

Actividad | 3 |

Aplicación para cálculo de RFC

Lenguajes de Programación 1

Ingeniería en Desarrollo de Software



TUTOR: Lic. Francisco Ortega Rivera

ALUMNO: Joaquin Herman Valenzuela Aridjis

FECHA: 23/03/2024

Índice

Introducción.....	3
Descripción.....	3
Justificación.....	4
Desarrollo.....	5
Conclusión.....	8

Introducción

En el ámbito laboral actual, es muy importante la correcta gestión de la información de los empleados para el funcionamiento eficiente de las empresas. En este contexto, la presente actividad tiene como objetivo desarrollar una aplicación en C++ que permita calcular el RFC (sin homoclave) de los nuevos empleados de la constructora AMC.

El proceso de cálculo del RFC se basa en reglas específicas que combinan datos personales como el nombre, apellidos y fecha de nacimiento además de un orden específico para colocar cada uno de los caracteres. Mediante el uso código en C++, esta aplicación buscará automatizar y agilizar el proceso de generación del RFC, garantizando su precisión y cumplimiento con las normativas establecidas por el SAT.

A lo largo de este documento, se detallará el desarrollo y la implementación de la aplicación, así como su relevancia en el contexto laboral y su potencial impacto en la eficiencia administrativa de la constructora AMC.

Descripción

La actividad de desarrollo de una aplicación en C++ para el cálculo del RFC de los empleados de la constructora AMC implica la aplicación práctica de conceptos fundamentales de programación orientada a objetos. Al utilizar datos como nombres, apellidos y fecha de nacimiento, se aplicarán reglas específicas para generar de manera automatizada el RFC, un documento crucial en el ámbito fiscal y administrativo en México.

El diseño de la aplicación requerirá la creación de clases y objetos que permitan capturar los datos necesarios, realizar los cálculos correspondientes según las reglas del RFC y generar el resultado final de manera eficiente y precisa. Este proceso de desarrollo no solo busca cumplir con los requisitos técnicos de la actividad, sino también promover el aprendizaje práctico de la

programación en C++ y su aplicación en situaciones reales de manejo de información y procesamiento de datos.

Justificación

La creación de una aplicación en C++ para el cálculo del RFC de los empleados de la constructora AMC es fundamental por varias razones. En primer lugar, esta actividad permite aplicar los conocimientos teóricos adquiridos en programación orientada a objetos de manera práctica y con un propósito concreto, lo cual fortalece el aprendizaje y la comprensión de los conceptos.

Además, la automatización del proceso de generación del RFC mediante la aplicación contribuye significativamente a la eficiencia y precisión en la gestión de datos de los empleados. Al utilizar algoritmos y estructuras de datos en C++, se agiliza el cálculo del RFC y se reducen los posibles errores humanos, lo que resulta en un proceso más confiable y eficiente. Esta solución tecnológica no solo beneficia a la constructora AMC al optimizar sus procesos administrativos, sino que también destaca la importancia de la programación como herramienta clave en la mejora continua de las operaciones empresariales.

Desarrollo

Dentro de esta sección de la actividad iremos explicando parte por parte la funcionalidad del código, ¿por qué? De cada una de las líneas con la intención de que sea más fácil entender el funcionamiento.

```
#include <iostream>
#include <string>
#include <cctype>

using namespace std;
```

- `#include <iostream>`: Permite el uso de las funciones de entrada y salida estándar como `cin` y `cout`.
- `#include <string>`: Proporciona la funcionalidad para trabajar con cadenas de texto (`string`).
- `#include <cctype>`: Incluye funciones para manipular caracteres, como `toupper` que convierte caracteres en minúsculas a mayúsculas.
- La línea “`using namespace std;`” en C++ es una directiva que permite utilizar todas las funciones, clases y objetos definidos en el espacio de nombres “`std`” sin tener que especificar explícitamente este espacio de nombres cada vez que se hace referencia a ellos.

```

7 char obtenerPrimeraVocalInterna(const string& str) {
8     for (size_t i = 1; i < str.length(); i++) {
9         char c = toupper(str[i]); // Convertir a mayúsculas para simplificar la comparación
10        if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
11            return c;
12    }
13    return 'X';
14 }

```

- “char obtenerPrimeraVocalInterna(const string& str)”: Define una función que recibe una cadena de texto (string) como parámetro y devuelve el primer carácter vocal interno de la cadena o 'X' si no se encuentra ninguna vocal.
- “for (size_t i = 1; i < str.length(); i++)”: Pasa sobre cada carácter de la cadena, empezando desde la posición 1 para omitir la primera letra.
- “char c = toupper(str[i]);”: Convierte el carácter a mayúsculas para simplificar la comparación.
- “if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')”: Verifica si el carácter es una vocal.
- “return c;”: Imprime la primera vocal interna encontrada.
- “return 'X';”: Imprime 'X' si no se encontró ninguna vocal interna en la cadena.

```
string calcularRFC(const string& nombre, const string& apellidoPaterno, const string& apellidoMaterno, const string& fechaNacimiento) {
    string rfc;

    char letraInicial = toupper(apellidoPaterno[0]); // Convertir a mayúsculas para el RFC
    char primeraVocalInterna = obtenerPrimeraVocalInterna(apellidoPaterno);

    char inicialApellidoMaterno = (!apellidoMaterno.empty()) ? toupper(apellidoMaterno[0]) : 'X'; // Convertir a mayúsculas para el RFC

    char inicialNombre = toupper(nombre[0]); // Convertir a mayúsculas para el RFC

    string anio = fechaNacimiento.substr(2, 2);
    string mes = fechaNacimiento.substr(5, 2);
    string dia = fechaNacimiento.substr(8, 2);
```

