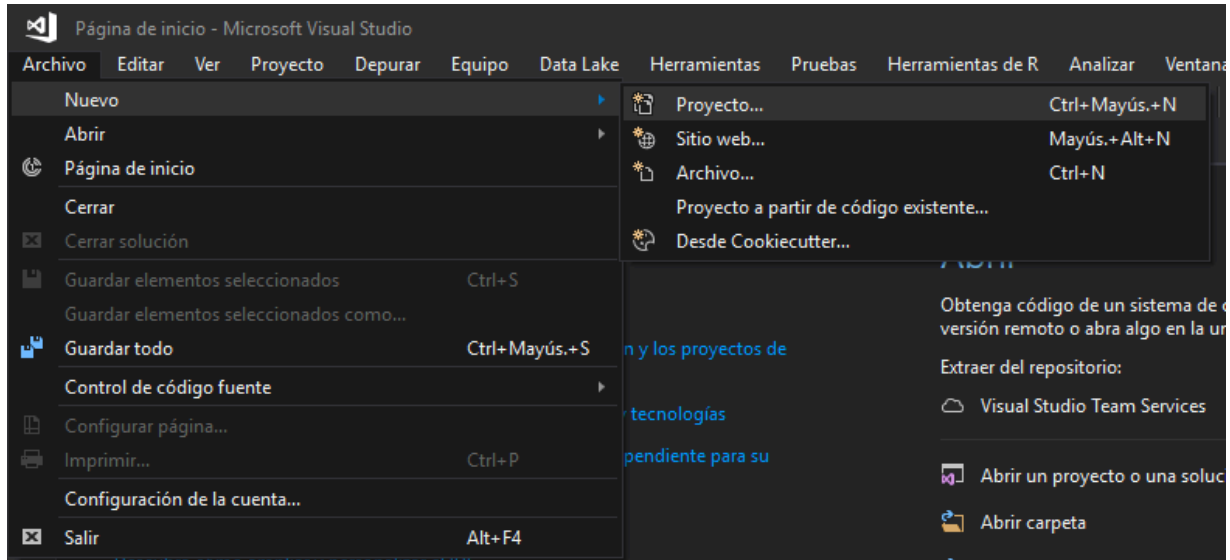


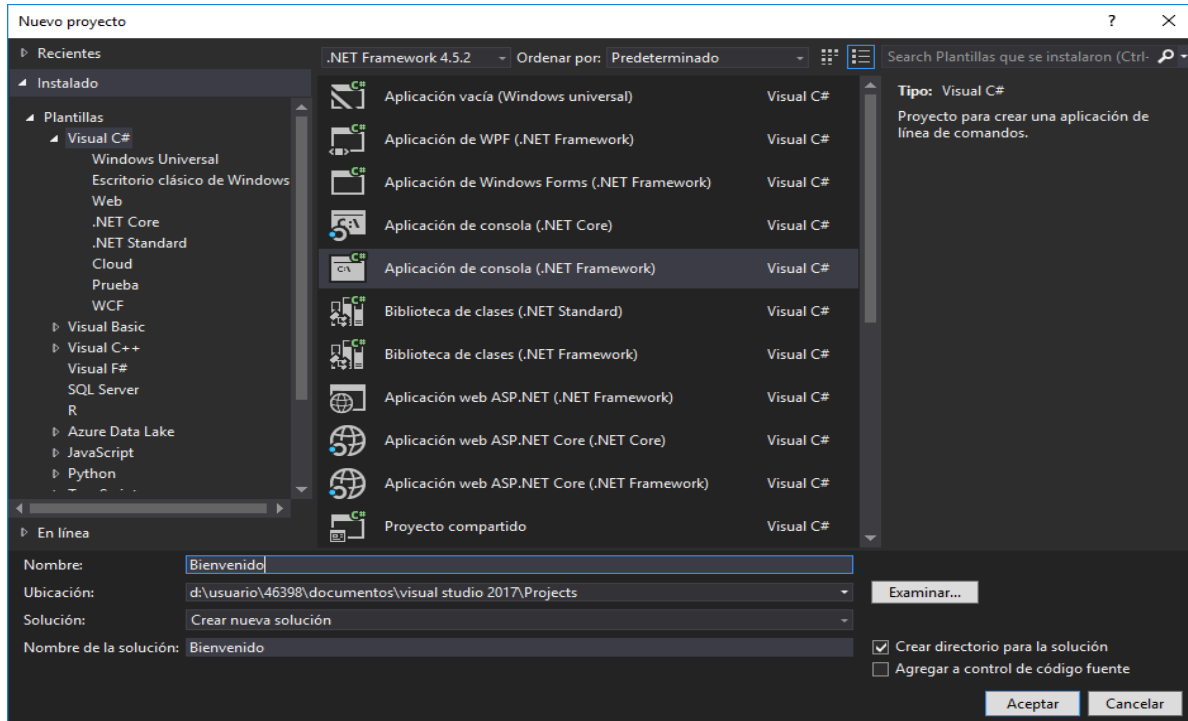
INICIO DE LA PROGRAMACIÓN CON C#

Pasos para crear un programa con C# Modo Consola

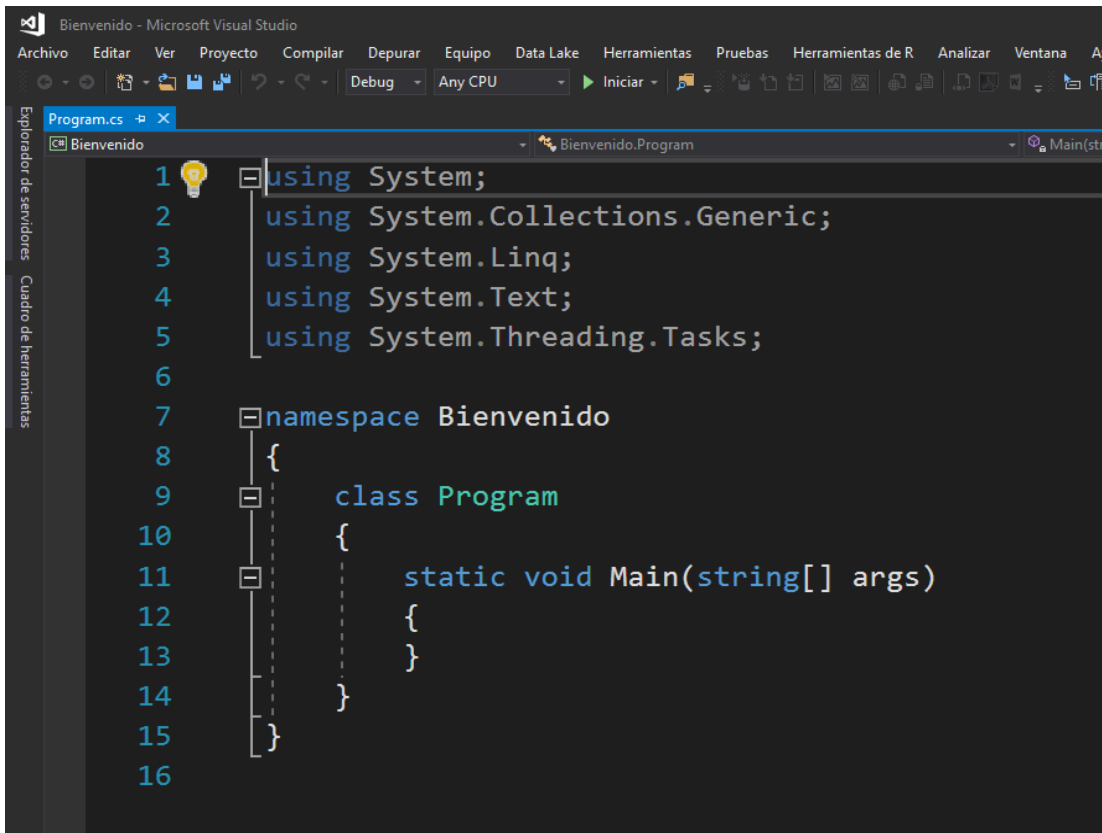
Crear un nuevo proyecto:



Seleccionar C#, Aplicación de Consola:



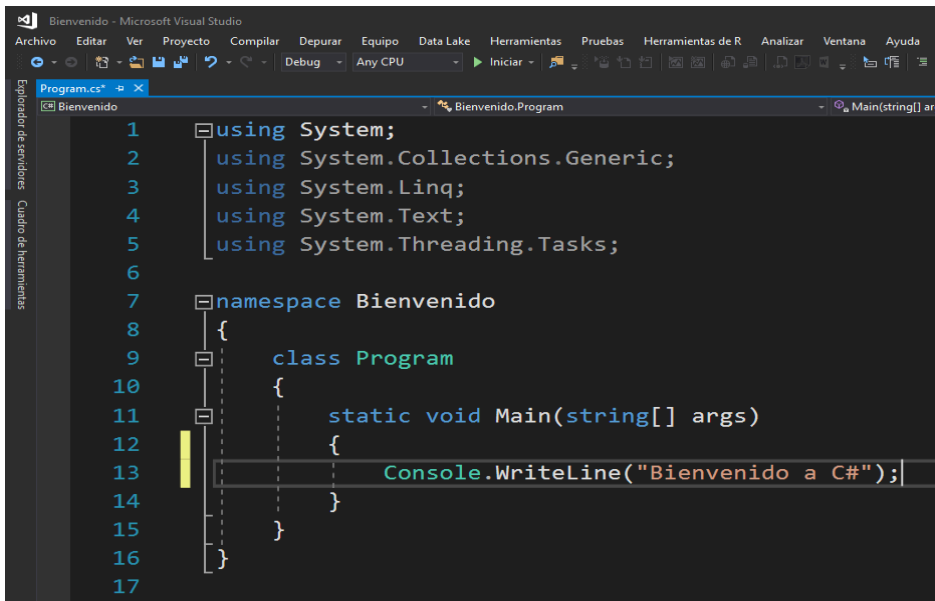
Consola de inicio, Librerías, Class Program, static void Main:



The screenshot shows the Microsoft Visual Studio IDE with a project named 'Bienvenido'. The 'Program.cs' file is open, displaying the following C# code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Bienvenido
