



Universidad Mariano Gálvez de Guatemala
Sede de Villa Nueva, Guatemala

Ingeniería en Sistemas de la Información y Ciencia Computacional

Tema:
Laboratorio #5

Curso: Programación I
Docente: Ing. Carlos Alejandro Arias

Estudiante: Pablo Sebastián Quan Montenegro
Carné: 5090-23-2625

21/02/2024

Introducción

Se presenta un programa en C++ que tiene un menú interactivo con diversas opciones matemáticas. El código se enfoca en el desarrollo de funciones que realizan operaciones como la suma de números pares o impares, el cálculo del factorial de un número, la determinación de la cantidad de dígitos de un número, la ordenación de tres números de menor a mayor, y la suma de los dígitos de un número.

Código Comentado

```
1  #include <iostream>
2
3  using namespace std;
4
5  int sumaParImpar(int tope, int opcion);
6  int factorial(int tope);
7  int cantDigitos(int num);
8  void ordenarNumeros(int a, int b, int c);
9  int sumaDigitos(int num);
10
11 void repeticion();
12
13 char respuesta;
14 bool repetir;
```

En esta parte del código se incluyen las bibliotecas correspondientes, además de la función que dependiendo de la elección del usuario se estará presentando dentro del main().

También está el la función de void repeticion() en dado caso si el usuario desea realizar más de una vez alguna acción.

Función int sumaParImpar(int tope, int opcion);

```
105  
106 int sumaParImpar(int tope, int opcion){  
107     int sumatoria = 0;  
108  
109     if (opcion == 1){  
110         for (int x = 0; x <= tope; x +=2){  
111             sumatoria = sumatoria + x;  
112         }  
113         return sumatoria;  
114     }  
115  
116     else if (opcion == 2){  
117         for (int x = 1; x <= tope; x +=2){  
118             sumatoria = sumatoria + x;  
119         }  
120         return sumatoria;  
121     }  
122 }  
123
```

Esta función tiene como objetivo que a partir del número dado definido como el límite y la opción ingresada según sea par o impar, hará la sumatoria de n cantidad de números ya sea par o impar.

Función int factorial(int tope);

```
123  
124 int factorial(int tope){  
125     int producto = 1;  
126     for (int x = 1; x <= tope; x++){  
127         producto = producto * x;  
128     }  
129     return producto;  
130 }  
131
```

Esta función tiene como objetivo regresar el factorial de un número, multiplicando n cantidad de veces según lo elegido por el usuario.

Función `int cantDigitos(int num);`

```
131  
132 int cantDigitos(int num){  
133     int suma;  
134  
135     suma = 0;  
136     do{  
137         num = num / 10;  
138         suma += 1;  
139     }while(num > 0);  
140     return suma;  
141 }  
142
```

Esta función tiene como objetivo contar la cantidad de dígitos de un número.

Se hace un ciclo el cual cada vez que se realice una división, habrá un contador que se ira

sumando. Por ejemplo, en el caso de 120, siguiendo el procedimiento del `do while`, sería:

- 1er ciclo:
 $120 / 10 = 12$
suma (cantidad de dígitos) = $0+1$
- 2do ciclo:
 $12 / 10 = 1.2$ (pero terminara siendo 1 ya que no se consideran los decimales por ser de tipo `int`)
suma (cantidad de dígitos) = $1+1$
- 3er ciclo:
 $1 / 10 = 0.1$ (el cual daría como resultado 0)
suma (cantidad de dígitos) = $2+1$
- 4to ciclo: antes de comenzar se evalúa si el numero sigue siendo mayor a 0, como ya no lo es
se termina el ciclo.

