

## Actividad | 2 | Conexión y tablas

### Nombre del curso

Ingeniería en Desarrollo de Software



TUTOR: Félix Acosta Hernández.

ALUMNO: Uziel de Jesús López Ornelas.

FECHA: 30 de Noviembre del 2024.

**Índice**

Introducción ..... 1

Descripción..... 1

Justificación..... 2

Desarrollo ..... 3

Conexión ..... 3

Tablas ..... 6

Código ..... 8

Conclusión..... 9

Referencias..... 10

## Introducción

En esta segunda actividad se creará la conexión de nuestro base de datos al lenguaje de programación en “C++”, para visualizar las tablas que nosotros tengamos registradas en el software “Microsoft SQL Server “, para ello se realizará lo siguiente de acuerdo a lo que la actividad lo pide, se instalará un entorno de trabajo para que este pueda ser utilizado en el lenguaje de “C++”, una vez dentro del entorno de trabajo se tiene que realizar la conexión de datos creada en la actividad número “1” que es en donde nosotros colocamos toda la información que se nos pedía en aquel entonces, para esta nueva actividad se tiene que descargar o tener el software de “Visual Studio” es aquel que nos permitirá trabajar en la consola de “C++”, anteriormente se ha trabajado con el programa antes mencionado así que en lo caso no es necesaria la instalación nuevamente, a lo largo de la actividad se mostrará toda la información en capturas de pantalla para mostrar el avance que se tiene con una breve explicación de lo que se significa cada captura de pantalla.

## Descripción

El “ODBC” por sus siglas en inglés “Open Database Connectivity” es un protocolo en el cual permite a las aplicaciones acceder a datos que se encuentran en las bases de datos de diferentes sistemas, gracias a ello se puede tener casi cualquier base de datos siempre y cuando se tenga un controlador de “ODBS”, algunos de los componentes que se tiene en “ODBS” son los siguientes:

- El API de ODBC.
- Administrador de controles de ODBC.
- Controladores de base de datos ODBC.
- Biblioteca de cursores ODBC.
- Administrador de ODBC.

Estos son los componentes que posee este software y que gracias a ello logramos tener acceso a todo tipo de funciones mediante las herramientas disponibles. Las clases de datos admiten es su totalidad o en su mayoría cualquier origen de datos para que el controlador de “ODBC”, en donde también una hoja de cálculo de Microsoft Excel o un archivo de texto pueden funcionar sin problemas con este software y

todo esto lo administra los controladores para que la conexión a las bases de datos sea precisa y eficiente, manteniendo la información certera y disponible en cualquier momento para su revisión.

### **Justificación**


El “ODBC” como ya lo mencionamos es muy útil para acceder a la información de una base de datos que esté vinculada por medio de todas las herramientas y controladores que este software les ofrece, y que sabemos que como en todo hay ventajas que son de admirar sobre este programa y que es importante mencionarlas ya que nos ayuda a conocer mejor sus funciones y utilidades. Las ventajas más notorias son las siguientes:

- ODBC tiene una interfaz simple, gracias a ello podemos usarla de manera sencilla y a la hora de querer configurarla no tenemos muchos problemas ya que es fácil de utilizar.
- Es multiplataforma, esto quiere decir que es compatible con muchos sistemas operativos, así como sus orígenes de datos, puede conectarse a una gran variedad de aplicaciones sin mayor problema volviéndolo accesible para un uso cotidiano.
- Acceso a una gran variedad de datos, como ya se mencionó anteriormente es una sola interfaz podemos acceder a mucha información que esté resguardada y que se quiera consultar en cualquier momento, proporcionando eficiencia y eficacia.
- Es una herramienta fundamental para un desarrollador, gracias a que posee muchas herramientas dirigidas a la tecnología en software un desarrollador tendrá un festín de consultas e improvisaciones para acceder y disfrutar de todas las características que ofrece el software ODBC.