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13        }
14    }
15 }
16
```

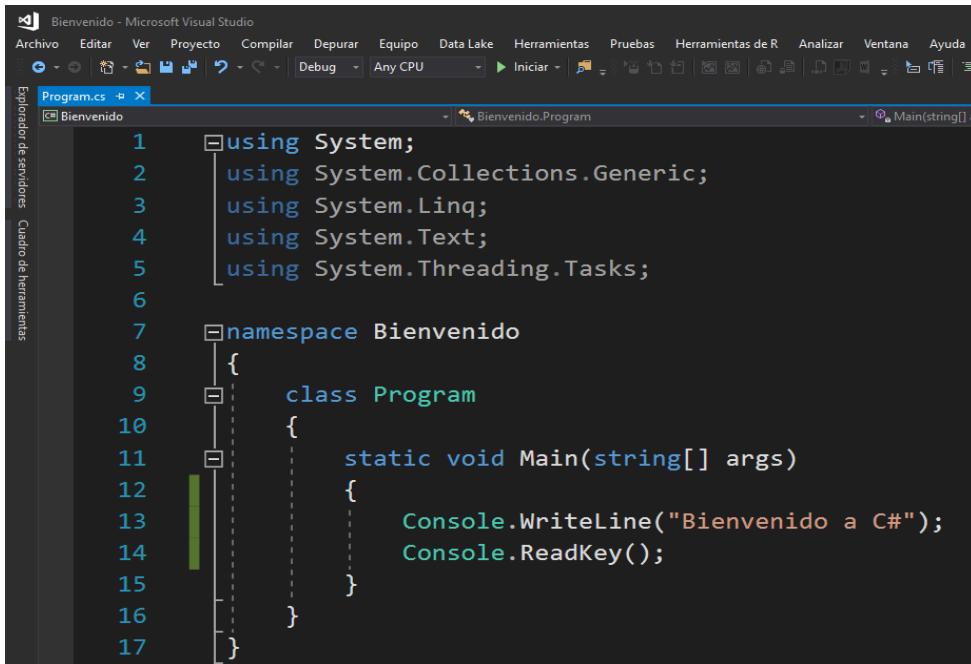
Mensaje de Bienvenido:



The screenshot shows the same Visual Studio IDE, but now the 'Main' method in the 'Program' class has been updated to include a welcome message. The code is as follows:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Bienvenido
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Console.WriteLine("Bienvenido a C#");
14        }
15    }
16 }
17
```