Función void ordenarNumeros(int a, int b, int c);

```
170
171 void ordenarNumeros(int a, int b, int c){
172     bool cambio;
173     /*
174
175     El ciclo de do while evalua que cuando el numero de la izquierda es mayor al de la derecha
176     realice un cambio, si en dado caso se cumple con al menos una vez con esa condicion, existe
177     la certeza de que el ciclo se volvera a ejecutar hasta comprobar que todos los numeros esten
178     ordenados correctamente de menor a mayor.
179
180     */
181     do{
182         cambio = false;
183         if (a > b){
184             swap(a, b);
185             cambio = true;
186         }
187         if (b > c){
188             swap(b, c);
189             cambio = true;
190         }
191     }while(cambio == true);
192
193     cout << "\nEl orden de menor a mayor de los numeros ingresados es: " << a << ", " << b << ", " << c << endl;
194 }
```

El ciclo de do while evalúa que cuando el numero de la izquierda es mayor al de la derecha realice un cambio, si en dado caso se cumple con al menos una vez con esa condición, existe la certeza de que el ciclo se volverá a ejecutar hasta comprobar que todos los números estén ordenados correctamente de menor a mayor.

Función `int sumaDigitos(int num);`

```
195
196 int sumaDigitos(int num){
197
198     /*
199
200     Lo que se realiza con el operador de residuo de una division, es que dividiendo el numero por 10
201     se sabe que el residuo que queda es el digito de la unidad, y ese digito es el que se ira sumando,
202     luego el siguiente proceso es dividir por 10 el numero, sabiedno que por ejemplo 1023 / 10, por
203     haber sido declarado como entero y diviendolo por 10, el decimal 102.3 no sera tomado en cuenta
204     por lo que de esa forma se elimina el digito que ya no se desea.
205
206     */
207     int suma;
208     suma = 0;
209     do{
210         suma += num % 10;
211         num = num / 10;
212     }while(num > 0);
213
214     return suma;
215 }
216
```

Lo que se realiza con el operador de residuo de una división, es que dividiendo el número por 10 se sabe que el residuo que queda es el dígito de la unidad, y ese dígito es el que se irá sumando, luego el siguiente proceso es dividir por 10 el número, sabiendo que por ejemplo $1023 / 10$, por haber sido declarado como entero y diviendolo por 10, el decimal 102.3 no será tomado en cuenta por lo que de esa forma se elimina el dígito que ya no se desea.

Función void repeticion();

```
140
141 void repeticion(){
142     do{
143         system("cls");
144         cout << "Desea realizar otra accion? S/N: ";
145         cin >> respuesta;
146     } while(respuesta != 'n' and respuesta != 'N' and respuesta != 'S' and respuesta != 's');
147
148     if (respuesta == 's' or respuesta == 'S'){
149         repetir = true;
150     }
151     else if (respuesta == 'n' or respuesta == 'N'){
152         repetir = false;
153     }
154 }
```

Esta función tiene como propósito verificar si el usuario desea seguir dentro del menú y elegir otra opción de las que están disponibles, o sencillamente decide que no y se termine la ejecución del programa. Además, esta función se aplica cada vez que se termina de ejecutar cualquiera de las opciones disponibles del menú.

Función principal `int main();`

Menú:

```
15
16 int main(){
17     do{
18         system("cls");
19         int opcion;
20         cout << "\tMenu:\n\n";
21         cout << "1. Suma de pares o impares\n";
22         cout << "2. Factorial\n";
23         cout << "3. Cantidad de digitos de un numero\n";
24         cout << "4. Ordenar numeros de menor a mayor\n";
25         cout << "5. Suma de digitos de un numero\n";
26         cout << "6. Salir del menu\n\n";
27         cout << "Opcion a elegir: ";
28         cin >> opcion;
29         system("cls");
```

