

## **Actividad | 3 | Cálculo de RFC**

### **Lenguajes de Programación I**

Ingeniería en Desarrollo de Software

---



TUTOR: Urbano Francisco Ortega Rivera

ALUMNO: Adriana Esteban López

FECHA: 10 de agosto de 2024

# INDICE

Introducción	03
--------------	----

.....

Descripción	04
-------------	----

.....

Justificación	05
---------------	----

.....

Desarrollo	06
------------	----

.....

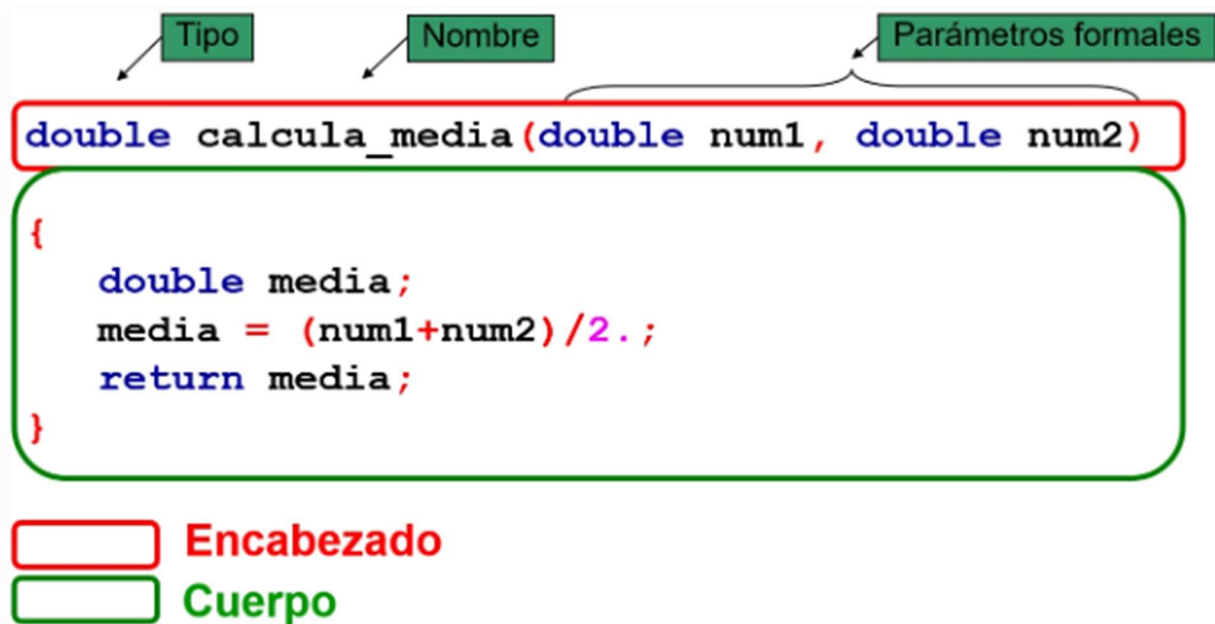
Conclusión	10
------------	----

.....

## INTRODUCCIÓN

Dentro de C++ tenemos el termino *funciones* que con bloques de código que van a realizar alguna operación en particular y devolver un valor como salida; las funciones son útiles para encapsular las operaciones comunes en un solo bloque *reutilizable* idealmente con un nombre que describa lo que hace la función.

En otras palabras, básicamente una función puede realizar las mismas acciones que un programa: aceptar datos, realizar cálculos determinados y devolver resultados y deben de contar con tres características: *definición, declaración y llamada*:



Ejemplo de una función.

