

# **Actividad 3 – Calculo de RFC**

## **Lenguajes de Programacion I**

### **Ingeniería en Desarrollo de Software**

**Tutor: Urbano Francisco Ortega Rivera**

**Alumno: Carolina Romero Seguame**

**Fecha: 30 de julio del 2023**



# INDICE

INTRODUCCION.....	2
DESCRIPCIÓN.....	3
JUSTIFICACION.....	4
DESARROLLO.....	5
• Codificacion	
• Prueba del sistema	
CONCLUSIÓN.....	8
REFERENCIAS.....	9



# INTRODUCCIÓN

El cálculo del RFC (Registro Federal de Contribuyentes) es un proceso importante en México para obtener un número de identificación fiscal único para personas físicas y morales. El RFC se utiliza para propósitos fiscales y administrativos, y se basa en ciertos datos personales como el nombre completo y la fecha de nacimiento.

El lenguaje C++ ofrece la capacidad de programar y realizar el cálculo del RFC de manera eficiente. Para calcular el RFC en C++, se deben seguir ciertas reglas y procedimientos específicos establecidos por las autoridades fiscales mexicanas.

Estas reglas abarcan pasos como la extracción y el formato de las iniciales del nombre y apellidos, la obtención de los dígitos de la fecha de nacimiento, y la aplicación de transformaciones y concatenaciones particulares para generar el RFC final.



## DESCRIPCIÓN

Se necesita crear un programa que permita calcular el RFC de los nuevos empleados de la constructora AMC. Este debe generarse a partir de la captura de nombre, apellido paterno, apellido materno y fecha de nacimiento.

Actividad:

Con base en las siguientes reglas (en esta aplicación se deberán de usar los conceptos de Objetos y Clases, vistos en la materia), realizar una aplicación que permita capturar el RFC de los usuarios. Esta debe realizarse con sus respectivos requerimientos en lenguaje C++:



## JUSTIFICACIÓN

Hay varias razones por las cuales puede ser útil realizar el cálculo del RFC en lenguaje C++. A continuación, se presentan algunas justificaciones importantes:

1. Automatización del proceso: El cálculo del RFC implica seguir una serie de reglas y procedimientos específicos.
2. Eficiencia y precisión: Al utilizar un programa en C++ para calcular el RFC, puedes aprovechar la capacidad de cálculo rápido y preciso del lenguaje.
3. Escalabilidad: Si necesitas calcular el RFC para múltiples personas o datos masivos, un programa en C++ puede adaptarse y manejar grandes cantidades de información de manera más eficiente que el cálculo manual.
4. Mantenimiento y actualización: Como mencionamos anteriormente, las reglas y requisitos para el cálculo del RFC pueden cambiar con el tiempo.
5. Reutilización del código: Una vez que tengas implementado un programa en C++ para el cálculo del RFC, puedes reutilizarlo en diferentes proyectos o aplicaciones que requieran esta funcionalidad.

En resumen, estos son aspectos clave para agilizar el proceso, reducir errores y optimizar el cumplimiento de las obligaciones fiscales y administrativas relacionadas con el RFC en México.



