

# Actividad | #3 | Ejecución de Datos en el Programa

## Lenguajes de Programación II

Ingeniería en Desarrollo de Software



TUTOR: Félix Acosta Hernández

ALUMNO: Erick Alfredo Quiroz Figueroa

FECHA: 07/12/2024

## 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>Código.....</b>         | <b>7</b>  |
| <b>Conclusión .....</b>    | <b>10</b> |
| <b>Bibliografía .....</b>  | <b>11</b> |

## **Introducción**

Como sabemos, en el mundo se está digitalizando la información y para ello nacen los lenguajes de programación y las bases de datos en las cuales se busca simplificar procesos y facilitar el trabajo de registros de objetos y personas dependiendo el tipo de base de datos que se refiera.

Para empezar, conocemos que existe un lenguaje de alto nivel llamado C++, que no es más que un lenguaje de programación, para la creación de aplicaciones o programas un poco complejos, por medio de su sintaxis que es amigable con el programador o usuario que trabaje con él.

También, por otro lado, existe un lenguaje de programación para la creación de bases de datos, tablas, relaciones, entre otras cosas; llamado SQL, que no es mas que un lenguaje de consulta que nos ayuda a insertar y crear las tablas necesarias para un sistema en específico.

Y juntando estos dos lenguajes y haciendo una conexión correcta por medio de protocolos específicos se puede llegar al resultado deseado.

### **Descripción**

En esta actividad tendrá la secuencia de las dos actividades anteriores.

En la primera actividad se creo la base de datos, creando a su vez las tablas necesarias que nos pedían, posteriormente, en la segunda actividad se crean las tablas en el lenguaje C++, haciendo el código necesario para que tenga una conexión limpia con SQL y tener a su vez un aditamento con el cual podemos hacer la conexión con dicha base de datos, buscando el servidor que utilizamos al momento de crear la base de datos.

Ahora, teniendo de referencia las dos actividades ya mencionadas buscaremos que se impriman los registros ya ingresados en la base de datos usando código en C++, y al momento de compilar se pueda reflejar en pantalla los datos ingresados en la misma.

Una vez ingresado el código y teniendo el resultado esperado en la pantalla al momento de compilar, compartiremos pantallazos o capturas de pantalla mostrando el trabajo realizado con éxito en la actividad en curso.

### **Justificación**

La finalidad de esto es conocer como es que se pueden imprimir los datos ingresados en una base de datos. Y como es que puedes crear de manera sencilla en la consola la tabla para que se vea decente la base de datos o las tablas de la misma base de datos.

Conocer dichos lenguajes de programación nos ayudara a crear sistemas de datos de manera eficaz y teniendo en cuenta dicho conocimiento crear oportunidades laborales ya que en el mundo en el que vivimos la demanda del área digital y las tecnologías lo piden de una manera rápida y en aumento.

Teniendo todo esto en cuenta, podemos decir que es sencillo y que solo depende de la creatividad y fluidez del programador para crear las bases de datos esperadas.

Lo importante de todo esto es que conociendo los lenguajes de programación es que podemos programar y dejando echar a volar nuestra lógica para la creación de dichos sistemas y que todo se vuelva mucho mas sencillo al momento de programar.

Dicho todo esto, damos inicio a la actividad.

## Desarrollo

### Ejecución

En esta captura de pantalla se puede apreciar el correcto funcionamiento del código en C++ y la conexión exitosa con SQL donde es que se creo la tabla que se muestra en la imagen.

Console de depuración de Mi X + -

Conectado a la base de datos exitosamente.

| No. | Nombre Completo        | Fecha      | RFC          | Centro_de_Trabajo | Puesto                    | Descripción del Puesto                            | Directivo |
|-----|------------------------|------------|--------------|-------------------|---------------------------|---|-----------|
| 1   | Juan Perez Gomez       | 1985-06-15 | JPEGOMYV06DD | Oficina Central   | Gerente de Ventas         | Responsable de la gestión del equipo de ventas    | Si        |
| 2   | Maria Lopez Sanchez    | 1990-04-22 | MLDSAMYV04DD | Sucursal Norte    | Analista de Datos         | Analiza y reporta datos de ventas y mercado       | No        |
| 3   | Pedro Ramirez Torres   | 1978-12-03 | PRATORVY12DD | Oficina Central   | Director Financiero       | Responsable de la planificación financiera y estr | Si        |
| 4   | Laura Martinez Jimenez | 1983-08-10 | LMAJIMYV08DD | Sucursal Sur      | Defa de Recursos Humanos  | Gestiona el departamento de recursos humanos      | No        |
| 5   | Carlos Herrera Ruiz    | 1995-02-10 | CHERRUYV02DD | Oficina Central   | Desarrollador de Software | Desarrollo y mantenimiento de aplicaciones intern | No        |

E:\Carrera\Programacion II\Archivos\Actividad\_3\x64\Debug\Actividad\_3.exe (proceso 12520) se cerró con el código 0 (0x0).  
 Presione cualquier tecla para cerrar esta ventana. . .