## DESCRIPCIÓN

En el desarrollo de esta actividad se estará creando un programa permita calcular el RFC de los nuevos empleados de la constructora AMC; el cual debe de generarse a partir de la captura de nombre, apellido paterno, apellido materno y fecha de nacimiento del empleado, para lo cual nos estaremos apegando a las siguientes reglas:

1. Los primeros 2 caracteres se crean con la primera letra del apellido paterno y la primera vocal interna del apellido paterno.
2. La 3ra posición corresponde a la inicial del apellido materno. De no existir un apellido materno, se utiliza una X.
3. La 4ta posición corresponde a la inicial del primer nombre.
4. La 5ta y 6ta posición corresponde a los dos últimos dígitos del año de nacimiento.
5. La 7ma y 8va posición corresponde al mes de nacimiento.
6. La 9na y 10ma posición corresponde al día de nacimiento
7. En caso de que se forme una palabra inconveniente, la letra que corresponde a la inicial del 1er nombre se reemplaza con una X.
8. En esta actividad, el RFC que genere el programa será sin homoclave

<b>Ejemplo RFC:</b>	<b>V</b>	<b>E</b>	<b>C</b>	<b>J</b>	<b>8</b>	<b>8</b>	<b>0</b>	<b>3</b>	<b>2</b>	<b>6</b>
<b>Posición</b>	<b>1</b>	<b>2</b>	<b>3</b>	<b>4</b>	<b>5</b>	<b>6</b>	<b>7</b>	<b>8</b>	<b>9</b>	<b>10</b>

Nuevamente estaremos trabajando en el entorno de OnLineGDB.

## JUSTIFICACIÓN

El uso de funciones tiene sus beneficios:

1. Cada función tiene código relacionado, lo cual facilita que se comprenda el código relacionado cuando se llaman dentro del programa.
2. Las funciones facilitan la programación al eliminar la repetición del código.
3. Las funciones facilitan la reutilización del código, ya que se puede llamar a la función para realizar una tarea en diferentes secciones del programa o incluso fuera del programa.

Algo importante dentro de la programación es que el uso de las funciones permite dividir el programa en módulos más pequeños, lo que facilita la creación y mantenimiento de programas más grandes y complejos.

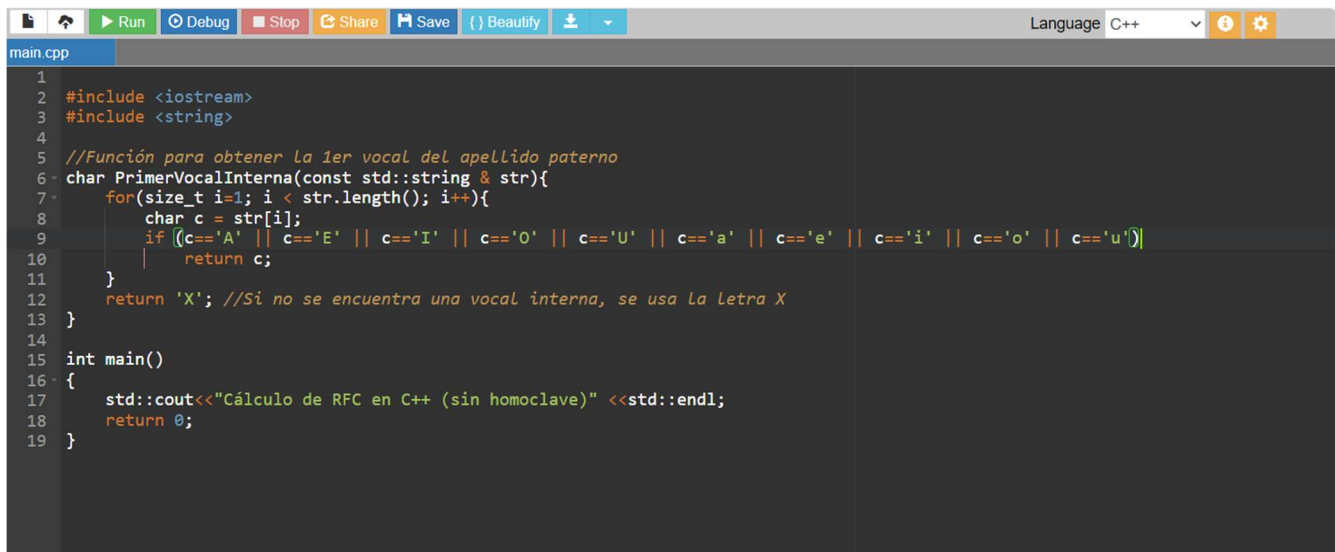
Dentro del desarrollo de la actividad estaremos haciendo uso de la función con la palabra clave ***const***, la cual especifica que el valor de una variable es constante e indica al compilador que evite que el programador lo modifique, es decir que es una función de solo lectura.

