|  |  |  |
| --- | --- | --- |
|  | **Carátula para entrega de prácticas** | |
| Facultad de Ingeniería | | Laboratorio de docencia |

Laboratorios de computación

salas A y B

|  |  |
| --- | --- |
| *Profesor:* | Claudia Rodríguez Espino |
| *Asignatura:* | Fundamentos de programación |
| *Grupo:* | 1104 |
| *No de Práctica(s):* | 6 |
| *Integrante(s):* | Aguilera Valderrama Alexis Fernando |
|  |  |
| *No. de Equipo de cómputo empleado* | 1 |
| *Semestre:* | 1 |
| *Fecha de entrega:* | 29/09/2018 |
| *Observaciones:* |  |
|  |  |

CALIFICACIÓN: \_\_\_\_\_\_\_\_\_\_

**Objetivos:**

Elaborar programas en lenguaje C utilizando las instrucciones de control de tipo secuencia, para realizar la declaración de variables de diferentes tipos de datos, así como efectuar llamadas a funciones externas de entrada y salida para asignar y mostrar valores de variables y expresiones.

**Desarrollo:**

Primeramente se mencionó que la parte de programación pertenece a las etapas de **codificación y pruebas.**

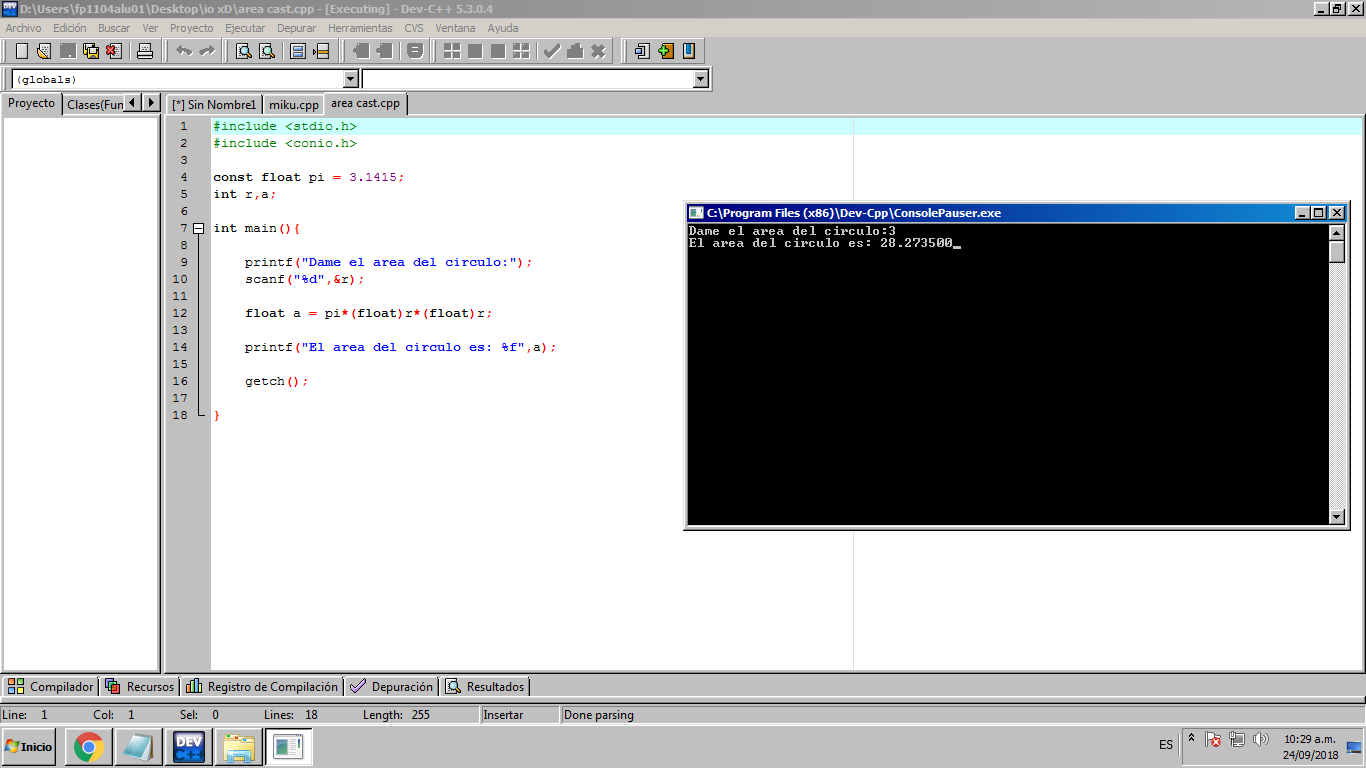
Para la elaboración de ésta práctica se utilizó el editor de texto y el compilador que están implementados en el software DevC++.

