

Actividad 3 - Cálculo de RFC.

Lenguaje de Programación I.

Ingeniería en Desarrollo de Software

Tutor: Urbano Francisco Ortega Rivera

Alumno: Ramón Ernesto Valdez Felix

Fecha: 22/07/23

Índice

Introducción	3
Descripción	3
Justificación	4
Desarrollo	4
Codificación	5
Prueba del Sistema	8
Conclusión	11
Referencias.....	11

Introducción

En esta actividad trabajaremos con el lenguaje de programación C++ y aprenderemos como crear un programa en el cual mostrara en pantalla como resultado el RFC de una persona, esto siendo la actividad principal que se solicita en el material de la tercera clase de lenguaje de programación 1 y en el documento de la actividad 3 que estamos cursando. En el programa que se pide realizar se ingresa cierta información sensible como el nombre, apellidos y fecha de nacimiento de una persona para que sea generado el RFC, toda esta información será utilizada para obtener el resultado deseado ya que el RFC se compone por las dos primeras letras del apellido paterno, la primera letra del apellido materno, la primera letra del nombre y la fecha de nacimiento con el formato aammdd, donde toda la unión de estos datos forma el RFC de una persona.

Descripción

En esta actividad trabajaremos con el lenguaje de programación C++, en el cual realizaremos un script o programa que nos permitirá ser calificado por el docente de la materia de lenguaje de programación 1 está siendo la tercera actividad de la materia y aprenderemos como crear un programa en el cual mostrará en pantalla como resultado el RFC de una persona. En el programa que realizaremos se requiere seguir una serie de requisitos para poder generar un RFC en los cuales se nos pide solicitar los siguientes datos o información sensible con la que tenemos que trabajar como el nombre, apellidos y fecha de nacimiento, toda esta información será utilizada para obtener el resultado deseado ya que el RFC se compone por las dos primeras letras del apellido paterno, la primera letra del apellido materno, la primera letra del nombre y la fecha de nacimiento con el formato aammdd, donde toda la unión de estos datos forma el RFC de una persona.

Justificación

En esta actividad se trabajará con el lenguaje de programación C++, en el cual se creará una script o programa en el cual nos basaremos en todos los puntos que tenemos que cumplir en la actividad 3 de la materia de lenguaje de programación 1:

- Como mínimo 600 palabras en dicho documento, representado en los siguientes: “Introducción, descripción, justificación y conclusión”.
- Crear un script que al introducir el nombre, apellidos y fecha de nacimiento. Generando como resultado en pantalla el RFC de la información de la persona ingresada.
- Guardar el script y comprimir en extensión zip para subirlo al GitHub compartiendo el link para que pueda consultar el maestro el script realizado.
- Agregar como evidencia en el punto siguiente con el programa realizado, código, pantallas de la ejecución de resultado en el sitio web proporcionado por la actividad y un segundo programa que puedas ser utilizado como alguno de los presentados en el foro de la materia del lenguaje de programación 1.

Desarrollo:

En esta actividad mostraremos el código y la evidencia que se recabara de la actividad 3 donde utilizaremos el lenguaje de programación de C++ para obtener el resultado del RFC de la persona ingresada.

Link: [GitHub](#)

Codificación

Código Actividad 3: Operaciones básicas en C++	Descripción Actividad 3: Operaciones básicas en C++
<pre>//Actividad 3. Cálculo de RFC //Actividad 3. Aplicación para Cálculo de RFC #include <iostream> #include <string> #include <algorithm> std::string calcRFC(const std::string& nom, const std::string& apellidoPa, const std::string& apellidoMa, const std::string& fechaNacimiento) { // Convertir los apellidos, nombre a mayúsculas y eliminan espacios. std::string apellidoPaMayus = apellidoPa; std::string apellidoMaMayus = apellidoMa; std::string nomMayus = nom;</pre>	<p>En este programa realizaremos la conjunción de los dígitos y así poder obtener el resultado requerido por la actividad de RFC.</p> <ul style="list-style-type: none"> • Utilizamos las librerías o bibliotecas estándar de C++, iostream que nos sirve para entradas y salidas, string que facilita la comparación de cadenas y algorithm donde esta define una colección de funciones especialmente diseñadas para usarse en rangos de elementos. • Declaramos cadena con los campos a utilizar (nombre= nom, apellido paterno= apellidoPa, apellido materno= apellidoMa y Fecha de Nacimiento= fechanacimineto).