## Desarrollo

### Conexión:

Para realizar la conexión de nuestra base de datos con el software “ODBC” tenemos que crear primero una base de datos en “Microsoft SQL Server” como la que tenemos actualmente:

 **Registro\_de\_Empleados**

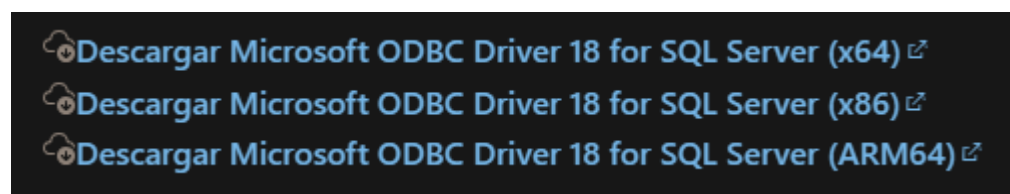
Después de ello tenemos que ir al link para descargar “ODBC”:

learn.microsoft.com › es-es › sql ▾

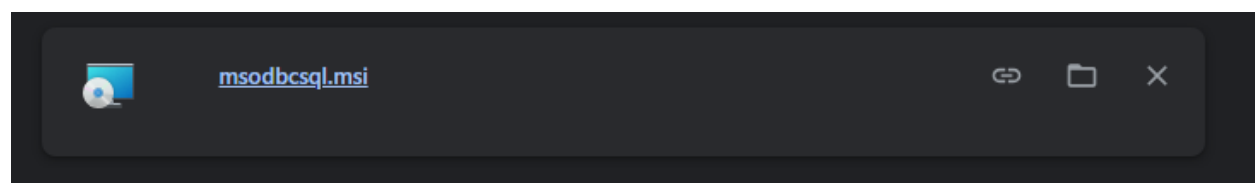
**Descargar controlador ODBC para SQL Server - ODBC Driver for ...** ✓

24 de sept. de 2024 · Aprenda cómo descargar e instalar el controlador **ODBC** para conectarse a SQL Server desde aplicaciones que usan API de código nativo. Elige entre las versiones 18 y 17,...

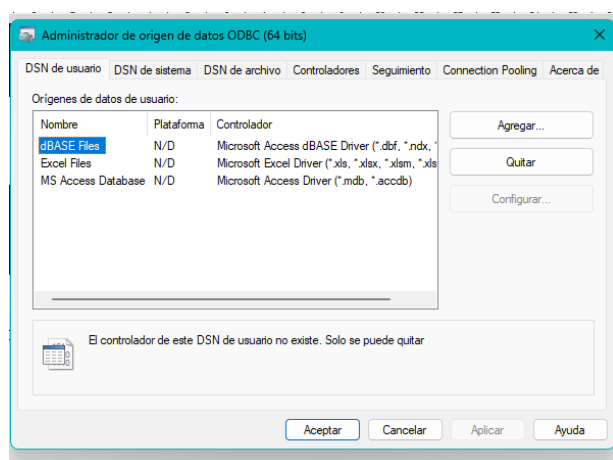
Al abrirlo nos saldrá una ventana en la que tendremos que elegir la versión de acuerdo a nuestro sistema operativo:



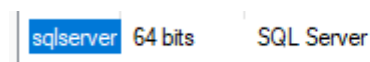
En mi caso descargaré la versión que está primero, se nos iniciará la descarga de un archivo que es el instalador del “ODBC”:



Ya que se nos descargó el archivo, tendremos este software en el que nos permitirá abrir el “ODBC”:



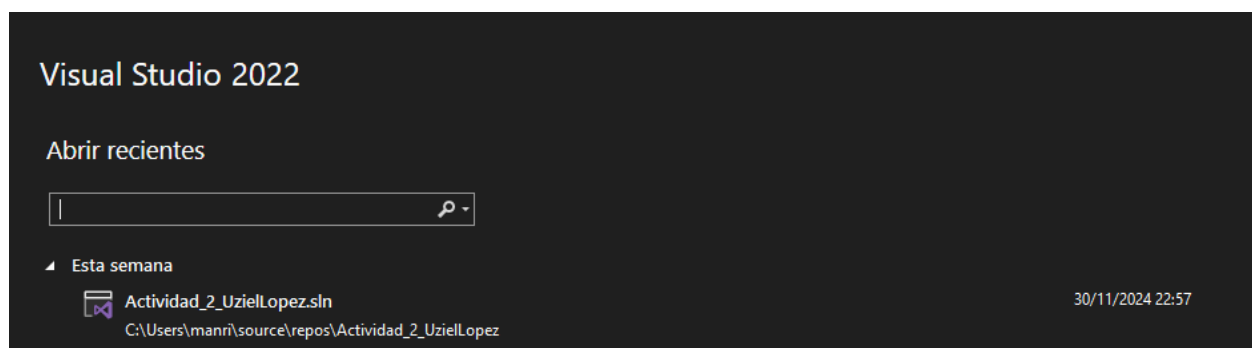
En el apartado de “DSN de sistema en donde colocaremos nuestra base de datos que esta vinculada a “Microsoft SQL Server”:



En mi caso ya la tengo instalada de acuerdo con los pasos que se dieron, en el que tenemos que darle un nombre que en mi caso fue sencillo, “sqlserver” y saber el nombre de nuestro servidor que se relaciona a la base de datos:



Cuando tengamos lista nuestra vinculación nos dirigiremos a “Visual Studio” para que con el lenguaje de programación en “C++” logremos vincular nuestra base de datos, para ello abrimos el software:



El código de vinculación es el siguiente, incluimos las librerías:

```

1  // Actividad_2_UzielLopez.conexión de la base de datos en SQL
2  //
3
4  #include <windows.h>
5  #include <sql.h>
6  #include <sqlext.h>
7  #include <iostream>

```

La sintaxis correcta para empezar a generar los códigos, manteniendo los comandos para estar enlazando el “ODBC” con “SQL”:

```

9  using namespace std;
10
11  int main() {
12      SQLHENV hEnv;
13      SQLHDBC hDbc;
14      SQLRETURN ret;
15
16      //Asignar un gestor de entorno
17      ret = SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &hEnv);
18      ret = SQLSetEnvAttr(hEnv, SQL_ATTR_ODBC_VERSION, (SQLPOINTER)SQL_OV_ODBC3, 0);
19
20      //Asignar un gestor de la conexión
21      ret = SQLAllocHandle(SQL_HANDLE_DBC, hEnv, &hDbc);
22
23      //Conectarse a la base de datos
24      ret = SQLConnect(hDbc, (SQLWCHAR*)L"sqlserver", SQL_NTS, (SQLWCHAR*)L"Username", SQL_NTS, (SQLWCHAR*)L"Password", SQL_NTS);
25

```

En la línea número “24” coloque el nombre de mi base de datos que le asigne al “ODBC” que es “sqlserver” un nombre sencillo para no olvidarlo fácilmente:

```

23      //Conectarse a la base de datos
24      ret = SQLConnect(hDbc, (SQLWCHAR*)L"sqlserver",

```

Si la conexión de datos fue correcta nos mostrara el mensaje que nos hemos conectado a la base de datos exitosamente:

```

26      if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
27          cout << "Conectado a la base de datos exitosamente." << endl;
28      }

```

Caso contrario se nos mandara le siguiente mensaje:

```

30      else {
31          cout << "Fallo la conexión de la base de datos." << endl;
32      }
33

```

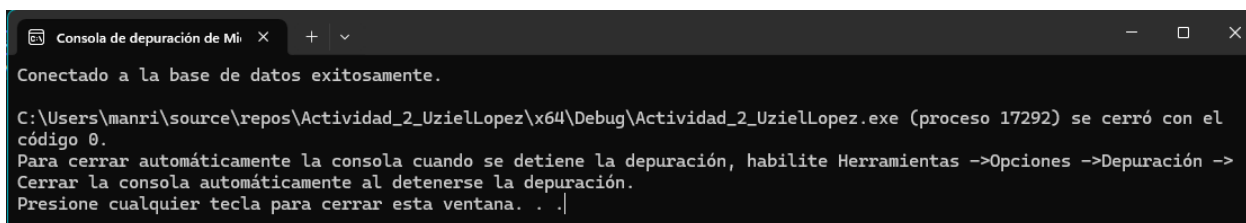
Tenemos los comandos para desconectar y liberar los manejadores que teníamos asignados en nuestro código de “C++”:

```
34      //Desconectar y liberar manejadores
35      SQLDisconnect(hDbc);
36      SQLFreeHandle(SQL_HANDLE_DBC, hDbc);
37      SQLFreeHandle(SQL_HANDLE_ENV, hEnv);
38
```