Se analizó la sintaxis más básica que tiene ofrece C para llevar acabo ciertos procesos tales como:

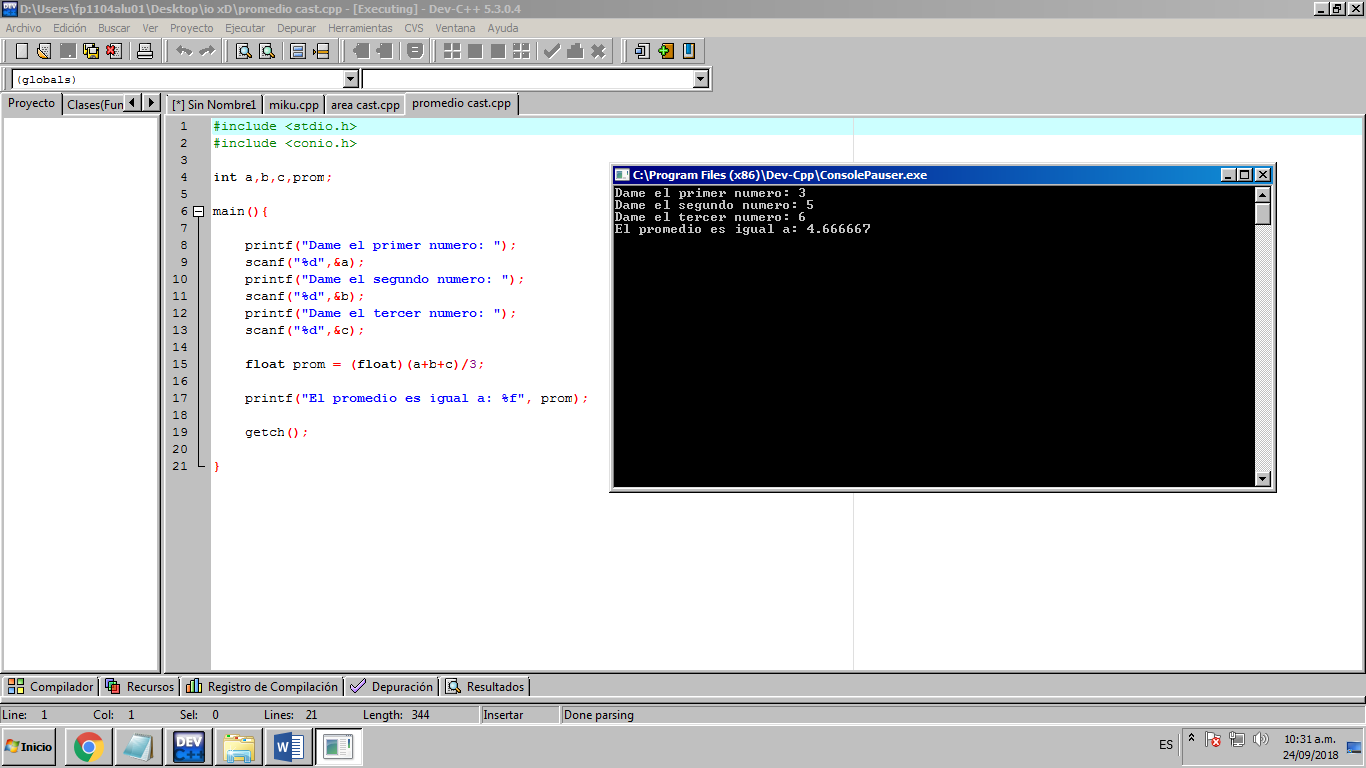
* Implementación de bibliotecas
* La declaración de variables con un identificador.
* Entrada y salida de datos.
* Modificadores de alcance.
* Operadores aritméticos y lógicos.
* Modelo o cast.
* Expresiones lógicas.

Con todo esto aprendido se logró recrear programas hechos en la práctica pasada, tales como el área del círculo, promedio de tres números y uno extra para sacar dinero anual acumulado conociendo años, deposito inicial y tasa de interés; todos ellos con la diferencia que se utilizó “cast” para convertir un tipo de dato a otro.