Switch

Case 1:

```
29     system("cls");
30     switch(opcion){
31     case 1:
32         int opcionSuma;
33         int numUsuario;
34         cout << "\tSuma de pares o impares\n\n";
35         cout << "1. Suma de pares\n2. Suma de impares\n\n";
36         cout << "Opcion a elegir: ";
37         cin >> opcionSuma;
38         if (opcionSuma == 1){
39             system("cls");
40             cout << "\tSuma de pares: \n";
41             cout << "Elige el numero hasta donde llegara la suma: ";
42             cin >> numUsuario;
43             cout << "\nLa sumatoria de los numeros pares hasta el " << numUsuario << " es de: " << sumaParImpar(numUsuario, opcionSuma) << endl;
44         }
45         else if (opcionSuma == 2){
46             system("cls");
47             cout << "\tSuma de impares: \n";
48             cout << "Elige el numero hasta donde llegara la suma: ";
49             cin >> numUsuario;
50             cout << "\nLa sumatoria de los numeros impares hasta el " << numUsuario << " es de: " << sumaParImpar(numUsuario, opcionSuma) << endl;
51         }
52         system("pause");
53         repeticion();
54         break;
```

Case 2:

```
54         break;
55     case 2:
56         int numFactorial;
57         cout << "\tFactorial\n";
58         cout << "Escribe un numero el cual se obtendra el factorial: ";
59         cin >> numFactorial;
60         cout << "\nEl factorial de " << numFactorial << " es: " << factorial(numFactorial) << endl;
61         system("pause");
62         repeticion();
63         break;
```

Case 3:

```
63         break;
64     case 3:
65         int numDigitos;
66         cout << "\tCantidad de digitos de un numero\n";
67         cout << "Escribe un numero: ";
68         cin >> numDigitos;
69         cout << "\nLa cantidad de digitos de " << numDigitos << " es de: " << cantDigitos(numDigitos) << " digitos" << endl;
70         system("pause");
71         repeticion();
72         break;
```

Case 4:

```
72         break;
73     case 4:
74         int num1, num2, num3;
75         cout << "\tOrdenar numeros de menor a mayor\n";
76         cout << "Escribe 3 numeros: \n\n";
77         cout << "Numero 1: ";
78         cin >> num1;
79         cout << "Numero 2: ";
80         cin >> num2;
81         cout << "Numero 3: ";
82         cin >> num3;
83         ordenarNumeros(num1, num2, num3);
84         system("pause");
85         repeticion();
86         break;
```

Case 5:

```
86         break;
87     case 5:
88         int numDigito;
89         cout << "\tSuma de digitos de un numero\n";
90         cout << "Escribe un numero: ";
91         cin >> numDigito;
92         cout << "\nLa suma de los digitos de " << numDigito << " es de: " << sumaDigitos(numDigito) << endl;
93         system("pause");
94         repeticion();
95         break;
```

Case 6:

```
96     case 6:
97         return 0;
```

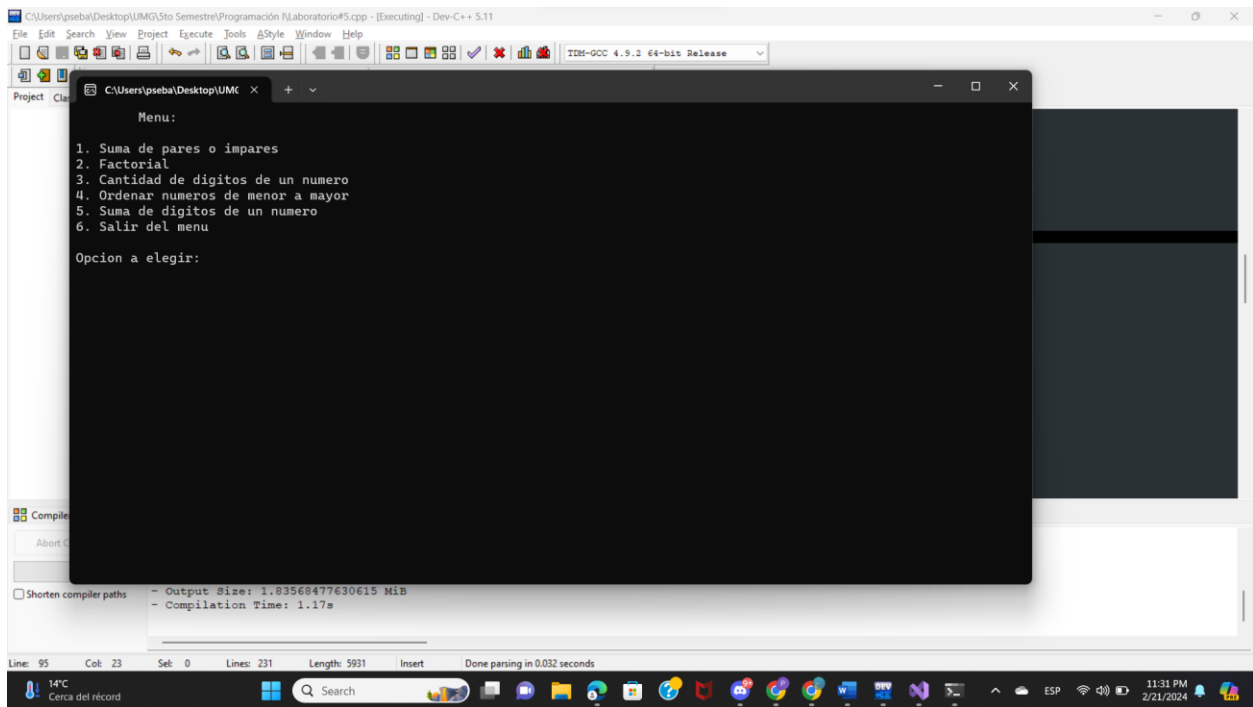
default:

```
98     default:
99         repetir = true;
```