## DESARROLLO

- **Codificación**

```
#include <stdio.h>
#include <iostream>
#include <string>

class Empleado {

private:
    std::string nombre;
    std::string apellidopaterno;
    std::string apellidomaterno;
    std::string Fechanacimiento;

public:
    void capturarDatos() {
        std::cout << "Ingrese Nombre: ";
        std::getline(std::cin >> std::ws, nombre);

        std::cout << "Ingrese Apellido Paterno: ";
        std::getline(std::cin >> std::ws, apellidopaterno);

        std::cout << "Ingrese Apellido Materno: ";
        std::getline(std::cin >> std::ws, apellidomaterno);

        std::cout << "Ingrese Fecha de Nacimiento: (DD/MM/AAAA)";
        std::getline(std::cin >> std::ws, Fechanacimiento);
    }

    std::string generarRFC(){
        std::string rfc;

        rfc += (apellidopaterno[0]);
        for (char c: apellidopaterno) {
            if(c=='A' || c=='E' || c=='I' || c=='O' || c=='U')
            {
                rfc+= c;
                break;
            }
        }

        if (apellidomaterno.empty()){ rfc += 'X';
        } else {
            rfc += apellidomaterno[0];
```



```

    }
    rfc += nombre[0];

    rfc += Fechanacimiento.substr(8,2);
    rfc += Fechanacimiento.substr(3,2);
    rfc += Fechanacimiento.substr(0,2);
return rfc;
}
};

int main() {
    Empleado empleado;
    empleado.capturarDatos();
    std::string rfc = empleado.generarRFC();

    std::cout << "su RFC generado es: " << rfc <<
    std::endl;

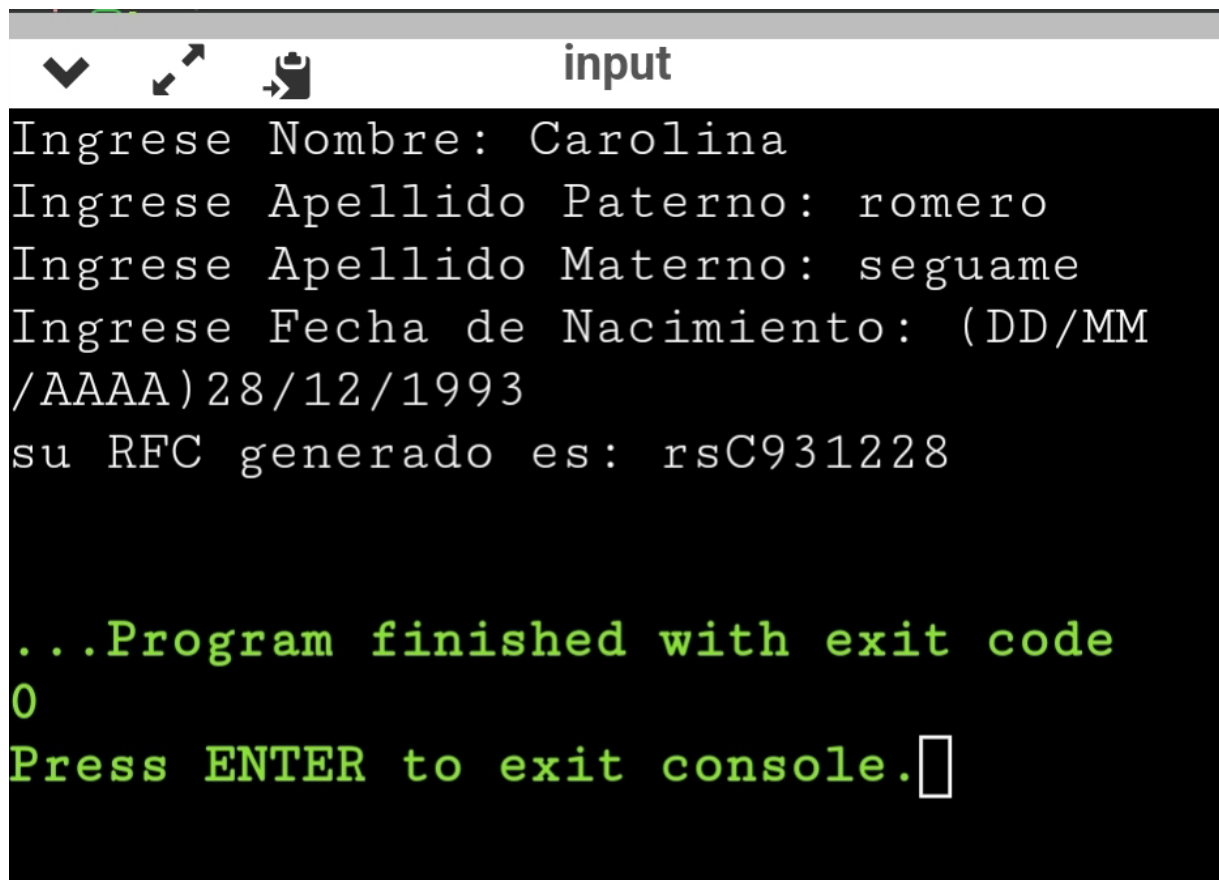
return 0;
}

```

Este es el código utilizado para el programa.



- Prueba del sistema



```
input
Ingrese Nombre: Carolina
Ingrese Apellido Paterno: romero
Ingrese Apellido Materno: seguame
Ingrese Fecha de Nacimiento: (DD/MM
/AAAA)28/12/1993
su RFC generado es: rsC931228

...Program finished with exit code
0
Press ENTER to exit console.
```

En esta captura se muestra el resultado del código con el RFC generado.



## CONCLUSIÓN

En conclusión, el lenguaje C++ es una poderosa herramienta para el desarrollo de aplicaciones y programas en múltiples áreas, incluyendo el cálculo del RFC en México. Su capacidad para realizar cálculos complejos de manera eficiente y su flexibilidad para adaptarse a cambios en las reglas y requisitos son ventajas importantes al implementar el cálculo del RFC en C++.

La automatización del proceso de cálculo, la mejora en la eficiencia y precisión, la escalabilidad, la facilidad de mantenimiento y actualización, y la reutilización del código son beneficios clave al utilizar C++ para el cálculo del RFC. Estas características ayudan a simplificar y agilizar el cumplimiento de las obligaciones fiscales y administrativas, así como a mantenerse al día con las regulaciones cambiantes relacionadas con el RFC.

En resumen, el uso de C++ para el cálculo del RFC proporciona una solución confiable y robusta, adaptada a las necesidades de cálculo del número de identificación fiscal en México.



## REFERENCIAS

Carolina Romero Seguame  
Capturas de pantalla

OnlineGDB  
[https://www.onlinegdb.com/online\\_c\\_compiler](https://www.onlinegdb.com/online_c_compiler)

GitHub  
<https://github.com/CarolinaRomeroSeguame12/LengProgr1/tree/main>

