

## Actividad 3

### Cálculo de RFC

#### lenguaje de programación I

Ingeniería en Desarrollo de  
Software



TUTOR: Francisco ortega rivera

ALUMNO: Brandon Garcia Martínez

FECHA: 10 de agosto 2024

## INDICE

•	Introducción	_____	3
•	Descripción	_____	3
•	Desarrollo	_____	4
•	Conclusión	_____	6
•	Referencias	_____	6

## **Introducción**

En este documento se demostrará el funcionamiento del lenguaje C++

En el ámbito de la programación en diferentes plataformas ya que con lleva una estructura y características de lenguaje muy desarrolladas para su objetivo, en fin.

Cuenta con gran reconocimiento en los videojuegos y en algunos navegadores web

También cuenta como lenguaje de programación de alto desempeño consideradas una de las mejores de aplicación de software rápido.

Este lenguaje fue diseñado por Bjarne Stroustrup con un objetivo o intención la cual era

Extender el desarrollo del lenguaje C para la manipulación de objetos esta intención y tecnología empezaría su evolución a mediados de los años 80.

El lenguaje C y C++ tienen sus parecidos la verdadera diferencia viene con la estructura interna ya que el lenguaje c solo podemos declarar variables al principio de un bloque en cambio el desarrollo de C++ tiene agregado segmentos que eliminan estas restricciones y muchas más.

## **Descripción**

En esta actividad desarrollaremos un programa que nos mostrara la complejidad sobre el lenguaje de programación en C++ sobre los datos en un entorno de datos exactos y formulados para el usuario

El ejercicio es bastante interesante, a decir verdad, pero para que se entienda lo mejor posible.

Empezando por las variables de las vocales para que puedan ser nuestras guías para que el usuario no ingrese malas interpretaciones

Haciendo si un trabajo más exacto para la entrega de dicha información

Utilizando símbolos de apoyo conocidos como arreglos &, [], etc.

Y serán integradas al programa con su respectivo orden y variable adecuada.

En este programa no utilizaremos int ya que es una variable de tipo numérica para solo números utilizaremos otras variables mas completas por ser este caso más especial como (char) y sentencias (for-if).

## Desarrollo

```

7 // iniciamos con la configuracion de vocales primordiales para no generar palabras malas
8
9
10 char ObtenerPrimeraVocalInterna(const string& str) {
11     for (size_t i = 1; i < str.length(); ++i) {
12         char c = str[i];
13         if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U') {
14             return c;
15         }
16     }
17     return 'X';
18 }

```

```

19 // configuracion de los caracteres a tomar en cuenta para generar el RFC
20
21 string CalcularRFC(const string& Nombre, const string& ApellidoPaterno, const string& ApellidoMaterno, const string& FechaNacimiento) {
22     string rfc;
23     char LetraInicial = ApellidoPaterno[0];
24     char PrimeraVocalInterna = ObtenerPrimeraVocalInterna(ApellidoPaterno);
25     // se configura el apellido materno por si el usuario no cuenta con el mismo
26     char InicialApellidoMaterno = (!ApellidoMaterno.empty()) ? ApellidoMaterno[0] : 'X';
27     char InicialNombre = Nombre[0];
28     // configuracion de los datos en posicion exacta a tomar de la fecha de nacimiento
29     string anio = FechaNacimiento.substr(2, 2);
30     string mes = FechaNacimiento.substr(5, 2);
31     string dia = FechaNacimiento.substr(8, 2);
32
33     rfc = LetraInicial;
34     rfc += PrimeraVocalInterna;
35     rfc += InicialApellidoMaterno;
36     rfc += InicialNombre;
37     rfc += anio;
38     rfc += mes;
39     rfc += dia;
40     return rfc;
41 }
42

