

## Actividad | 3 | Cálculo de RFC

### Lenguajes de Programación I

#### Ingeniería en Desarrollo de Software

---



Tutor: Francisco Ortega Rivera

Alumno: José Ramón Pérez García

Fecha: 09/08/2025

<b>Índice .....</b>	<b>2</b>
<b>Introducción.....</b>	<b>3</b>
<b>Descripción .....</b>	<b>3</b>
<b>Justificación.....</b>	<b>3</b>
<b>Desarrollo .....</b>	<b>4</b>
Código.....	4
Prueba del sistema .....	6
<b>Referencias .....</b>	<b>7</b>

**Introducción**

En previas actividades conocimos el lenguaje y sus principios básicos, así como operaciones sencillas, en esta ocasión, jugamos un poco más con las clases y la creación de funciones, sin dejar de lado el uso de las variables y el manejo del tipo de datos.

**Descripción**

El reto era pedir al usuario la captura de su nombre, apellidos y fecha de nacimiento, y con eso hacer el cálculo de su RFC (sin homoclave) y dar el resultado al usuario.

Para llegar a eso, se consideró una lista de palabras que causarían una excepción en caso de que el RFC forme una de ellas, y se aplicó una corrección para evitar que eso pase. Así mismo, agregue el uniformar la captura de los datos convirtiendo todo a mayúsculas para que las validaciones no fallen.

**Justificación**

El código se divide en dos partes, la primera es la creación de las funciones necesarias, una para evitar palabras indebidas, otra para localizar la primer vocal en el apellido paterno o en su defecto sustituir su ausencia, y una tercera para el cálculo del rfc que es donde se realizan las extracciones de la información.

Al final, en la segunda parte del código, esta la función principal que es donde hay interacción con el usuario, solicitando captura de información, empleando la función previamente creada y otorgando un resultado.

## Desarrollo

[Link a código](#)

### Código

Se incluyen las bibliotecas necesarias para el código.

```
1 #include <iostream>
2 #include <vector>
3 #include <string>
4 #include <algorithm>
5 #include <cctype>
6
```

Se crea un diccionario con palabras que no se permitirán al momento de armar el RFC, y se crea una función para que cuando se cumpla con esta condición se sustituya la ultima letra por X

```
7 // Diccionario de palabras no permitidas:
8 const std::vector<std::string> palabras_prohibidas = {
9     "PENE", "POPO", "PUTO"
10 };
11
12 // Verificar y modificar palabras prohibidas del Diccionario
13 std::string corregirRFC(const std::string& rfc) {
14     for (const std::string& palabra : palabras_prohibidas){
15         if (rfc == palabra) {
16             return rfc.substr(0, 3) + "X"; //reemplaza última letra por X
17         }
18     }
19     return rfc;
20 }
21
```

Se obtiene la primera Vocal del primer apellido.

```
// Función para obtener la primera vocal interna de una cadena
char obtenerprimeravocal(const std::string& str){
    for (size_t i = 1; i < str.length(); ++i){
        char c = str[i];
        if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
            return c;
    }
    return 'X';
}
```

Se crea función para armar el RFC, primero se realiza la extracción de primera letra en nombre y apellidos, así como extraer año, mes y día de fecha de nacimiento y se aplica la función previamente creada para extraer la primer vocal del primer apellido. Con todo esto se generan 7 variables.

```

32 ~ std::string calcularRFC(const std::string& nombre, const std::string& apellidoPaterno,
33 ~     const std::string& apellidoMaterno, const std::string& fechanacimiento){
34
35     //se declara la variable en donde se guarda el RFC
36     std::string rfc;
37
38     //se obtiene la letra inicial y la primera vocal del apellido apellidoPaterno
39     char letraInicial = apellidoPaterno[0];
40     char primeraVocal = obtenerprimeravocal(apellidoPaterno);
41     //se obtiene la letra inicial y la primera vocal del apellido apellidoMaterno
42     char inicialApellidoMaterno = apellidoMaterno[0];
43     //se obtiene la inicial del primer nombre
44     char inicialNombre = nombre[0];
45     // se obtiene los dos ultimos digitos del año de fechanacimiento
46     std::string anio = fechanacimiento.substr(8,2);
47     // se obtiene mes y día de fechanacimiento
48     std::string mes = fechanacimiento.substr(3,2);
49     std::string dia = fechanacimiento.substr(0,2);
50

```

Finalmente en la misma función, se van agregando las variables en el orden debido para obtener como resultado final el RFC

```

51     //se contruye el RFC sumando el texto de la info
52
53     rfc = letraInicial;
54     rfc += primeraVocal;
55     rfc += inicialApellidoMaterno;
56     rfc += inicialNombre;
57
58     //comprar con palabras prohibidas
59     rfc = corregirRFC(rfc);
60
61     rfc += anio;
62     rfc += mes;
63     rfc += dia;
64     return rfc;
65 }
66

```

En la función principal, se solicita al usuario la captura de su nombre, apellido paterno, apellido materno y fecha de nacimiento.

Nombre y apellidos se convierten a mayúsculas tras guardarse como variable.

Por último, se aplica la función previamente creada para el cálculo del RFC y se manda el resultado al usuario.