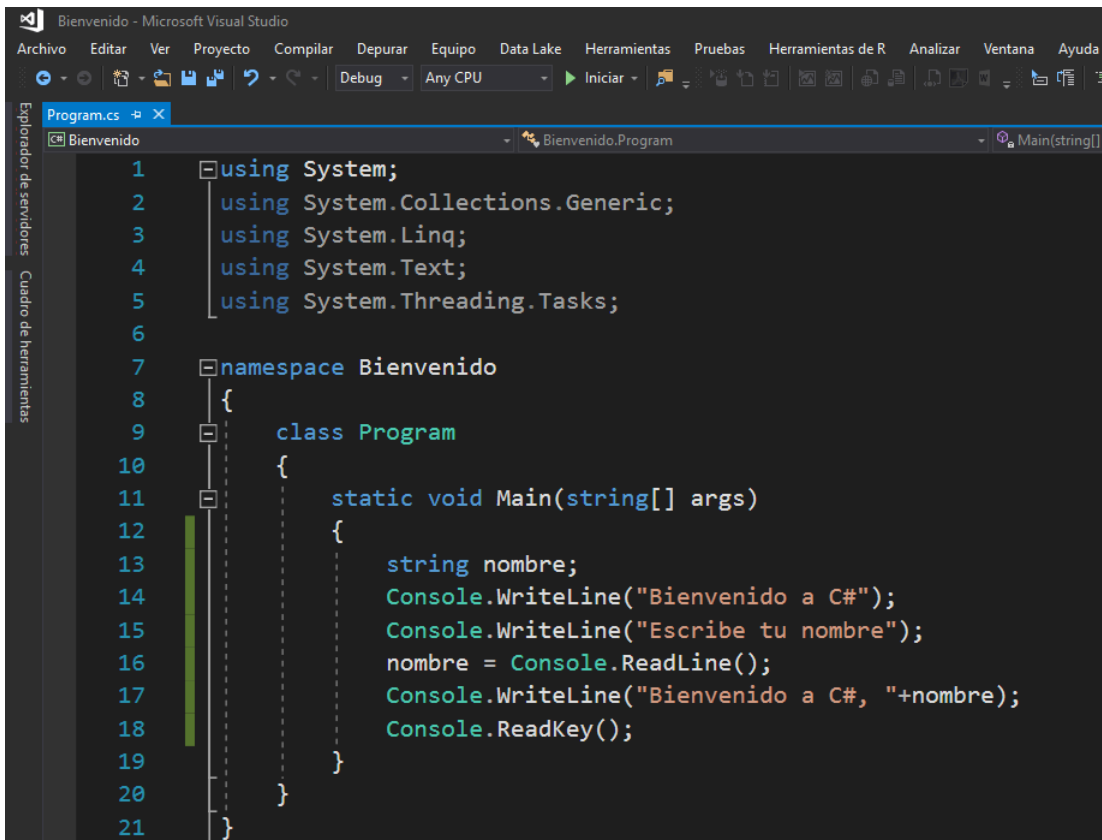
Agregamos pausa a la consola:



The screenshot shows the Microsoft Visual Studio interface with a C# file named Program.cs. The code defines a namespace 'Bienvenido' containing a class 'Program' with a static method 'Main'. The 'Main' method writes 'Bienvenido a C#' to the console and then calls 'Console.ReadKey()' to pause the program.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Bienvenido
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            Console.WriteLine("Bienvenido a C#");
14            Console.ReadKey();
15        }
16    }
17 }
```

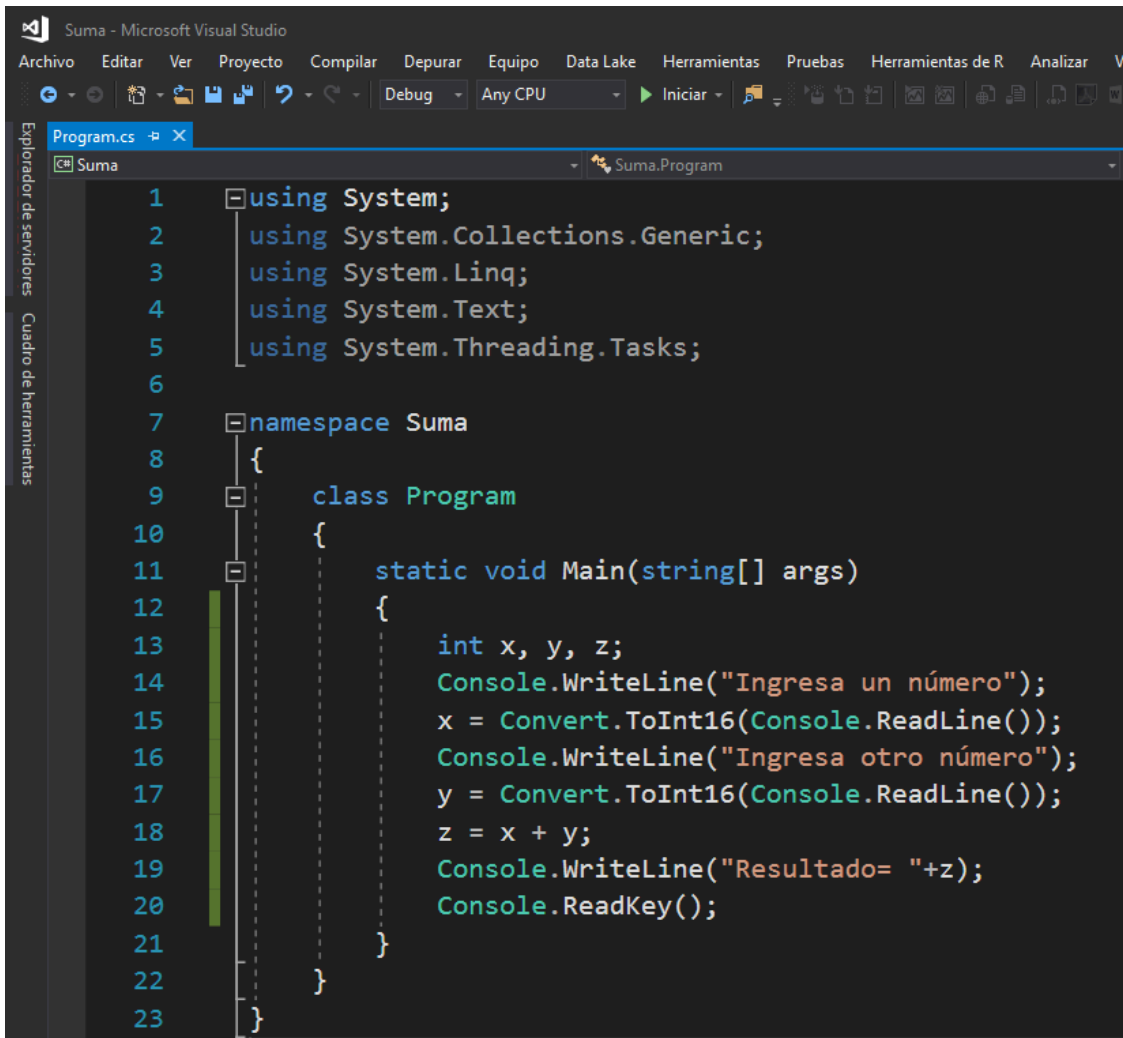
Uso de una variable en el programa, almacenamiento de información y su impresión:



The screenshot shows the Microsoft Visual Studio interface with a C# file named Program.cs. The code defines a namespace 'Bienvenido' containing a class 'Program' with a static method 'Main'. The 'Main' method declares a string variable 'nombre', writes 'Bienvenido a C#' and 'Escribe tu nombre' to the console, reads a line of input into 'nombre', writes 'Bienvenido a C#, ' + nombre to the console, and then calls 'Console.ReadKey()' to pause the program.

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Bienvenido
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            string nombre;
14            Console.WriteLine("Bienvenido a C#");
15            Console.WriteLine("Escribe tu nombre");
16            nombre = Console.ReadLine();
17            Console.WriteLine("Bienvenido a C#, " + nombre);
18            Console.ReadKey();
19        }
20    }
21 }
```

Programa que suma 2 números:



The image shows a screenshot of the Microsoft Visual Studio IDE. The title bar at the top reads "Suma - Microsoft Visual Studio". The menu bar includes "Archivo", "Editar", "Ver", "Proyecto", "Compilar", "Depurar", "Equipo", "Data Lake", "Herramientas", "Pruebas", "Herramientas de R", "Analizar", and "Ver". The toolbar below the menu bar contains icons for opening files, saving, undo, redo, and running the program. The "Debug" dropdown menu is set to "Any CPU", and the "Iniciar" button is highlighted. The Explorer pane on the left shows the project structure with "Suma" and "Suma.Program". The Solution Explorer on the right shows the "Program.cs" file. The main editor window displays the following C# code:

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Suma
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             int x, y, z;
14             Console.WriteLine("Ingresa un número");
15             x = Convert.ToInt16(Console.ReadLine());
16             Console.WriteLine("Ingresa otro número");
17             y = Convert.ToInt16(Console.ReadLine());
18             z = x + y;
19             Console.WriteLine("Resultado= "+z);
20             Console.ReadKey();
21         }
22     }
23 }
```

Operaciones básicas

```
static void Main(string[] args)
{
    int NUM1, NUM2, RESUL;
    string linea;
    Console.Write("PRIMER NÚMERO :");
    linea = Console.ReadLine();
    NUM1 = int.Parse(linea);
    Console.Write("SEGUNDO NÚMERO :");
    linea = Console.ReadLine();
    NUM2 = int.Parse(linea);
    Console.WriteLine();
    RESUL = NUM1 + NUM2;
    Console.WriteLine("LA SUMA ES {0}: ", RESUL);
    RESUL = NUM1 - NUM2;
    Console.WriteLine("LA RESTA ES: {0} - {1} = {2} ", NUM1, NUM2, RESUL);
    RESUL = NUM1 * NUM2;
    Console.WriteLine("LA MULTIPLICACIÓN ES: " + RESUL);
    RESUL = NUM1 / NUM2;
    Console.WriteLine("LA DIVISIÓN ES: " + RESUL);
    RESUL = NUM1 % NUM2;
    Console.WriteLine("EL RESIDUO ES: " + RESUL);
    Console.Write("Pulse una Tecla:"); Console.ReadLine();
}
```

COMPRA EN RESTAURANT

