

Proyecto # III | Cálculo de RFC

Lenguajes de programación I



TUTOR: FRANCISCO ORTEGA RIVERA

ALUMNO: JOSE JOEL LANDEROS SANTOS

FECHA: 20 DE ABRIL DE 2025

Introducción

En este proyecto se trabajara en un ejercicio usando el lenguaje C++ que a mi parecer es el mejor lenguaje de bajo nivel por excelencia en cuestión de lógica por lo que es muy usado en competencias matemáticas ya que uno de sus pros son los segundos en los que puede correr un código y convertirlo en binario algo que a su competencia directa se le complica como lo es Java que usa un compilador intermediario instalable para poder correr lo escrito en ese lenguaje algunos datos encontrados es que el backend de Facebook y de Google maps fue escrito en C++ y también da inicio a aprender del framework de .Net que en general usa C# que es uno de los backends más utilizados en la industria actualmente, este ejercicio no es para nada un desafío pero es más que nada para ver cómo funciona la sintaxis del código.

Descripción

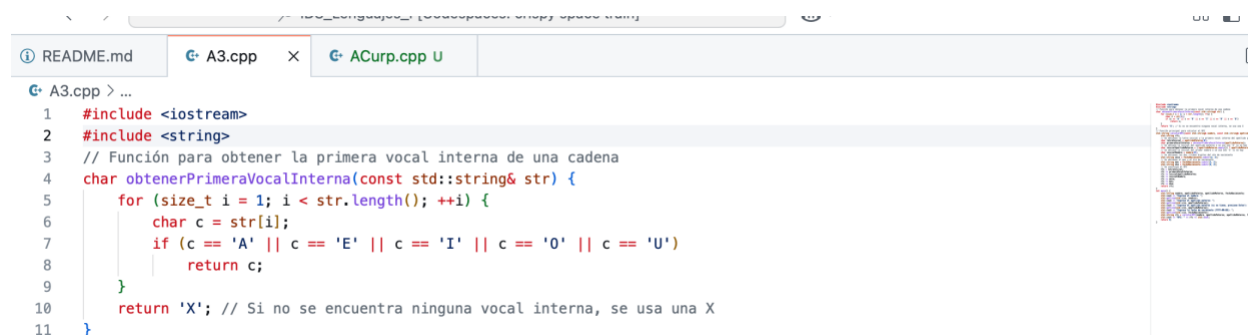
Las diferencias que he notado al usar este lenguaje con otros lenguajes a los que he usado es su sintaxis en el import que se maneja con un `#include <>` lo cual agrega librerías de esa manera, en este ejercicio se usó `iostream` que `i` es de in que significa entrada y `o` de out de salida y `stream` es flujo o mecanismo que podríamos interpretarlo como el flujo de las entradas y salidas que es muy usado para ver y agregar cosas en la consola como si la consola fuera una interfaz que es muy usado para practicar estos ejercicios de lógica de programación que en este caso usa una sintaxis algo compleja a mi parecer por lo que yo me propondría a mejor crear un método que reciba la información y otra que la imprima ya que sería una manera mas sencilla pero no se hará por que por el momento aquí se tiene que aprender a como se tiene que hacer con un paradigma funcional ya que si no la materia se llamaría orientado a objetos.

Justificación

Aun en esta 3ra actividad me siento cómodo utilizando el Visual Studio Code ya que lo copia de manera virtual sin necesidad de instalación, lo que hice fue escribir en `github.com` donde estaba guardado mi proyecto le cambie el `.com` por `.dev` y ahí instale el paquete de C/C++ y de allí ya solo me aparecio el boton de depurar y ejecutar ademas de eso desde ese lugar pude encontrar la manera de usar el comando “git add .” luego el “git commit “agregar ejercicio 2” para despues solo escribir un “git push “ y que se haga el cambio directamente en mi repositorio en lo personal he avanzado bastante desde mi primer ejercicio.