## DESARROLLO

Para el desarrollo de esta actividad, el usuario estará ingresando datos de tipo char (caracteres) por lo cual es necesario tener una librería que nos permita manipular este tipo de cadenas, lo cual se logra a través de la **librería string**,

Iniciamos el código de este programa creando funciones que nos permitan ir cumpliendo con cada una de las condicionantes que se requieren para formular el RFC:

**1. El 2do carácter corresponde a la 1ra vocal del apellido paterno, esta se considera la principal condicionante para este cálculo en particular.**



```
1
2 #include <iostream>
3 #include <string>
4
5 //Función para obtener la 1er vocal del apellido paterno
6 char PrimerVocalInterna(const std::string & str){
7     for(size_t i=1; i < str.length(); i++){
8         char c = str[i];
9         if ((c=='A' || c=='E' || c=='I' || c=='O' || c=='U' || c=='a' || c=='e' || c=='i' || c=='o' || c=='u'))
10            return c;
11     }
12     return 'X'; //Si no se encuentra una vocal interna, se usa la letra X
13 }
14
15 int main()
16 {
17     std::cout<<"Cálculo de RFC en C++ (sin homoclave)" <<std::endl;
18     return 0;
19 }
```

Iniciamos con un ciclo for esta función, este ciclo se va a estar repitiendo tantas veces sea necesario (de acuerdo a la longitud de la palabra) para comparar letra por letra e identificar si corresponde a una vocal.

Dentro de este ciclo for utilizamos la letra **i** (incremento y el que estará marcando el número de veces que se repite el ciclo de acuerdo a la longitud de la palabra), la instrucción `str.length` para obtener la longitud de la palabra y en caso de no encontrar una vocal, estará devolviendo una X en su lugar.

Vamos a declarar la variable **c** de tipo char en la cual estaremos almacenando la letra que se encuentre en determinada posición del apellido paterno y con la cual se estará realizando la comparación de acuerdo al ciclo if.

Pasamos ahora a realizar la función principal que estará generando el RFC de acuerdo a los datos ingresados por el usuario:

```
//Se obtiene la 1ra letra y la 1er vocal del apellido paterno
char LetraInicialP = ApellidoPaterno[0]; //Toma la letra que se encuentre en la 1ra posición del apellido ApellidoPaterno
char VocalInterna = PrimerVocalInterna (ApellidoPaterno);
```

Variable **LetraInicialP** alacena la letra que tenga el apellido paterno en la posición [0], recordemos que en un arreglo, la posición 1 esta enumerada con el número 0.

Variable **VocalInterna** almacena la 1er vocal que contenga el apellido paterno, y en este caso ese dato lo obtenemos con el uso de la función que declaramos inicialmente y que nombramos **PrimerVocalInterna**.

```
//Se obtiene la 1er letra del apellido materno y en caso de no haber apellido materno, estará asignando una X
char LetraInicialM = (!ApellidoMaterno.empty()) ? ApellidoMaterno[0]:'X';
```

Variable **LetraInicialM** almacena 1er letra del apellido materno, pero si no hay apellido materno tendrá que asignarle una X a la variable, y para ello utilizamos la función **empty** que devuelve un valor verdadero/falso; de tal manera que si devuelve un resultado falso es cuando le asigna la letra X a la variable LetraInicialM.

```
//Se obtienen Los digitos correspondientes a La fecha de nacimiento
std::string aa = FechaNacimiento.substr(2,2); //dos últimos dígitos del año del año de nacimiento
std::string mm = FechaNacimiento.substr(5,2); // dígitos correspondientes al mes de nacimiento
std::string dd = FechaNacimiento.substr(8,2); // dígitos correspondientes al día de nacimiento
```

Para obtener los datos correspondientes a la fecha de nacimiento (año, mes y día) estaremos utilizando la función **substr** que devuelve caracteres del valor cadena que comienzan desde la posición de caracteres que se especifican, en este caso las posiciones que colocamos entre corchetes.

```
//Con la información obtenida vamos a construir el RFC
RFC = LetraInicialP;
RFC += VocalInterna;
RFC += LetraInicialM;
RFC += LetraInicialNombre;
RFC += aa;
RFC += mm;
RFC += dd;
```

Ahora con la información que hemos obtenido a través de las diferentes funciones, vamos a construir el RFC, el cual estaremos almacenando en la variable **RFC**.