```
static void Main(string[] args)
{
    byte CANB, CANH, CANP;
    double APAGAR;
    string linea;
    const double PRECIOB = 0.8;
    const double PRECIOH = 2;
    const double PRECIOP = 1.2;
    Console.Write("CANTIDAD DE HAMBURGUESAS :"); linea = Console.ReadLine();
    CANH = byte.Parse (linea);
    Console.Write("CANTIDAD DE PAPAS :"); linea = Console.ReadLine();
    CANP = byte.Parse (linea);
    Console.Write("CANTIDAD DE BEBIDAS :"); linea = Console.ReadLine();
    CANB = byte.Parse (linea);
    Console.WriteLine();
    APAGAR = (CANH * PRECIOH) + (CANP * PRECIOP) + (CANB * PRECIOB);
    Console.WriteLine("VALOR A PAGAR: " + APAGAR);
    Console.Write("Pulse una Tecla:"); Console.ReadLine();
}
```

FORMATOS DE SALIDA

```
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            double BASE, ALTURA, RESULT;
            string linea;
            Console.Write("DIGITE LA BASE :"); linea = Console.ReadLine();
            BASE = double.Parse (linea);
            Console.Write("DIGITE LA ALTURA:"); linea = Console.ReadLine();
            ALTURA= double.Parse (linea);
            RESULT = (BASE * ALTURA) / 2;
            Console.WriteLine("AREA TRIANGULO :" + String.Format("{0:####.00}", RESULT));
            Console.WriteLine("AREA TRIANGULO :" + String.Format("{0:c}", RESULT));
            Console.WriteLine("AREA TRIANGULO :" + String.Format("{0:f}", RESULT));
            Console.WriteLine("AREA TRIANGULO :" + String.Format("{0:g}", RESULT));
            Console.WriteLine();
            Console.WriteLine("HOY ES: " + String.Format("Hoy es {0:F}", DateTime.Now));
            Console.WriteLine("HOY ES: " + String.Format("Hoy es {0:dddd}{0:dd/MM/yy}"));
            Console.WriteLine("Pulse una Tecla:"); Console.ReadLine();
        }
    }
}
```

EJERCICIO PROPUESTO: El usuario debe ingresar dos números y el programa mostrará el resultado de la operación $(a+b)*(a-b)$

Ejercicios: Modo Consola.

- Dada la matrícula y 3 calificaciones de un alumno obtenidas a lo largo del semestre, que imprima la matricula del alumno y su promedio de calificaciones.
- Elabora un programa que convierta un x número de galones en litros, ten en cuenta que hay 3.7854 litros en un galón.
- Dado un número entero positivo calcular e imprimir el cuadrado y el cubo de ese número.
- Calcula el número de segundos que hay en un determinado número de días.

CONDICIONES

Cuando tenemos más de un camino para encontrar la solución a un problema, es necesario aplicar las condiciones, para ello utilizamos a los operadores de relación:

Operador algebraico estándar de igualdad o de relación	Operador de igualdad o de relación en C	Ejemplo de una condición en C	Significado de la condición en C
<i>Operadores de igualdad</i>			
=	==	x == y	x es igual que y
≠	!=	x != y	x no es igual que y
<i>Operadores de relación</i>			
>	>	x > y	x es mayor que y
<	<	x < y	x es menor que y
≥	>=	x ≥ y	x es mayor o igual que y
≤	<=	x ≤ y	x es menor o igual que y

Operadores de igualdad y de relación.

Veamos unos ejemplos condicionales if en modo consola.

Condición del alumno

```
Program.cs x
calificación calificación.Program Main(string[])
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace calificación
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            float promedio;
14            Console.WriteLine("Programa que evalúa la condición de un alumno\n");
15            Console.WriteLine("Ingresa tu promedio final de la materia");
16            promedio = Convert.ToSingle(Console.ReadLine());
17            if (promedio < 7)
18            {
19                Console.WriteLine("Estás reprobado "+promedio);
20            }
21            else if(promedio >= 7)
22            {
23                Console.WriteLine("Aprobado "+promedio);
24            }
25            Console.ReadKey();
26        }
27    }
28 }
```

Calificación de un alumno

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace calificación
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            float promedio;
14            Console.WriteLine("Programa que evalúa la calificación de un alumno\n");
15            Console.WriteLine("Ingresa el promedio final de la materia");
16            promedio = Convert.ToSingle(Console.ReadLine());
17            if (promedio < 7 && promedio >= 0)
18            {
19                Console.WriteLine("Estás reprobado ");
20            }
21            else if (promedio >= 7 && promedio < 7.5)
22            {
23                Console.WriteLine("Promedio Final = 7");
24            }
25            else if (promedio >= 7.5 && promedio < 8.5)
26            {
27                Console.WriteLine("Promedio Final = 8 ");
28            }
29            else if (promedio >= 8.5 && promedio < 9.5)
30            {
31                Console.WriteLine("Promedio Final = 9");
32            }
33            else if (promedio >= 9.5 && promedio <= 10)
34            {
35                Console.WriteLine("Promedio Final = 10");
36            }
37            else if (promedio < 0 || promedio > 10)
38            {
39                Console.WriteLine("Valor no válido");
40            }
41            Console.ReadKey();
42        }
43    }
44 }
```

EJERCICIO PROPUESTO: Crear un programa que pida al usuario su nombre, y le diga "Hola" si se llama "David", o bien le diga "No te conozco" si teclea otro nombre.