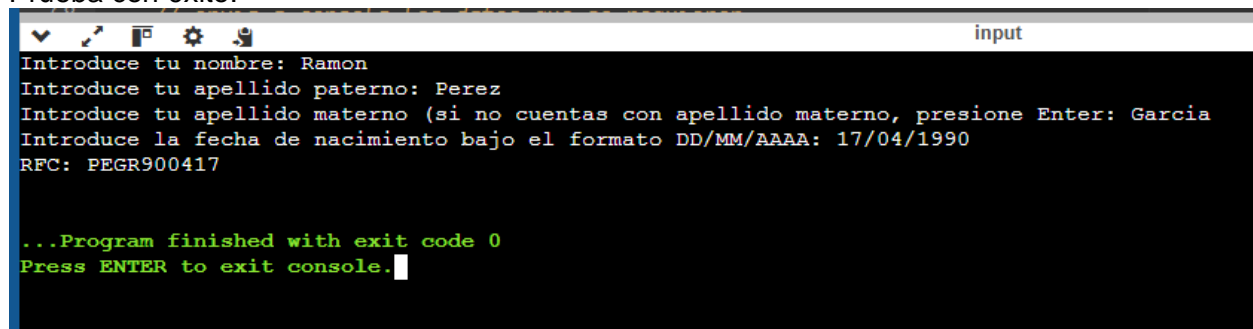
```

67- int main() {
68-     // envía a consola los datos que se requieren
69-     std::string nombre, apellidoPaterno, fechanacimiento;
70-     std::cout << "Introduce tu nombre: ";
71-     std::getline(std::cin, nombre);
72-     std::transform(nombre.begin(), nombre.end(), nombre.begin(), ::toupper);
73-     std::cout << "Introduce tu apellido paterno: ";
74-     std::getline(std::cin, apellidoPaterno);
75-     std::transform(apellidoPaterno.begin(), apellidoPaterno.end(), apellidoPaterno.begin(), ::toupper);
76-     std::cout << "Introduce tu apellido materno (si no cuentas con apellido materno, presione Enter: ";
77-     std::getline(std::cin, apellidoMaterno);
78-     std::transform(apellidoMaterno.begin(), apellidoMaterno.end(), apellidoMaterno.begin(), ::toupper);
79-     std::cout << "Introduce la fecha de nacimiento bajo el formato DD/MM/AAAA: ";
80-     std::getline(std::cin, fechanacimiento);
81-     std::string rfc = calcularRFC(nombre, apellidoPaterno, apellidoMaterno, fechanacimiento);
82-     std::cout << "RFC: " << rfc << std::endl;
83-
84-     return 0;
85- }

```

## Prueba del sistema

Prueba con éxito:



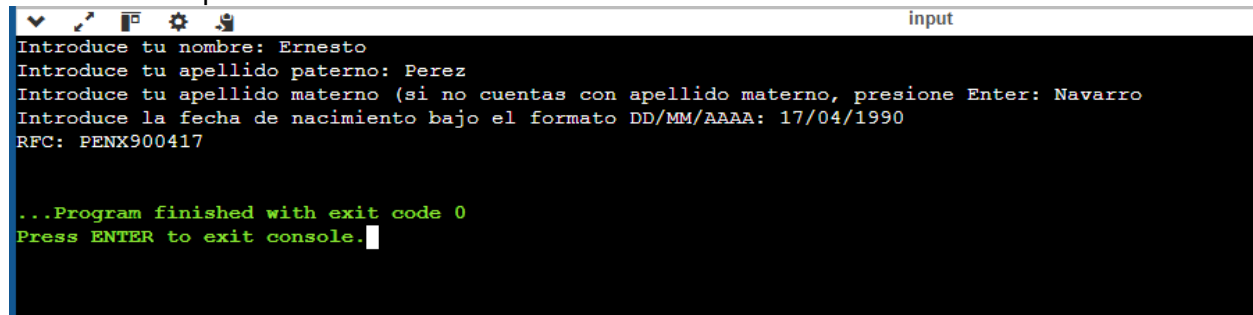
```

input
Introduce tu nombre: Ramon
Introduce tu apellido paterno: Perez
Introduce tu apellido materno (si no cuentas con apellido materno, presione Enter: Garcia
Introduce la fecha de nacimiento bajo el formato DD/MM/AAAA: 17/04/1990
RFC: PEGR900417

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

```

Caso con excepción:



```

input
Introduce tu nombre: Ernesto
Introduce tu apellido paterno: Perez
Introduce tu apellido materno (si no cuentas con apellido materno, presione Enter: Navarro
Introduce la fecha de nacimiento bajo el formato DD/MM/AAAA: 17/04/1990
RFC: PENX900417

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

```

## Conclusión

Tras la elaboración de este programa se refuerza el conocimiento básico sobre el lenguaje en C++, teniendo en esta primera instancia como conclusión y aprendizaje principal, la importancia de la sintaxis en el código, el uso de inteligente de las variables. Además, más allá de lo aprendido, surge la buena inquietud de las posibilidades y el interés de aprender a resolver interrogantes que surgen con este ejercicio que ayudaran a mejorar el código, retos tales como considerar los nombres comunes como José y María, considerar o blindar la captura de la fecha de nacimiento.

## Referencias