
#include <iomanip>
using namespace std;
int main() {
    SQLHENV hEnv;
    SQLHDBC hDbc;
    SQLRETURN ret;
    // Asignar un gestor de entorno
    ret = SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &hEnv);
    ret = SQLSetEnvAttr(hEnv, SQL_ATTR_ODBC_VERSION,
(SQLPOINTER)SQL_OV_ODBC3, 0);
    // Asignar un gestor de conexión
    ret = SQLAllocHandle(SQL_HANDLE_DBC, hEnv, &hDbc);
    // Conectarse a la base de datos
    ret = SQLConnect(hDbc, (SQLWCHAR*)L"sqlserver", SQL_NTS,
(SQLWCHAR*)L"Username", SQL_NTS, (SQLWCHAR*)L>Password", SQL_NTS);
    if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
        cout << "Conectado a la base de datos exitosamente." << endl;
    }
}
```





```

        wcout << L"+-----+-----+-----+
-----+ " << endl;
        wcout << L"|           |           |
|" << endl;
        wcout << L"|           |           |
|" << endl;
        wcout << L"|           |           |
|" << endl;
        wcout << L"|           |           |
|" << endl;
        wcout << L"|           |           |
|" << endl;

    ret = SQLAllocHandle(SQL_HANDLE_STMT, hDbc, &hStmt);

    //Ejemplo de consulta SELECT tabla Directivos
    ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT * FROM
Directivo", SQL_NTS);
    if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
        int num_empleado;
        int NumeroCentroSupervisado;
        int PrestacionCombustible;
        while (SQLFetch(hStmt) == SQL_SUCCESS) {
            SQLGetData(hStmt, 1, SQL_C_LONG, &num_empleado, 0,
NULL);
            SQLGetData(hStmt, 2, SQL_C_LONG,
&NumeroCentroSupervisado, 0, NULL);
            SQLGetData(hStmt, 3, SQL_C_LONG,
&PrestacionCombustible, 0, NULL);

                wcout << L"    " << num_empleado << L"    "
<< NumeroCentroSupervisado << L"    " <<
PrestacionCombustible << endl;

        }

    }

    wcout << L"+-----+-----+-----+
-----+" << endl;

    // Liberar el manejador de conexión
    SQLFreeHandle(SQL_HANDLE_STMT, hStmt);

```

```

    }
    else {
        cout << "Fallo la conexion a la base de datos" << endl;
    }
    // Desconectar y liberar gestores de entorno
    SQLDisconnect(hDbc);
    SQLFreeHandle(SQL_HANDLE_DBC, hDbc);
    SQLFreeHandle(SQL_HANDLE_ENV, hEnv);
    return 0;
}

```

Como primera instancia se declaran las librerías en la consola.

```
#include <windows.h>
#include <sql.h>
#include <sqlext.h>
#include <iostream>
#include <iomanip>
using namespace std;
```

Posteriormente se genera el main declarando las variables a utilizar para la conexión

```
int main() {
    SQLHENV hEnv;
    SQLHDBC hDbc;
    SQLRETURN ret;
```

Se asigna el gestor de datos tanto de entorno como de conexión

```
// Asignar un gestor de entorno
ret = SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &hEnv);
ret = SQLSetEnvAttr(hEnv, SQL_ATTR_ODBC_VERSION, (SQLPOINTER)SQL_OV_ODBC3, 0);
// Asignar un gestor de conexión
ret = SQLAllocHandle(SQL_HANDLE_DBC, hEnv, &hDbc);
```

Se procede a hacer la conexión a la base de datos

[illegible]

Seleccionamos la base de datos

```
//Select a la base de datos
ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT * FROM Empleado", SQL_NTS);
if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
    int num_empleado;
    SQLWCHAR name[50];
    while (SQLFetch(hStmt) == SQL_SUCCESS) {
        SQLGetData(hStmt, 1, SQL_C_LONG, &num_empleado, 0, NULL);
        SQLGetData(hStmt, 2, SQL_C_WCHAR, name, sizeof(name), NULL);
        wcout << " " << num_empleado << " " << name << endl;
    }
}

wcout << L"| " << endl;
wcout << L"+-----+-----+-----+-----+ " << endl;
```

Hacemos las consultas

```
//Select a la base de datos
ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT * FROM Empleado", SQL_NTS);
if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
    int num_empleado;
    SQLWCHAR name[50];
    while (SQLFetch(hStmt) == SQL_SUCCESS) {
        SQLGetData(hStmt, 1, SQL_C_LONG, &num_empleado, 0, NULL);
        SQLGetData(hStmt, 2, SQL_C_WCHAR, name, sizeof(name), NULL);
        wcout << " " << num_empleado << " " << name << endl;
    }
}

wcout << L"| " << endl;
wcout << L"+-----+-----+-----+-----+ " << endl;
```

Liberamos memoria y conexiones

```
// Liberar el manejador de conexión
SQLFreeHandle(SQL_HANDLE_STMT, hStmt);
}
else {
    cout << "Fallo la conexion a la base de datos" << endl;
}
// Desconectar y liberar gestores de entorno
SQLDisconnect(hDbc);
SQLFreeHandle(SQL_HANDLE_DBC, hDbc);
SQLFreeHandle(SQL_HANDLE_ENV, hEnv);
return 0;
```

## **Conclusión**

Como pudimos apreciar los comandos y las instrucciones que se programan en el entorno IDE para la conexión son específicas y exclusivas del lenguaje para una buena comunicación entre ambos lenguajes de programación.

Se mostro parte por parte como es que funciona el código programado en C++ para que tenga una conexión limpia y estable con el lenguaje de consulta SQL, que es donde creamos nuestras tablas y bases de datos.

Una vez generado el código se explica imagen por imagen como es que funciona cada línea y que hace cada parte de dicho código para entender de manera mas sencilla y no tengamos problemas de comprensión.

Esto nos ayudara a en un futuro saber como es que se programa y saber cómo es el código que se necesita para hacer una conexión de SQL con C++ usando programación orientada a objetos y su sintaxis amigable con el programador o usuario.

### **Bibliografia**

Link de GitHub: [https://github.com/Erick123Alf/Programacion\\_II/tree/main/Actividad\\_2](https://github.com/Erick123Alf/Programacion_II/tree/main/Actividad_2)