```

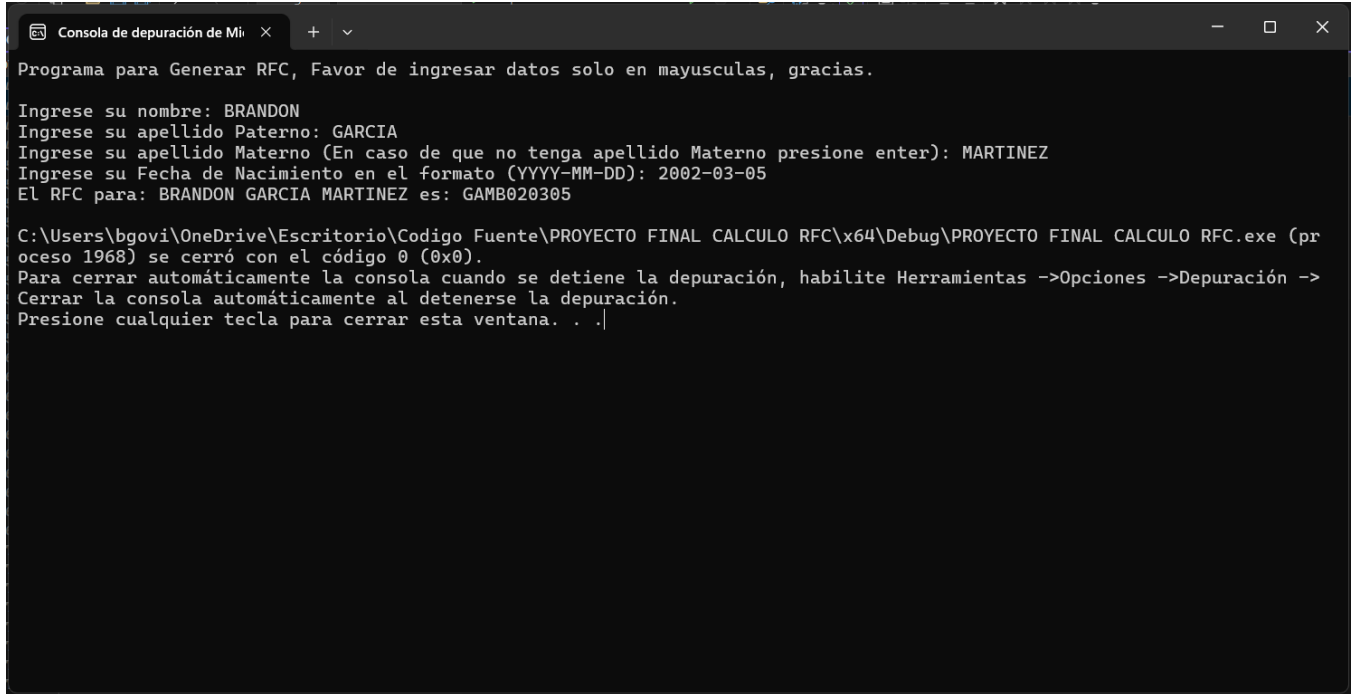
```
// configuración que ayuda a verificar las palabras inconvenientes con una base de datos
bool ValidarRFC(string rfc) {
    if (rfc.substr(0, 4) == "BUEI" || rfc.substr(0, 4) == "CACA" || rfc.substr(0, 4) == "CAGA" || rfc.substr(0, 4) == "CAKA" || rfc.substr(0, 4) == "COGE" || rfc.substr(0, 4) == "COJE")
        return false;
    return true;
}
```

```

01 // inicio con la interfaz del usuario para el ingreso de su informacion
02 int main() {
03     string Nombre, ApellidoPaterno, ApellidoMaterno, FechaNacimiento;
04
05     cout << "Programa para Generar RFC, Favor de ingresar datos solo en mayusculas, gracias." << endl << endl;
06
07     cout << "Ingrese su nombre: ";
08     getline(cin, Nombre);
09
10     cout << "Ingrese su apellido Paterno: ";
11     getline(cin, ApellidoPaterno);
12
13     cout << "Ingrese su apellido Materno (En caso de que no tenga apellido Materno presione enter): ";
14     getline(cin, ApellidoMaterno);
15
16     cout << "Ingrese su Fecha de Nacimiento en el formato (YYYY-MM-DD): ";
17     getline(cin, FechaNacimiento);
18
19     // se formula el rfc con los para metros que configuramos
20     string rfc = CalcularRFC(Nombre, ApellidoPaterno, ApellidoMaterno, FechaNacimiento);
21     if (!ValidarRFC(rfc)) {
22         rfc[3] = 'X';
23     }
24     // se entrega al usuario su rfc para que pueda confirmarlo
25     cout << "El RFC para: " << Nombre << " " << ApellidoPaterno << " " << ApellidoMaterno << " es: " << rfc << endl;
26
27     return 0;
28 }

```

## Prueba del sistema



```
Consola de depuración de Mi x + v
Programa para Generar RFC, Favor de ingresar datos solo en mayusculas, gracias.

Ingrese su nombre: BRANDON
Ingrese su apellido Paterno: GARCIA
Ingrese su apellido Materno (En caso de que no tenga apellido Materno presione enter): MARTINEZ
Ingrese su Fecha de Nacimiento en el formato (YYYY-MM-DD): 2002-03-05
El RFC para: BRANDON GARCIA MARTINEZ es: GAMB020305

C:\Users\bgovi\OneDrive\Escritorio\Codigo Fuente\PROYECTO FINAL CALCULO RFC\x64\Debug\PROYECTO FINAL CALCULO RFC.exe (proceso 1968) se cerró con el código 0 (0x0).
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. . .|
```

RFC original es: GAMB0203052S5 casi exacto.

## Conclusión

En este ejercicio pudimos entender cómo funcionan las nuevas librerías y arreglos. Mostrándonos los siguientes puntos:

La configuración de las vocales, una pequeña base de datos de groserías para una exactitud de datos, el uso correcto de los arreglos para determinar las posiciones y organización mas exacta de los datos dados por el usuario y la inclusión de el apellido materno como un dato que puede o no existir dependiendo de nuestro usuario.

Cabe recalcar que intente otra configuración mas exacta y disminuida en código, pero hay detalles para la simplificación de este.

De igual manera se trato de generar una interfaz entendible y formal para nuestro usuario.

Espero la pequeña base de datos no ofenda al lector y sea lo mas entendible posible ya que es un método un poco diferente a lo simplificado que puede ser.

Un dato para tomar en cuenta en este caso y tal vez alguno parecido es que al no tener apellido materno lo remplazara una x para rellenar el dato y poder brindar el RFC al usuario.

Dejo este link para mis evidencias de este y mis demás trabajos  
<https://github.com/BrandonGarcia0532/proyevto-final.git>

## Referencias

Brandon. (s.f.). *git hub*. Obtenido de <https://github.com/BrandonGarcia0532/proyevto-final.git>

Sierra, F. J. (s.f.). *Curso de programación c y c++*. Obtenido de [https://drive.google.com/drive/folders/15BOhHuaAQFeRSftbQkgaldv\\_jce\\_lix7?usp=drive\\_link](https://drive.google.com/drive/folders/15BOhHuaAQFeRSftbQkgaldv_jce_lix7?usp=drive_link)