<pre> std::transform(apellidoPaMayus.begin(), apellidoPaMayus.end(), apellidoPaMayus.begin(), ::toupper); std::transform(apellidoMaMayus.begin(), apellidoMaMayus.end(), apellidoMaMayus.begin(), ::toupper); std::transform(nomMayus.begin(), nomMayus.end(), nomMayus.begin(), ::toupper); // Tomar las primeras 2 letras del apellido paterno std::string primerasLetrasApellidoPa = apellidoPaMayus.substr(0, 2); // Tomar la primera letra del apellido materno, se asigna "X" si no hay apellido materno. std::string primeraLetraApellidoMa; if (apellidoMaMayus.empty()) { primeraLetraApellidoMa = "X"; } else { primeraLetraApellidoMa = apellidoMaMayus.substr(0, 1); } </pre>	<ul style="list-style-type: none"> • Realizamos la conversión a mayúsculas y la eliminación de espacio en los elementos a ingresar. • Recolectamos los dígitos que ocuparemos para formar el RFC, donde tomamos los 2 primeras letras del apellido paterno, la primera letra del apellido materno y en caso de tener asignar una letra X, primera letra del nombre y de la fecha de nacimiento usamos el siguiente formato aammdd que significa que tomaremos los dos últimos dígitos del año, los dos del mes y por último los dos últimos de día. • Se combinan todos los campos para generar el RFC. <p>Se realizó la función principal que nos servirá para ingresar el nombre, los apellidos y la fecha de nacimiento pasa así calcular el RFC y poder mandarlo mostrar</p>
--	---

<pre> // Tomar las primera letras del nombre std::string primeraLetrasNom = nomMayus.substr(0, 1); // Tomar los últimos dígitos del año de nacimiento std::string annusNac = fechaNacimiento.substr(6, 7); // Tomar el mes de nacimiento std::string mesNac = fechaNacimiento.substr(2, 3); // Tomar el día de nacimiento std::string diaNac = fechaNacimiento.substr(1, 1); // Se combina todo para formar el RFC std::string rfc = primerasLetrasApellidoPa + primeraLetraApellidoMa + primeraLetrasNom + annusNac + mesNac + diaNac; return rfc; } int main() { std::string nom; std::string apellidoPa; std::string apellidoMa; </pre>	<p>en pantalla como el resultado en el código realizado de calcular RFC.</p>
--	--

<pre> std::string fechaNacimiento; std::cout << "Ingrese el Nombre: "; std::getline(std::cin, nom); std::cout << "Ingrese el Apellido Paterno: "; std::getline(std::cin, apellidoPa); std::cout << "Ingrese el Apellido Materno o (dejar vacío si no tiene): "; std::getline(std::cin, apellidoMa); std::cout << "Ingrese la Fecha de Nacimiento (formato ddmmaaaa): "; std::getline(std::cin, fechaNacimiento); std::string rfc = calcRFC(nom, apellidoPa, apellidoMa, fechaNacimiento); std::cout << "RFC calculado: " << rfc << std::endl; return 0; } </pre>	
--	--

Prueba de Sistema

Evidencia 1: Pantalla https://www.onlinegdb.com/online_c_compiler: En este punto se anexa 2 pantalla que ejecuta el programa de c++ en el sitio web solicitado por la actividad 3, La primera pantalla muestra el ingreso de todos los campos y el resalta del RFC. En la

segunda evidencia se llenan los campos excepto el campo de apellido materno el cual asigna una letra X al no ser llenado dando el resultado esperado del RFC.

```

main.cpp
3  #include <iostream>
4  #include <string>
5  #include <algorithm>
6
7  std::string calcRFC(const std::string& nom, const std::string& apellidoPa, const std::string& ape
8      // Convertir los apellidos, nombre a mayúsculas y eliminan espacios.
9      std::string apellidoPaMayus = apellidoPa;
10     std::string apellidoMaMayus = apellidoMa;
11     std::string nomMayus = nom;
12     std::transform(apellidoPaMayus.begin(), apellidoPaMayus.end(), apellidoPaMayus.begin(), ::toupper);
13     std::transform(apellidoMaMayus.begin(), apellidoMaMayus.end(), apellidoMaMayus.begin(), ::toupper);
14     std::transform(nomMayus.begin(), nomMayus.end(), nomMayus.begin(), ::toupper);
15
input
Ingrese el Nombre: Ramón Ernesto
Ingrese el Apellido Paterno: Valdez
Ingrese el Apellido Materno o (dejar vacío o si no tiene): Felix
Ingrese la Fecha de Nacimiento (formato ddmmaaaa): 19081983
RFC calculado: VAFR830819

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

```

```

main.cpp
17 // Tomar la primera letra del apellido materno, se asigna "X" si no hay apellido
18 std::string primeraLetraApellidoMa;
19 if (apellidoMaMayus.empty()) {
20     primeraLetraApellidoMa = "X";
21 } else {
22     primeraLetraApellidoMa = apellidoMaMayus.substr(0, 1);
23 }
24 // Tomar las primera letras del nombre
25 std::string primeraLetrasNom = nomMayus.substr(0, 1);
26 // Tomar los últimos dígitos del año de nacimiento
27 std::string annusNac = fechaNacimiento.substr(6, 7);
28 // Tomar el mes de nacimiento
29
input
Ingrese el Nombre: Ramón
Ingrese el Apellido Paterno: Valdez
Ingrese el Apellido Materno o (dejar vacío o si no tiene):
Ingrese la Fecha de Nacimiento (formato ddmmaaaa): 19081985
RFC calculado: VAXR850819

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