Condición para que se repita el menú y fin del programa:

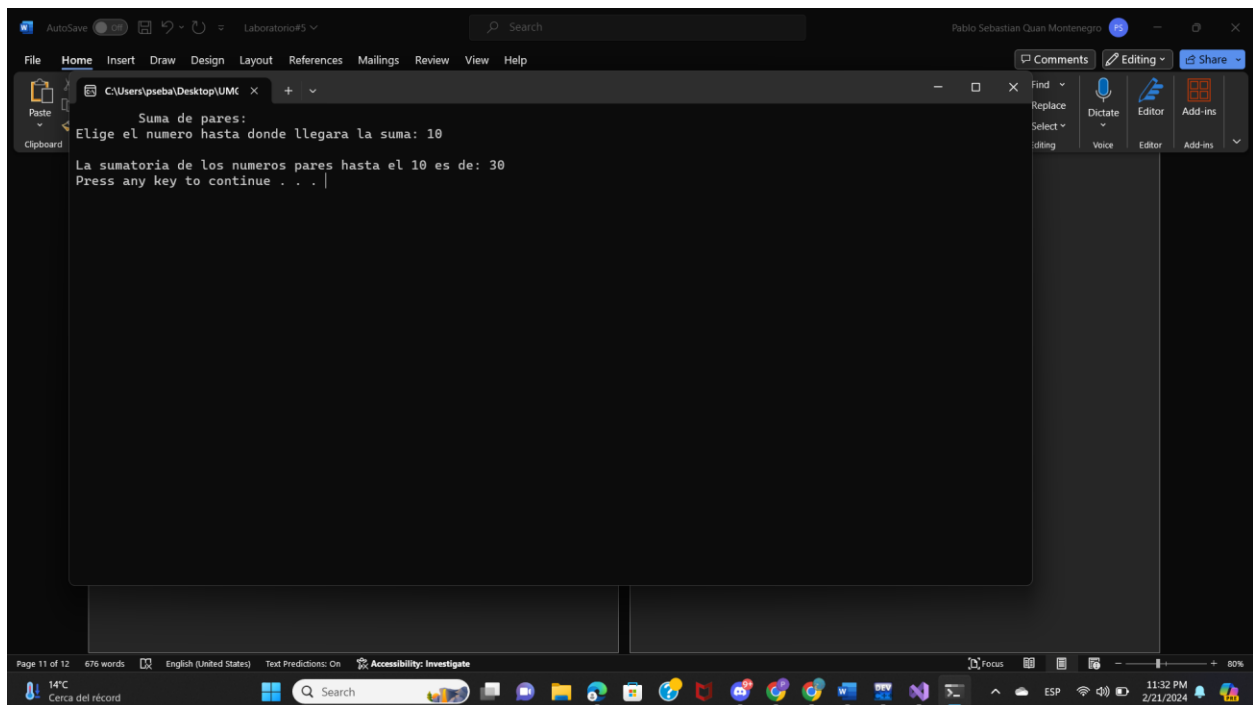
```
100     }
101     } while(repetir == true);
102
103     return 0;
104 }
105
```