Una vez que terminamos con la declaración de funciones, pasamos a la parte final del programa en la cual declaramos las variables a utilizar, las cuales serán de tipo carácter (char) ya que todo el programa se basa en el uso de cadenas; así mismo solicitar al usuario que ingrese los datos que se requieren para el cálculo del RFC.

```
int main()
{
    //Declaración de las variables en las cuales estaremos almacenando los datos ingresados por el usuario
    std::string Nombre, ApellidoPaterno, ApellidoMaterno, FechaNacimiento;

    std::cout<<"Cálculo de RFC en C++ (sin homoclave)" <<std::endl;

    //Solicitamos al usuarios sus datos y los vamos almacenando en las variables

    std::cout << "Ingrese su Nombre: ";
    std::getline (std::cin,Nombre);

    std::cout << "Ingrese su Apellido Paterno: ";
    std::getline (std::cin,ApellidoPaterno);

    std::cout << "Ingrese su Apellido Materno, en caso de que no tenga, solo de enter: ";
    std::getline (std::cin,ApellidoMaterno);

    std::cout << "Ingrese su fecha de nacimiento (YYYY/MM/DD): ";
    std::getline (std::cin,FechaNacimiento);
```

```
//Aquí llamamos a la función CalcularRFC que declaramos en líneas arriba
std::string RFC = CalcularRFC (Nombre,ApellidoPaterno,ApellidoMaterno,FechaNacimiento);

std::cout << "El RFC es: " << RFC << std::endl; //finalmente hacemos la impresión del resultado final
```

Aquí hacemos uso de la función CalcularRFC que se declaró al inicio, para posteriormente hacer la impresión del RFC.

Vamos a ejecutar el programa para revisar los resultados:



```
main.cpp
29
30 //Se obtienen Los digitos correspondientes a la fecha de nacimiento
31 std::string aa = FechaNacimiento.substr(2,2); //dos últimos dígitos del año del año de nacimiento
32 std::string mm = FechaNacimiento.substr(5,2); // dígitos correspondientes al mes de nacimiento
33 std::string dd = FechaNacimiento.substr(8,2); // dígitos correspondientes al día de nacimiento
34
35 //Con La información obtenida vamos a construir el RFC
36 RFC = LetraInicialP;
37 RFC += VocalInterna;
38 RFC += LetraInicialM;
39 RFC += LetraInicialNombre;
40 RFC += aa;
41 RFC += mm;
42 RFC += dd;
43
44 return RFC;
45 }
46
47 int main()
48 {
49 //Declaración de Las variables en Las cuales estaremos almacenando Los datos ingresados por el usuario
50 std::string Nombre, ApellidoPaterno, ApellidoMaterno, FechaNacimiento;
51
52 std::cout<<"Cálculo de RFC en C++ (sin homoclave)" <<std::endl;
53
54 //Solicitamos al usuarios sus datos y Los vamos almacenando en Las variables
55
56 std::cout << "Ingrese su Nombre: ";
57 std::getline (std::cin,Nombre);
58
59 std::cout << "Ingrese su Apellido Paterno: ";
60 std::getline (std::cin,ApellidoPaterno);
61
62 std::cout << "Ingrese su Apellido Materno: ";
63 std::getline (std::cin,ApellidoMaterno);
64
65 std::cout << "Ingrese su fecha de nacimiento (YYYY/MM/DD): ";
66 FechaNacimiento = std::getline (std::cin,FechaNacimiento);
67
68 //Calculo de RFC
69 RFC = calcularRFC(Nombre,ApellidoPaterno,ApellidoMaterno,FechaNacimiento);
70
71 std::cout << "El RFC es: " << RFC << std::endl;
72
73 return 0;
74 }
```

input

```
Cálculo de RFC en C++ (sin homoclave)
Ingrese su Nombre: ADRIANA
Ingrese su Apellido Paterno: ESTEBAN
Ingrese su Apellido Materno, en caso de que no tenga, solo de enter: LOPEZ
Ingrese su fecha de nacimiento (YYYY/MM/DD): 1985/09/1985
El RFC es: EELA850919

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