- Define una función que calcula el RFC a partir de los datos de nombre, apellidos y fecha de nacimiento.
- Primero obtiene la primera letra del apellido paterno en mayúsculas.
- busca la primera vocal interna del apellido paterno.
- Luego obtiene la inicial del apellido materno en mayúsculas si existe, de lo contrario, asigna 'X'.
- Después obtiene la inicial del nombre en mayúsculas.
- Identifica los dos últimos dígitos del año de nacimiento.
- Obtiene el mes de nacimiento.
- Finalmente obtiene el día de nacimiento.

```
    rfc = letraInicial;
    rfc += primeraVocalInterna;
    rfc += inicialApellidoMaterno;
    rfc += inicialNombre;
    rfc += anio;
    rfc += mes;
    rfc += dia;

    return rfc;
```

- En esta parte de código arma internamente el RFC en el orden indicado, toma el valor que cada uno de esos parámetros tiene guardado, y lo acomoda de manera consecutiva

- `return rfc;`: imprime el RFC generado.

```
int main() {
    string nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento;

    cout << "Ingresa tu nombre(s): ";
    getline(cin, nombre);

    cout << "Ingresa tu apellido paterno: ";
    getline(cin, apellidoPaterno);

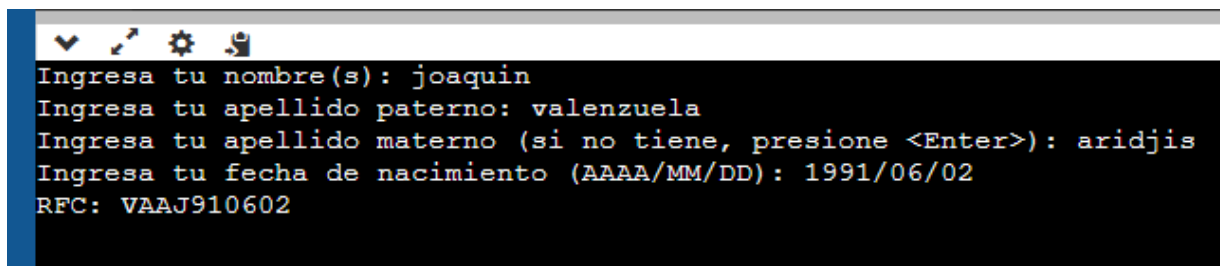
    cout << "Ingresa tu apellido materno (si no tiene, presione <Enter>): ";
    getline(cin, apellidoMaterno);

    cout << "Ingresa tu fecha de nacimiento (AAAA/MM/DD): ";
    getline(cin, fechaNacimiento);

    string rfc = calcularRFC(nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento);
    cout << "RFC: " << rfc << endl;

    return 0;
}
```

- en esta ultima parte del codigo, generamos las peticiones que hara la terminal, de donde el codigo obtendra la informacion para despues generar el RFC, asignando a cada variable el valor del cual despues calculara cada una de las pocisiones.



```
Ingresa tu nombre(s): joaquin
Ingresa tu apellido paterno: valenzuela
Ingresa tu apellido materno (si no tiene, presione <Enter>): aridjis
Ingresa tu fecha de nacimiento (AAAA/MM/DD): 1991/06/02
RFC: VAAJ910602
```

Por ultimo vemos el código trabajando correctamente, en este ejemplo observamos como a pesar de que la información la ingresamos en minúsculas, el código sabe reconocer cada carácter y el resultado lo imprime en mayúsculas, antes de incluir esta opción si ingresabas la información en minúsculas, al no reconocer las vocales debido al formato que le dimos, automáticamente el segundo valor lo generaba como "X", y para evitar errores y confusiones en los usuarios, decidí

incluir esta nueva regla.

Conclusión

En conclusión, la creación de la aplicación en C++ para calcular el RFC de los empleados de la constructora AMC ha sido un ejercicio enriquecedor y significativo en el ámbito académico y profesional. Esta actividad nos ha permitido aplicar de manera práctica los conceptos de programación orientada a objetos, fortaleciendo nuestra comprensión de la sintaxis del lenguaje y el diseño de clases y objetos.

La implementación de la aplicación no solo ha representado un desafío técnico, sino también una oportunidad para comprender la importancia de la automatización en la gestión de datos y la optimización de procesos en entornos empresariales. La eficiencia y precisión logradas al calcular el RFC demuestran el potencial de la tecnología para mejorar la operatividad y la toma de decisiones en el ámbito laboral.

En este sentido, la experiencia adquirida en esta actividad es valiosa para nuestro desarrollo profesional, al resaltar la relevancia de la programación como herramienta clave en la resolución de problemas y la innovación en diversas áreas de trabajo.

Referencias

<https://github.com/Aridjis/Lenguajes-de-programaci-n-1.git>