Ejecución

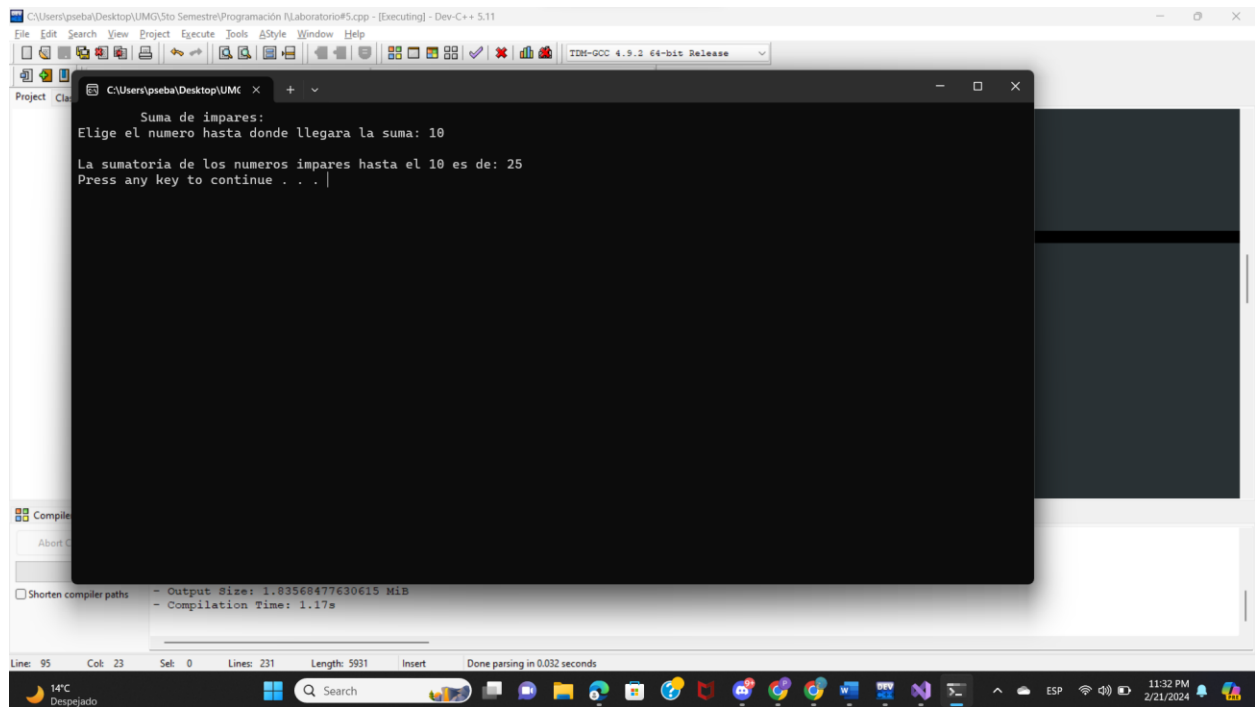


Función `int sumaParImpar(int tope, int opcion);`

Sumar pares:



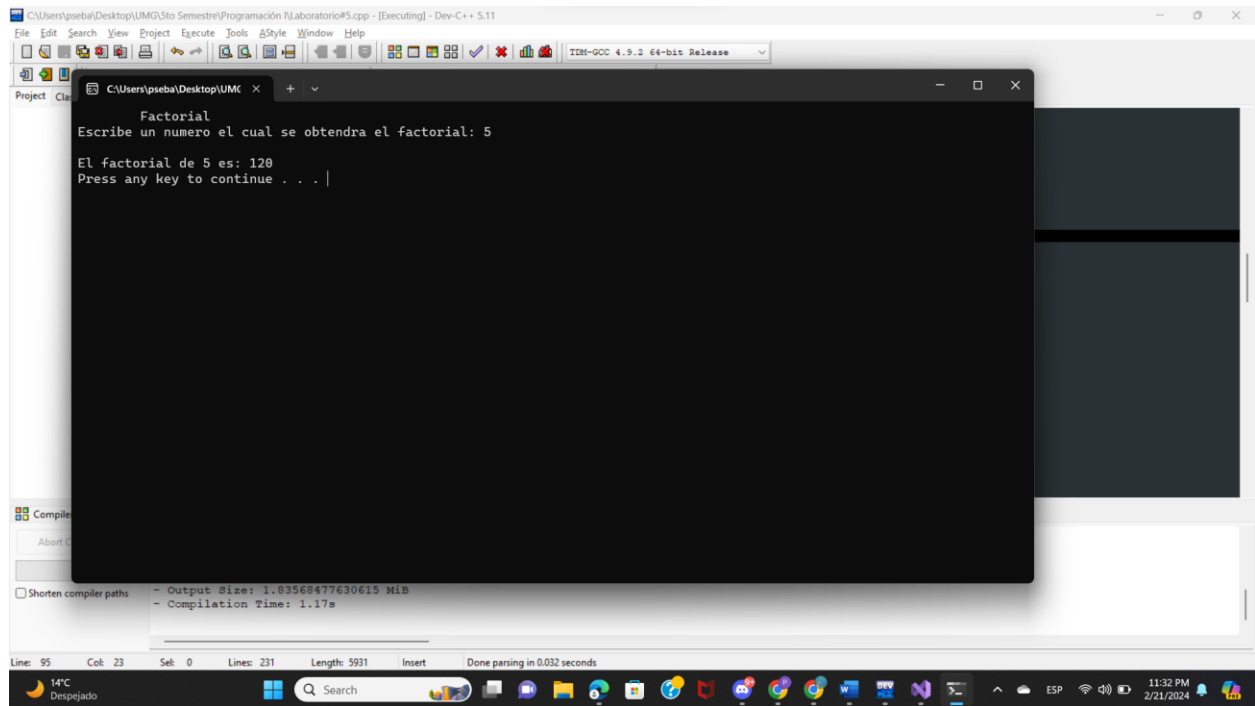
Sumar impares:



```
Suma de impares:  
Elige el numero hasta donde llegara la suma: 10  
  
La sumatoria de los numeros impares hasta el 10 es de: 25  
Press any key to continue . . . |
```