```

Evidencia 2: Dev-C++: En este punto se ejecuta el programa de c++ usando uno del listado del IDE solicitado en el material de la materia de lenguaje de programación 1 anexando 2 pantallas de evidencia. La primera pantalla muestra el ingreso de todos los campos y el resurta del RFC. En la segunda evidencia se llenan los campos excepto el campo de apellido materno el cual asigna una letra X al no ser llenado dando el resultado esperado del RFC.

```

C:\Users\ramon.valdez\Desktop\Un\Periodo_2\05_Lenguajes_Programación\A3\RamonValdez_A3.cpp - [Executing] - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
(globals)
Proyecto Clases(Funciones) RamonValdez_A3.cpp
1 //Actividad 3. Cálculo de RFC
2 //Actividad 3. Aplicación para Cálculo de RFC
3 #include <iostream>
4 #include <string>
5 #include <algorithm>
6
7 std::string calcRFC(const std::string& nom, const std::string& apellidoPa,
8 // Convertir los apellidos, nombre a mayúsculas y eliminar espacios.
9 std::string apellidoPaMayus = apellidoPa;
10 std::string apellidoMaMayus = apellidoMa;
11 std::string nomMayus = nom;
12 std::transform(apellidoPaMayus.begin(), apellidoPaMayus.end(), apellidoPaMayus.begin(), ::toupper);
13 std::transform(apellidoMaMayus.begin(), apellidoMaMayus.end(), apellidoMaMayus.begin(), ::toupper);
14 std::transform(nomMayus.begin(), nomMayus.end(), nomMayus.begin(), ::toupper);
15 // Tomar las primeras 2 letras del apellido paterno
16 std::string primerasLetrasApellidoPa = apellidoPaMayus.substr(0, 2);
17
18
19 Ingrese el Nombre: Ramón Ernesto
20 Ingrese el Apellido Paterno: Valdez
21 Ingrese el Apellido Materno o (dejar vacío si no tiene): Felix
22 Ingrese la Fecha de Nacimiento (formato ddmmaaaa): 19081983
23 RFC calculado: VAFR830819
24
25 -----
Process exited after 29.63 seconds with return value 0
Presione una tecla para continuar . . .
  
```

```

C:\Users\ramon.valdez\Desktop\Un\Periodo_2\05_Lenguajes_Programación\A3\RamonValdez_A3.cpp - [Executing] - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
(globals)
Proyecto Clases(Fun) RamonValdez_A3.cpp
17 // Tomar la primera letra del apellido materno, se asigna "X" si no hay apell
18 std::string primeraLetraApellidoMa;
19 if (apellidoMaMayus.empty()) {
20     primeraLetraApellidoMa = "X";
21 } else {
22     primeraLetraApellidoMa = apellidoMaMayus.substr(0, 1);
23 }
24
25
26 Ingrese el Nombre: Ramón
27 Ingrese el Apellido Paterno: Valdez
28 Ingrese el Apellido Materno o (dejar vacío si no tiene):
29 Ingrese la Fecha de Nacimiento (formato ddmmaaaa): 19081985
30 RFC calculado: VAXR850819
31
32 -----
33 Process exited after 18.67 seconds with return value 0
34 Presione una tecla para continuar . . .
35 }
36 int
37

```

Conclusión

En conclusión, la programación a través de los años ha tomado un papel fundamental para el desarrollo de la especie humana ya que este grandioso proceso ha permitido el surgimiento de la llamada era informática o tecnológica, siendo fundamental para el desarrollo humano ya que nos permite hacer que las maquinas realicen procesos que los humanos no podemos hacer con gran facilidad e incluso muchas veces nos resultan imposibles de realizar. El cálculo de RFC en la vida laboral se puede realizar de manera rápida y sencilla sin tener que realizar trámites tardados en alguna dependencia de gobierno. El lenguaje de programación C es muy popular y podríamos decir que es el pilar de la programación estructurada, permitiéndonos simplificar la lectura del código; pero lo más importante es que, una función sencilla, que realiza una tarea específica, puede ser utilizada fácilmente en cualquier otro programa.

Referencias

(N.d.). Onlinegdb.com. Retrieved July 18, 2023, from https://www.onlinegdb.com/online_c_compiler

Hernández, I. (2017, October 12). La Programacion en la vida cotidiana. *Blogspot.com*.
<http://progratics.blogspot.com/2017/10/la-programacion-en-la-vida-cotidiana.html>

No title. (n.d.). Cplusplus.com. Retrieved July 29, 2023, from <https://cplusplus.com/reference/algorithm/>