Desarrollo

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.



```

1  #include <iostream>
2  #include <string>
3  // Función para obtener la primera vocal interna de una cadena
4  char obtenerPrimeraVocalInterna(const std::string& str) {
5      for (size_t i = 1; i < str.length(); ++i) {
6          char c = str[i];
7          if (c == 'A' || c == 'E' || c == 'I' || c == 'O' || c == 'U')
8              return c;
9      }
10     return 'X'; // Si no se encuentra ninguna vocal interna, se usa una X
11 }
  
```

```

12 // Función principal para calcular el RFC
13 std::string calcularRFC(const std::string& nombre, const std::string& apellidoPaterno, const std::string& apellidoMaterno, const std::string& fechaNacimiento) {
14     std::string rfc;
15     // Se obtienen la letra inicial y la primera vocal interna del apellido paterno
16     char letraInicial = apellidoPaterno[0];
17     char primeraVocalInterna = obtenerPrimeraVocalInterna(apellidoPaterno);
18     // Se obtiene la inicial del apellido materno o se usa una 'X' si no hay
19     char inicialApellidoMaterno = (!apellidoMaterno.empty()) ? apellidoMaterno[0] : 'X';
20     // Se obtiene la inicial del primer nombre o se usa una 'X' si no hay
21     char inicialNombre = nombre[0];
22     // Se obtienen los dos últimos dígitos del año de nacimiento
23     std::string anio = fechaNacimiento.substr(2, 2);
24     // Se obtienen el mes y el día de nacimiento
25     std::string mes = fechaNacimiento.substr(5, 2);
26     std::string dia = fechaNacimiento.substr(8, 2);
27     // Se construye el RFC
28     rfc = letraInicial;
29     rfc += primeraVocalInterna;
30     rfc += inicialApellidoMaterno;
31     rfc += inicialNombre;
32     rfc += anio;
33     rfc += mes;
34     rfc += dia;
35     return rfc;
36 }

```

Imagen 1.1 Captura de pantalla en Visual Studio Code donde nos muestra el metodo que resive los strings de nuestros inputs y con la clase string tenemos los metodos substring ahí esta la clave.

```

36 }
37 int main() {
38     std::string nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento;
39     std::cout << "Ingrese el nombre: ";
40     std::getline(std::cin, nombre);
41     std::cout << "Ingrese el apellido paterno: ";
42     std::getline(std::cin, apellidoPaterno);
43     std::cout << "Ingrese el apellido materno (si no tiene, presione Enter): ";
44     std::getline(std::cin, apellidoMaterno);
45     std::cout << "Ingrese la fecha de nacimiento (YYYY-MM-DD): ";
46     std::getline(std::cin, fechaNacimiento);
47     std::string rfc = calcularRFC(nombre, apellidoPaterno, apellidoMaterno, fechaNacimiento);
48     std::cout << "RFC: " << rfc << std::endl;
49     return 0;
50 }

```

Imagen 1.2 Captura de pantalla en Visual Studio Code donde se puede ver la logica del codigo creado donde la clave para que el codigo sea limpio es en utilizar el metodo calcularRFC().



Imagen 1.3 Captura de pantalla en Visual Studio Code donde nos muestra el código corriendo sin errores de compilación, ejercicio se muestra sin homoclave.

Conclusión

Este proyecto no tuvo dificultades, es muy interesante C++ y creo que puede ser uno de mis lenguajes favoritos para programar lógica, y tener un panorama más amplio de los lenguajes de programación en lo personal hasta el momento no lo siento tan complicado es orientado a objetos utiliza métodos y creo yo que es uno de los lenguajes que hemos estado infravalorando más en la industria sin embargo verlo en esta área de aprendizaje de hace muy alegre ya que no había tenido la oportunidad de encontrarme con el frente a frente y me hizo muy interesante para volver a hacer problemas sencillos pero en lenguajes que no manejo, en cuestión de sacar dudas internet ya resolvió varias dudas desde hace muchos años y es muy difícil que te tropieces con algo que alguien mas no se haya topado con eso antes por lo que se puede aprender mucho de este lenguaje.

Referencias

1. Horton, I. (2018).
Beginning C++17: From novice to professional (5ª ed.). Apress.
2. Lafore, R. (2002).
Object-oriented programming in C++ (3ª ed.). Sams Publishing.

GitHub

https://github.com/josejoell/IDS_Lenguajes_I.git