Ejercicios:

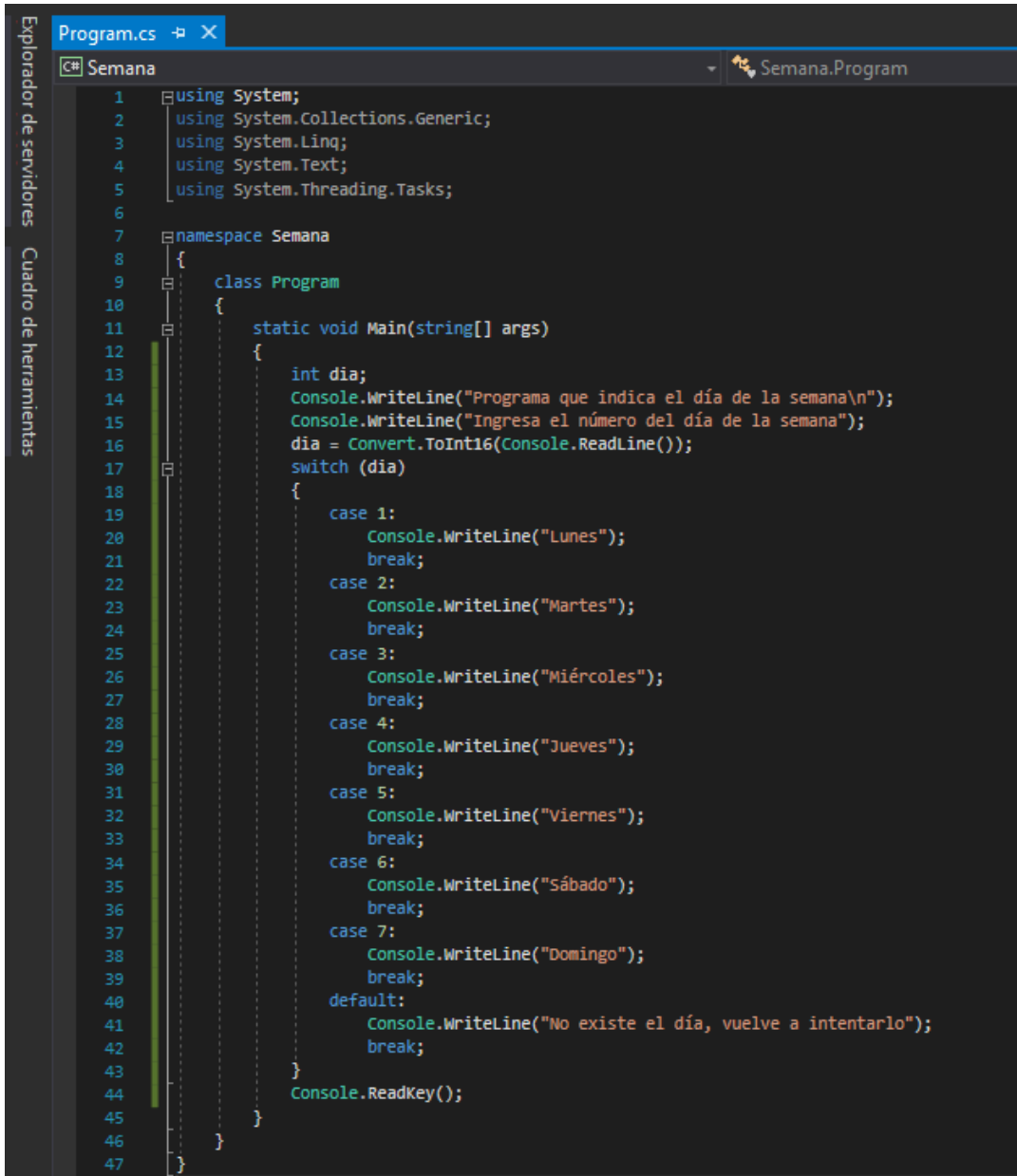
- Elabora un programa llamado Beca, el cual, dependiendo del promedio final de la materia asigna un porcentaje de beca:
 - 10, tiene 50% de Beca.
 - 9, el 30%.
 - 8, solo el 10%.
 - 7, no tiene beca.
 - Menor que 7, está reprobado, no tiene beca.
- Por último, el programa solo acepta valores dentro del rango de 0 a 10.

- Dados los tres lados de un triángulo determina el tipo de triángulo que se trate, Equilátero, Escaleno o Isósceles.

Estructura de Selección Múltiple

Cuando existen más de dos posibles soluciones a un programa, utilizamos la estructura de selección múltiple (switch):

Día de la semana:

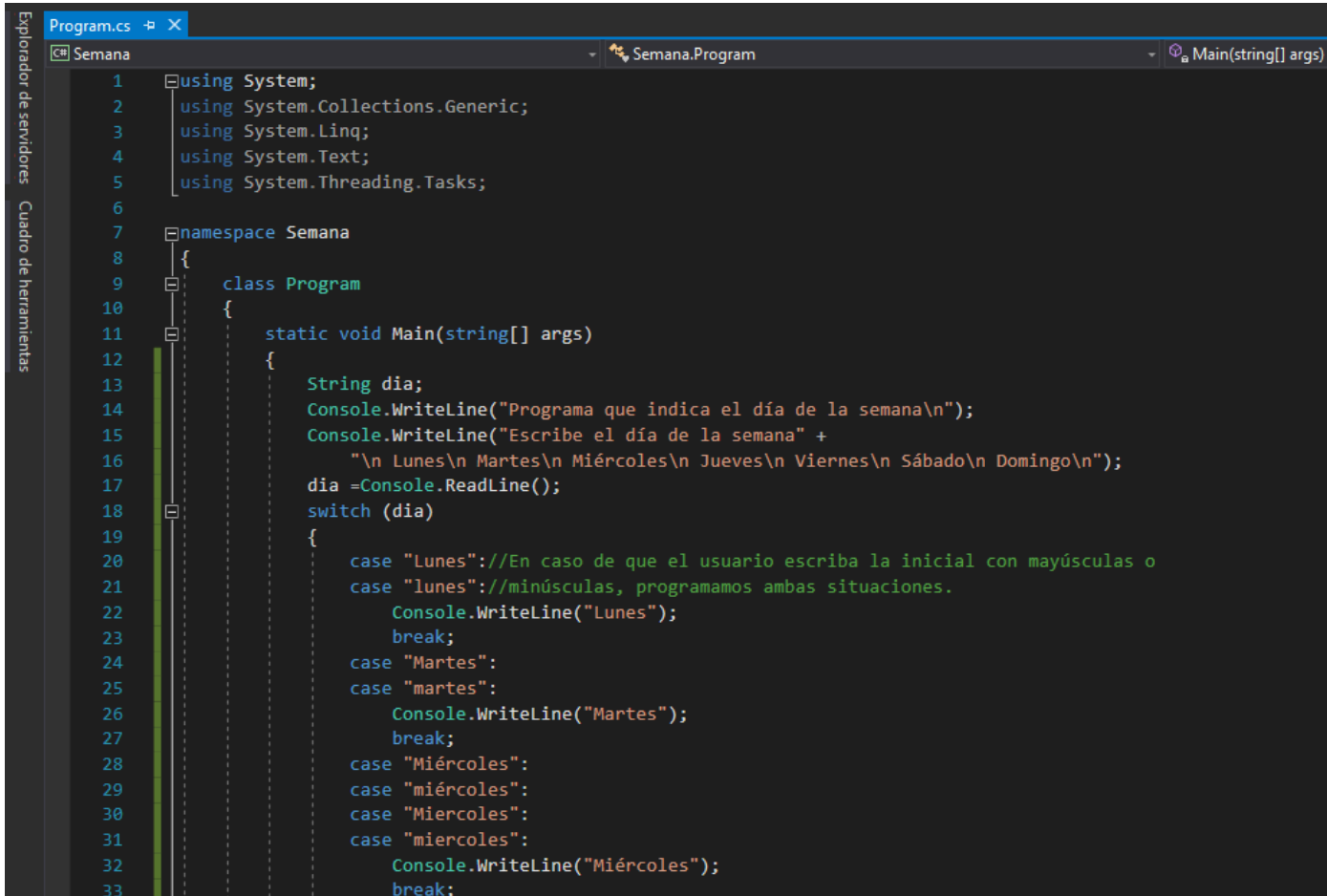


```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Semana
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             int dia;
14             Console.WriteLine("Programa que indica el día de la semana\n");
15             Console.WriteLine("Ingresa el número del día de la semana");
16             dia = Convert.ToInt16(Console.ReadLine());
17             switch (dia)
18             {
19                 case 1:
20                     Console.WriteLine("Lunes");
21                     break;
22                 case 2:
23                     Console.WriteLine("Martes");
24                     break;
25                 case 3:
26                     Console.WriteLine("Miércoles");
27                     break;
28                 case 4:
29                     Console.WriteLine("Jueves");
30                     break;
31                 case 5:
32                     Console.WriteLine("Viernes");
33                     break;
34                 case 6:
35                     Console.WriteLine("Sábado");
36                     break;
37                 case 7:
38                     Console.WriteLine("Domingo");
39                     break;
40                 default:
41                     Console.WriteLine("No existe el día, vuelve a intentarlo");
42                     break;
43             }
44             Console.ReadKey();
45         }
46     }
47 }
```

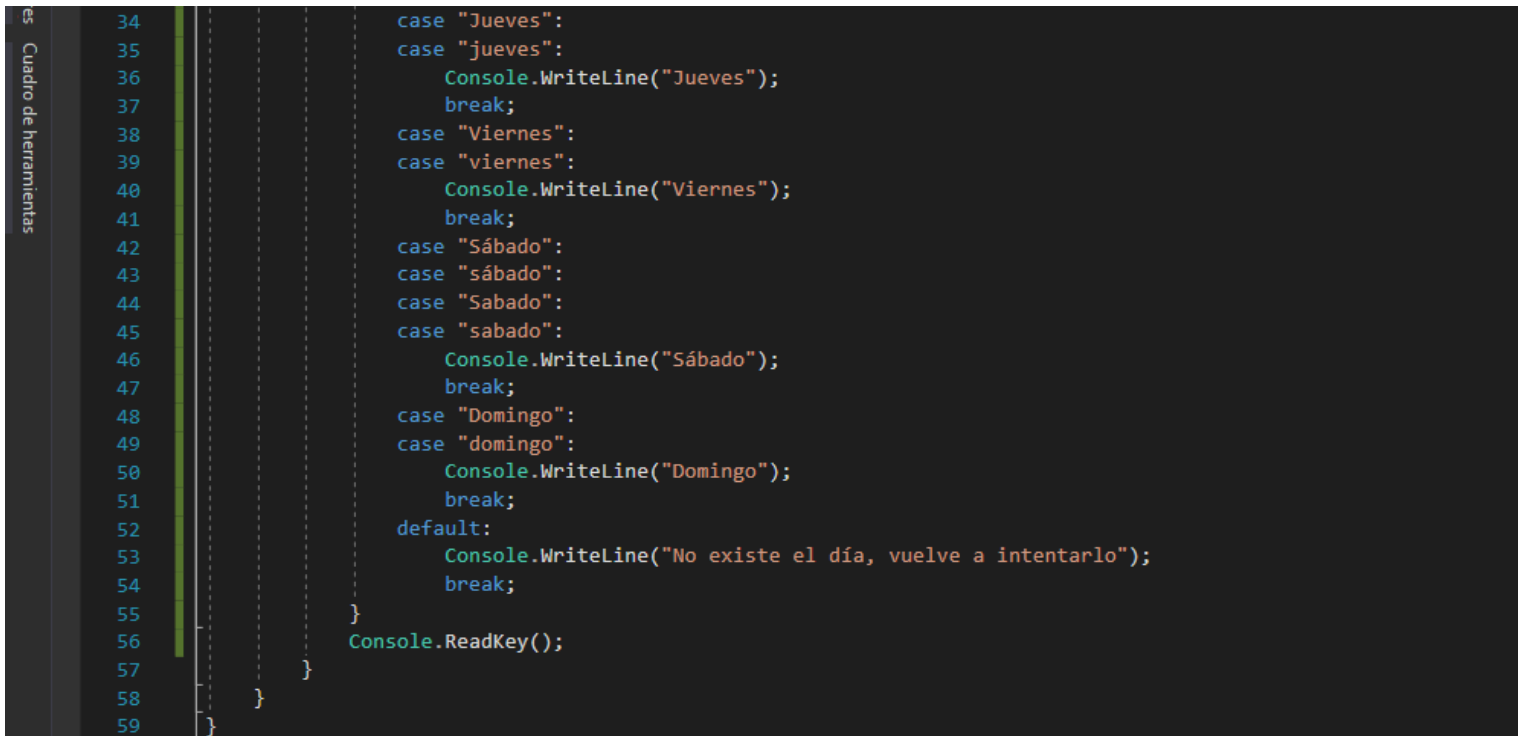
La variable de control del switch, puede ser un entero, pero también un carácter, como lo podemos ver a continuación:

```
Program.cs* X
Semana Semana.Program Main(string[] args)
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Semana
8 {
9     class Program
10    {
11        static void Main(string[] args)
12        {
13            char dia;
14            Console.WriteLine("Programa que indica el día de la semana\n");
15            Console.WriteLine("Ingresa la inicial del día de la semana" +
16                "\nL)Lunes\nM)Martes\nR)Miércoles\nJ)Jueves\nV)Viernes\nS)Sábado\nD)Domingo");
17            dia = Convert.ToChar(Console.ReadLine());
18            switch (dia)
19            {
20                case 'L': //En caso de que el usuario escriba la inicial con mayúsculas o
21                case 'l': //minúsculas, programamos ambas situaciones.
22                    Console.WriteLine("Lunes");
23                    break;
24                case 'M':
25                case 'm':
26                    Console.WriteLine("Martes");
27                    break;
28                case 'R':
29                case 'r':
30                    Console.WriteLine("Miércoles");
31                    break;
32                case 'J':
33                case 'j':
34                    Console.WriteLine("Jueves");
35                    break;
36                case 'V':
37                case 'v':
38                    Console.WriteLine("Viernes");
39                    break;
40                case 'S':
41                case 's':
42                    Console.WriteLine("Sábado");
43                    break;
44                case 'D':
45                case 'd':
46                    Console.WriteLine("Domingo");
47                    break;
48                default:
49                    Console.WriteLine("No existe el día, vuelve a intentarlo");
50                    break;
51            }
52            Console.ReadKey();
53        }
54    }
55 }
```