```
main.cpp
29
30 //Se obtienen Los digitos correspondientes a la fecha de nacimiento
31 std::string aa = FechaNacimiento.substr(2,2); //dos últimos dígitos del año del año de nacimiento
32 std::string mm = FechaNacimiento.substr(5,2); // dígitos correspondientes al mes de nacimiento
33 std::string dd = FechaNacimiento.substr(8,2); // dígitos correspondientes al día de nacimiento
34
35 //Con La información obtenida vamos a construir el RFC
36 RFC = LetraInicialP;
37 RFC += VocalInterna;
38 RFC += LetraInicialM;
39 RFC += LetraInicialNombre;
40 RFC += aa;
41 RFC += mm;
42 RFC += dd;
43
44 return RFC;
45 }
46
47 int main()
48 {
49 //Declaración de Las variables en Las cuales estaremos almacenando Los datos ingresados por el usuario
50 std::string Nombre, ApellidoPaterno, ApellidoMaterno, FechaNacimiento;
51
52 std::cout<<"Cálculo de RFC en C++ (sin homoclave)" <<std::endl;
53
54 //Solicitamos al usuarios sus datos y Los vamos almacenando en Las variables
55
56 std::cout << "Ingrese su Nombre: ";
57 std::getline (std::cin,Nombre);
58
59 std::cout << "Ingrese su Apellido Paterno: ";
60 std::getline (std::cin,ApellidoPaterno);
61
62 std::cout << "Ingrese su Apellido Materno: ";
63 std::getline (std::cin,ApellidoMaterno);
64
65 std::cout << "Ingrese su fecha de nacimiento (YYYY/MM/DD): ";
66 FechaNacimiento = std::getline (std::cin,FechaNacimiento);
67
68 //Calculo de RFC
69 RFC = calcularRFC(Nombre,ApellidoPaterno,ApellidoMaterno,FechaNacimiento);
70
71 std::cout << "El RFC es: " << RFC << std::endl;
72
73 return 0;
74 }
```

input

```
Cálculo de RFC en C++ (sin homoclave)
Ingrese su Nombre: ERIK
Ingrese su Apellido Paterno: OROZCO
Ingrese su Apellido Materno, en caso de que no tenga, solo de enter:
Ingrese su fecha de nacimiento (YYYY/MM/DD): 1978/12/08
El RFC es: OOXE781208

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

## CONCLUSIÓN

Las funciones son herramientas poderosas que elevan nuestro código a nivel de eficiencia y claridad, al utilizarlas de manera efectiva no solo se mejora la estructura de nuestro programa, sino que también se facilita el trabajo colaborativo y el mantenimiento a largo plazo.

En el desarrollo de esta actividad se generan 2 funciones: PrimerVocalInterna, que como ya se mencionó en el Desarrollo, nos ayuda a obtener la primera vocal del apellido paterno que ingresa el usuario; así como también Calcular RFC, la cual hace o conforma ya el RFC con los datos que ingresa el usuario, y es dentro de esta función que llamamos a la función PrimerVocalInterna.

Se agrega dicha actividad a la plataforma de GitHub a través del siguiente link (archivo PDF):

<https://github.com/22HADRIA/Lenguajes-de-Programaci-n-I>

Visualización del código en GitHub

<https://fluffy-space-orbit-4jqw654g6qpg2jjxw.github.dev/>

Para consultar el código, se puede acceder al siguiente enlace:

[https://drive.google.com/file/d/1cVMPz62jivs1Tk6CjJyb4CJ5LzkeMGX8/view?usp=drive\\_link](https://drive.google.com/file/d/1cVMPz62jivs1Tk6CjJyb4CJ5LzkeMGX8/view?usp=drive_link)

El siguiente link es donde se capturo el código del programa:

[Actividad 2: Operaciones Básicas en C++ - GDB online Debugger | Code, Compile, Run, Debug online C, C++ \(onlinegdb.com\)](#)