Output Size: 1.83568477630615 MiB
Compilation Time: 1.17s

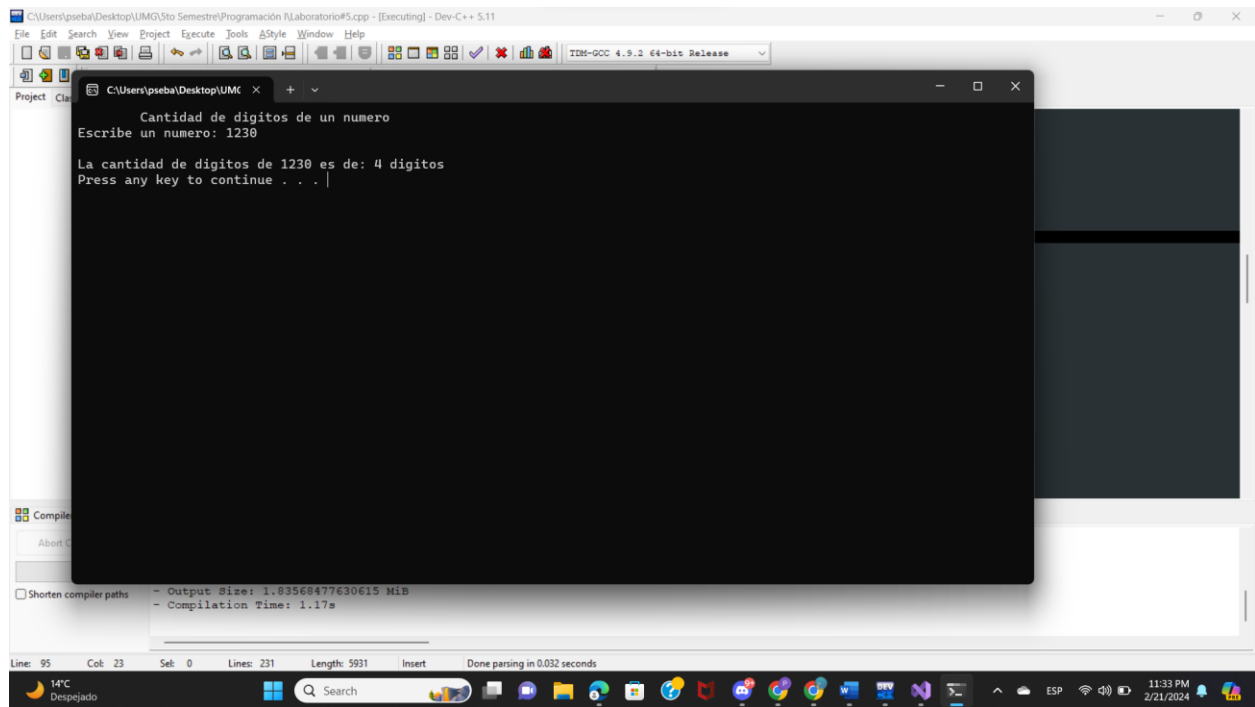
Función `int factorial(int tope);`



```
Factorial  
Escribe un numero el cual se obtendra el factorial: 5  
  
El factorial de 5 es: 120  
Press any key to continue . . . |
```

Output Size: 1.83568477630615 MiB
Compilation Time: 1.17s

Función int cantDigitos(int num);



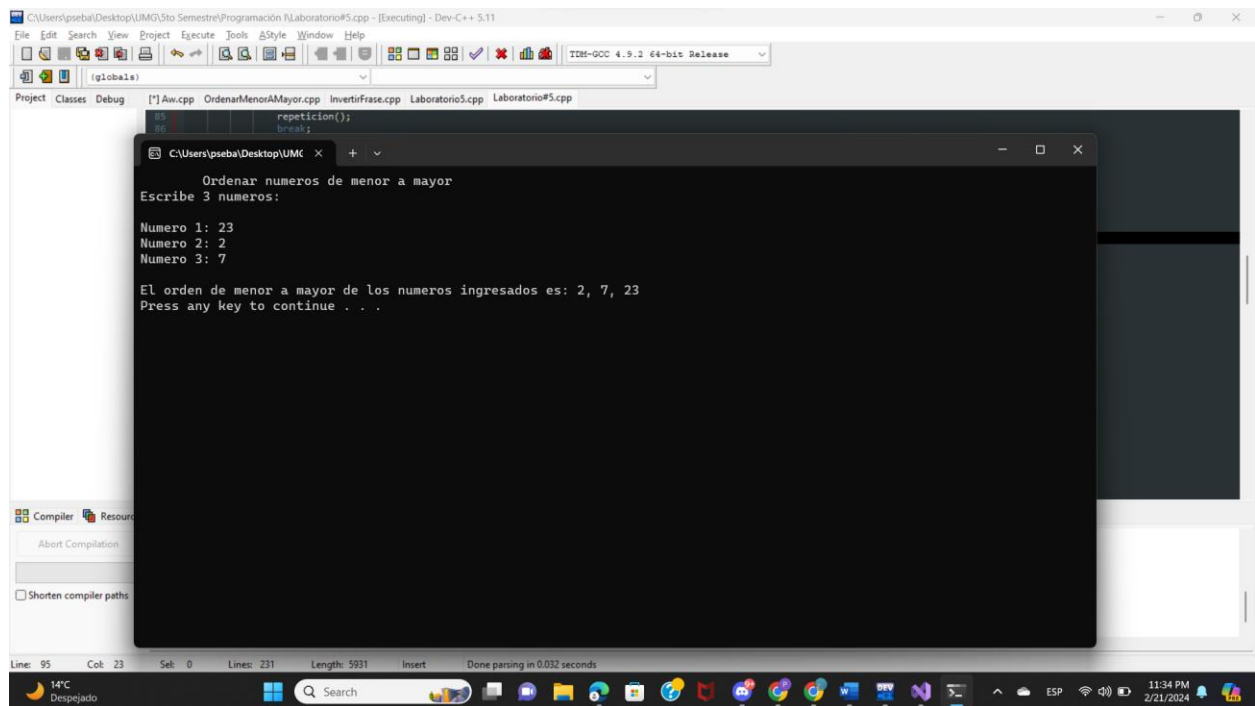
The screenshot shows the Dev-C++ IDE with a project named 'Laboratorio5.cpp'. A console window is open, displaying the following text:

```
Cantidad de digitos de un numero
Escribe un numero: 1230

La cantidad de digitos de 1230 es de: 4 digitos
Press any key to continue . . . |
```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), a toolbar, and a status bar at the bottom showing 'Line: 95', 'Col: 23', 'Sel: 0', 'Lines: 231', 'Length: 5931', and 'Done parsing in 0.032 seconds'. The Windows taskbar at the bottom shows the date and time as '11:33 PM 2/21/2024'.