## Código

Como primer paso se declaran las librerías

```

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

```

Se declaran variables y los gestores de conexión y entorno

```

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);
}

```

Se hace la conexión con la base de datos

```

// 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"
    wcout << L" | No. | Nombre Completo | Fecha | RFC | Centro_de_Trabajo | Puesto | Descripcion del Puesto | Directivo |
    wcout << L"

```

Se selecciona la tabla y se declaran las variables

```
//Select a la base de datos
ret = SQLExecDirect(hStmt, (SQLWCHAR*)L"SELECT * FROM Empleado2", SQL_NTS);
if (ret == SQL_SUCCESS || ret == SQL_SUCCESS_WITH_INFO) {
    int num_empleado;
    SQLWCHAR name[50];
    SQLWCHAR last_name[50];
    SQLWCHAR slast_name[50];
    SQLWCHAR directivo[50];
    SQLWCHAR b_day[50];
    SQLWCHAR rfc[50];
    SQLWCHAR Centro_Trabajo[50];
    SQLWCHAR puesto[50];
    SQLWCHAR desc_puesto[50];
}
```

Se manda llamar un ciclo while

```
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);
    SQLGetData(hStmt, 3, SQL_C_WCHAR, last_name, sizeof(last_name), NULL);
    SQLGetData(hStmt, 4, SQL_C_WCHAR, slast_name, sizeof(slast_name), NULL);
    SQLGetData(hStmt, 5, SQL_C_WCHAR, b_day, sizeof(b_day), NULL);
    SQLGetData(hStmt, 6, SQL_C_WCHAR, rfc, sizeof(rfc), NULL);
    SQLGetData(hStmt, 7, SQL_C_WCHAR, Centro_Trabajo, sizeof(Centro_Trabajo), NULL);
    SQLGetData(hStmt, 8, SQL_C_WCHAR, puesto, sizeof(puesto), NULL);
    SQLGetData(hStmt, 9, SQL_C_WCHAR, desc_puesto, sizeof(desc_puesto), NULL);
    SQLGetData(hStmt, 10, SQL_C_WCHAR, directivo, sizeof(directivo), NULL);
}
```

Se concatenan los apellidos

```
//Concatenar apellido paterno + apellido materno

wstring last_names = wstring(last_name) + L" " + wstring(slast_name);
wstring full_name = wstring(name) + L" " + wstring(last_names);
wstring dir = wstring(directivo);
wstring resp;

if (dir == L"1")
{
    resp = L"Si";
}
else
{
    resp = L"No";
}
```



Se codifica para mandar imprimir los datos

```
//Impresion de datos de la fila
wcout << L"| " << setw(6) << left << num_empleado <<
L"| " << setw(22) << left << full_name <<
L"| " << setw(13) << left << b_day <<
L"| " << setw(12) << left << rfc <<
L"| " << setw(20) << left << Centro_Trabajo <<
L"| " << setw(25) << left << puesto <<
L"| " << setw(51) << left << desc_puesto <<
L"| " << setw(10) << left << resp << L"| " << endl;
```

Se liberan la conexión y los gestores de entorno

```
// 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**

En esta actividad se observo el comportamiento del código realizado en el lenguaje C++ teniendo una conexión exitosa con SQL y se mencionó o se describió parte por parte como es que el código funciona dándonos así un correcto funcionamiento de la base de datos.

Esto nos ayudara a comprender y analizar mas a detalle como es que el código trabaja que partes de la sintaxis son importantes conocer para no tener errores del mismo código al momento de depurar y compilar.

Aprendimos a realizar una base de datos limpia y sencilla en el lenguaje SQL ingresando los datos que nos fueron pidiendo, todo esto con el fin de poder realizar dicha actividad y desarrollar nuestras habilidades lógicas al momento de tirar código.

Dicho de esta manera podemos dar por concluida la actividad llevándonos un aprendizaje en la materia de programación. Conociendo así también los paradigmas de la programación Orientada a Objetos.

### **Bibliografia**

Link de GitHub: [https://github.com/Erick123Alf/Programacion\\_II/tree/main/Actividad\\_3](https://github.com/Erick123Alf/Programacion_II/tree/main/Actividad_3)