Y como siempre cerramos el comando por medio del código “Return”:

```
39      return 0;
40  }
```

Ejecutamos el código para que este nos muestre la ventana de que la conexión es correcta:



```
Consola de depuración de Mi x + - □ ×
Conectado a la base de datos exitosamente.
C:\Users\manri\source\repos\Actividad_2-UzielLopez\x64\Debug\Actividad_2-UzielLopez.exe (proceso 17292) se cerró con el código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración -> Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .|
```

## Tablas:

Para realizar las tablas en el programa de “C++” se continua con el código anterior, agregando lo que se necesita, en este caso manejadores para los “States”:

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

Después para las tablas crearemos unas líneas de código para acomodar la información de acuerdo a los espacios que se necesitan:

```
31      wcout << L"+-----+-----+ " << endl;
32      wcout << L"| No. | Nombre Empleado | " << endl;
33      wcout << L"+-----+-----+ " << endl;
```

Se realiza la consulta en la base de datos para lograr obtener la información requerida:

```
35      //Seleccionar la base de datos
36      ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT * FROM Empleados", SQL_NTS);
```



Se realiza un “if” para saber si la consulta que se hizo fue exitosa o no, si fue exitosa se declaran las variables:

```
38     if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
39         int num_empleado;
40         SQLWCHAR name[30];
```

El proceso “while” nos ayuda a codificar los procesos que nosotros queremos que se plasmen, manteniendo relación con lo que está en la base de datos que nosotros tenemos en “SQL”:

```
41     while (SQLFetch(hStmt) == SQL_SUCCESS) {
42         SQLGetData(hStmt, 1, SQL_C_LONG, &num_empleado, 0, NULL);
43         SQLGetData(hStmt, 2, SQL_C_WCHAR, name, sizeof(name), NULL);
44
45         wcout << "    " << num_empleado << "    " << name << endl;
```

Se imprime una línea de código para lograr cerrar la tabla con la información solicitada:

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

Se tiene que liberar los manejadores:

```
50     //Liberar el manejador de Conexión
51     SQLFreeHandle(SQL_HANDLE_STMT, hStmt);
52
```

Y con esto, si nosotros queremos imprimir la información que tenemos en nuestra base de datos solamente ejecutamos el código:

```
Consola de depuración de Mi...
Conectado a la base de datos exitosamente.
+-----+-----+
| No. | Nombre Empleado |
+-----+-----+
1    Alejandra
2    Norma
3    Jorge
4    Alexis
5    Mariana
6    Perla
+-----+-----+

C:\Users\manri\source\repos\Actividad_2_UzielLopez\x64\Debug\Actividad_2_UzielLopez.exe (proceso 26332) se cerró con el
código 0.
Para cerrar automáticamente la consola cuando se detiene la depuración, habilite Herramientas ->Opciones ->Depuración ->
Cerrar la consola automáticamente al detenerse la depuración.
Presione cualquier tecla para cerrar esta ventana. . .|
```

Si la comparamos con la base de datos que tenemos en “SQL Server” nos daremos cuenta de que la información coincide correctamente:

	Num_Empleado	Nombre
1	1	Alejandra
2	2	Norma
3	3	Jorge
4	4	Alexis
5	5	Mariana
6	6	Perla