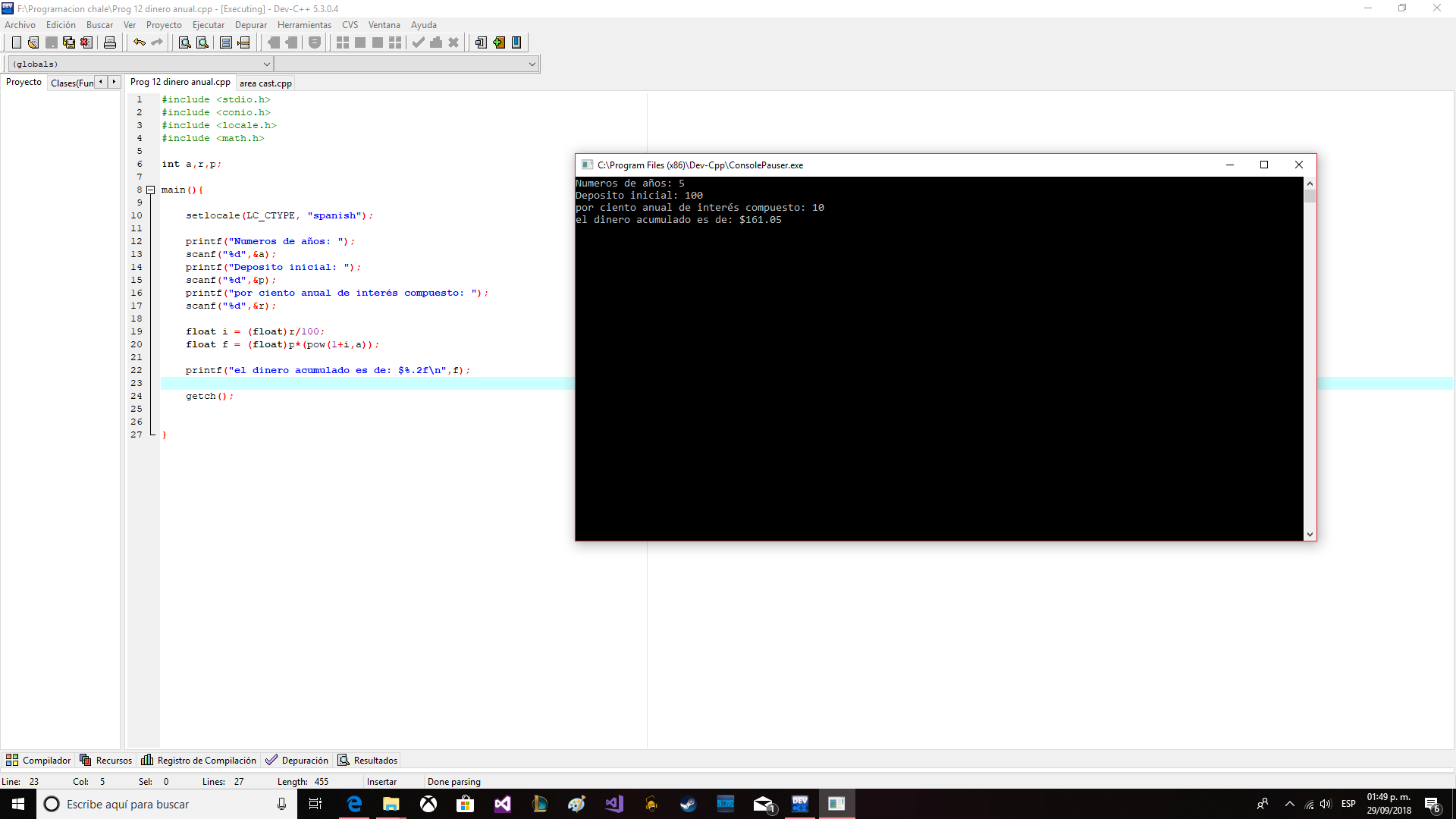
Área del círculo:



Promedio de 3 números:



Dinero acumulado:



**Conclusión:**

Se logró hacer con mucha facilidad los 3 programas propuestos en DevC++ ya que se habían utilizado para prácticas anteriores.

También el “cast” se implementó de manera correcta para convertir el tipo de dato entero a flotante y guardarlo en otro flotante.