Función void ordenarNumeros(int a, int b, int c);



The screenshot shows the Dev-C++ IDE with a project named 'Laboratorio5.cpp'. A console window is open, displaying the following text:

```
Ordenar numeros de menor a mayor
Escribe 3 numeros:

Numero 1: 23
Numero 2: 2
Numero 3: 7

El orden de menor a mayor de los numeros ingresados es: 2, 7, 23
Press any key to continue . . .
```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), a toolbar, and a status bar at the bottom showing 'Line: 95', 'Col: 23', 'Sel: 0', 'Lines: 231', 'Length: 5931', and 'Done parsing in 0.032 seconds'. The Windows taskbar at the bottom shows the date and time as '11:34 PM 2/21/2024'.

Función int sumaDigitos(int num);

The screenshot shows a C++ IDE with a project named "Laboratorio5.cpp". The code in the editor is as follows:

```
46 system("cls");  
47 cout << "La suma de los digitos de un numero es de: ";  
48
```

The output window displays the following text:

```
Suma de digitos de un numero  
Escribe un numero: 4125  
La suma de los digitos de 4125 es de: 12  
Press any key to continue . . . |
```

The IDE interface includes a menu bar (File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help), a toolbar, and a status bar at the bottom showing "Line: 95", "Col: 23", "Sel: 0", "Lines: 231", "Length: 5931", and "Done parsing in 0.032 seconds". The Windows taskbar at the bottom shows the date and time as "11:35 PM 2/21/2024".

Función void repeticion();

The screenshot shows the same C++ IDE with the same project. The code in the editor is as follows:

```
46 system("cls");  
47 cout << "La suma de los digitos de un numero es de: ";  
48
```

The output window displays the following text:

```
Desea realizar otra accion? S/N: |
```

The IDE interface is identical to the previous screenshot, showing the same menu bar, toolbar, and status bar. The Windows taskbar at the bottom shows the date and time as "11:35 PM 2/21/2024".

Conclusión

Durante el ejercicio de C++ se ha aportado conocimiento en la implementación de un programa modular con un menú interactivo y funciones específicas para realizar diversas operaciones matemáticas. Se aplicó conceptos fundamentales de programación estructurada, como la modularidad, la utilización de funciones, estructuras de control de flujo y el manejo de variables.

Link Github

<https://github.com/pabloquan/Programaci-n-Laboratorio-5>