### Código:

```

1  // Actividad_2_UzielLopez.conexión de la base de datos en SQL
2  //
3
4  #include <windows.h>
5  #include <sql.h>
6  #include <sqlext.h>
7  #include <iostream>
8
9  using namespace std;
10
11 int main() {
12     SQLHENV hEnv;
13     SQLHDBC hDbc;
14     SQLRETURN ret;
15
16     //Asignar un gestor de entorno
17     ret = SQLAllocHandle(SQL_HANDLE_ENV, SQL_NULL_HANDLE, &hEnv);
18     ret = SQLSetEnvAttr(hEnv, SQL_ATTR_ODBC_VERSION, (SQLPOINTER)SQL_OV_ODBC3, 0);
19
20     //Asignar un gestor de la conexión
21     ret = SQLAllocHandle(SQL_HANDLE_DBC, hEnv, &hDbc);
22
23     //Conectarse a la base de datos
24     ret = SQLConnect(hDbc, (SQLWCHAR*)L"sqlserver", SQL_NTS, (SQLWCHAR*)L"Username", SQL_NTS, (SQLWCHAR*)L>Password", SQL_NTS);
25
26     if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
27         cout << "Conectado a la base de datos exitosamente." << endl;

```

```

28
29     SQLHSTMT hStmt;
30     ret = SQLAllocHandle(SQL_HANDLE_STMT, hDbc, &hStmt);
31     wcout << L"+-----+ " << endl;
32     wcout << L"| No. | Nombre Empleado | " << endl;
33     wcout << L"+-----+ " << endl;
34
35     //Seleccionar la base de datos
36     ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT * FROM Empleados", SQL_NTS);
37
38     if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
39         int num_empleado;
40         SQLWCHAR name[30];
41         while (SQLFetch(hStmt) == SQL_SUCCESS) {
42             SQLGetData(hStmt, 1, SQL_C_LONG, &num_empleado, 0, NULL);
43             SQLGetData(hStmt, 2, SQL_C_WCHAR, name, sizeof(name), NULL);
44
45             wcout << " " << num_empleado << " " << name << endl;
46
47         }
48         wcout << L"+-----+ " << endl;
49     }
50     //Liberar el manejador de Conexión
51     SQLFreeHandle(SQL_HANDLE_STMT, hStmt);

```

```

50      //Liberar el manejador de Conexión
51      SQLFreeHandle(SQL_HANDLE_STMT, hStmt);
52
53  }
54  else {
55      cout << "Fallo la conexión de la base de datos." << endl;
56  }
57
58  //Desconectar y liberar manejadores
59  SQLDisconnect(hDbc);
60  SQLFreeHandle(SQL_HANDLE_DBC, hDbc);
61  SQLFreeHandle(SQL_HANDLE_ENV, hEnv);
62
63  return 0;
64  }

```

### Conclusión

En esta actividad se observó cómo vincular una base de datos de “SQL Server” a través de un controlador como lo es “ODBC” con unos pasos sencillos pero precisos para lograr la conexión, después de ello toco al programa “Visual Studio” para que ahora con el lenguaje de programación en “C++” realicemos nuestra conexión, el código generado nos apoyó para que nosotros ingresaremos los datos de acuerdo a la información que le asignamos a nuestra base en “SQL Server”, y que gracias a ello se puedan consultar cierta información que queramos que nos muestre lo que necesitemos, por el momento solo consultemos el número de empleado y el nombre de los mismos , sin embargo, no está de más también encerrar la información en tablas para mantener la información más coherente y precisa. En este documento logre darme cuenta de la potencia que maneja un lenguaje de programación básico, con el solo hecho de saber cómo funciona es suficiente para sacarle el máximo provecho.

### Link para GitHub:

<https://github.com/Levzu-Ing/Lenguaje-de-Programaci-n-II.git>

## Referencias

David-Engel. (2023, 23 mayo). *¿Qué es ODBC? - ODBC API Reference*. Microsoft Learn.

<https://learn.microsoft.com/es-es/sql/odbc/reference/what-is-odbc?view=sql-server-ver16>

insightsoftware. (2024, 21 octubre). *Qué es ODBC - Conectividad abierta de bases de datos*.

Insightsoftware. <https://insightsoftware.com/es/blog/what-is-odbc/>