Por último, el switch también permite el uso de cadenas (String), para las diferentes opciones, veamos el siguiente ejemplo:



```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Semana
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             String dia;
14             Console.WriteLine("Programa que indica el día de la semana\n");
15             Console.WriteLine("Escribe el día de la semana" +
16                 "\n Lunes\n Martes\n Miércoles\n Jueves\n Viernes\n Sábado\n Domingo\n");
17             dia = Console.ReadLine();
18             switch (dia)
19             {
20                 case "Lunes"://En caso de que el usuario escriba la inicial con mayúsculas o
21                 case "lunes"://minúsculas, programamos ambas situaciones.
22                     Console.WriteLine("Lunes");
23                     break;
24                 case "Martes":
25                 case "martes":
26                     Console.WriteLine("Martes");
27                     break;
28                 case "Miércoles":
29                 case "miércoles":
30                 case "Miercoles":
31                 case "miercoles":
32                     Console.WriteLine("Miércoles");
33                     break;
```



```
34         case "Jueves":
35         case "jueves":
36             Console.WriteLine("Jueves");
37             break;
38         case "Viernes":
39         case "viernes":
40             Console.WriteLine("Viernes");
41             break;
42         case "Sábado":
43         case "sábado":
44         case "Sabado":
45         case "sabado":
46             Console.WriteLine("Sábado");
47             break;
48         case "Domingo":
49         case "domingo":
50             Console.WriteLine("Domingo");
51             break;
52         default:
53             Console.WriteLine("No existe el día, vuelve a intentarlo");
54             break;
55     }
56     Console.ReadKey();
57 }
58 }
59 }
```

EJERCICIO PROPUESTO: Realizar una calculadora con las 4 operaciones básicas. Nota: no se puede dividir entre 0 y solo existen 4 operaciones.

ciclos

- while
- do..while
- for

Funcionan siempre y cuando, la condición que evalúa, sea verdadera.

- El ciclo do..while se ejecuta cuando menos una vez, si la condición es verdadera se vuelve a ejecutar.
- Por el contrario, while y el ciclo for evalúan primero la condición, en caso de ser falsa, los ciclos nunca se ejecutan.

Para que la condición vaya cambiando, incluimos un contador dentro del código del ciclo, el cual avanza cada vez que se ejecuta el ciclo. Y dentro de la condición se evalúa a la variable del contador.

- La fórmula del contador es $C++$; ó $C=C+1$;

Además, utilizamos a los acumuladores, como su nombre lo dice para acumular valores que se van generando cada vez que se ejecuta el ciclo, como por ejemplo las ventas de los diferentes productos que se venden.

La fórmula del contador es: $acumVenta+=venta$; ó $acumVenta=acumVenta+venta$;

Ciclo For

Esta estructura nos permite repetir partes de un programa una y otra vez, mientras la condición del ciclo sea verdadera

La sintaxis del Ciclo es:

The diagram illustrates the syntax of a for loop: `for (int i=0; i<=10; i++)`. Three arrows point to specific parts of the code with labels:

- An arrow points from the label **Inicialización de Variable** to the `i=0` part of the code.
- An arrow points from the label **Limite de Variable** to the `i<=10` part of the code.
- An arrow points from the label **Incrementador** to the `i++` part of the code.

El ciclo for

está conformado por tres partes:

- Inicio del ciclo, desde donde comienza.
- Condición, límite del ciclo, el cual se repite mientras la condición sea verdadera.
- Incremento o decremento, el cual puede ser de 1 en 1 o de x en x, de forma positiva o negativa.

Programa que imprime los números comprendidos del 1 al 10:

```
Program.cs*  X
For1 For1.Program

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace For1
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("Imprime números del 1 al 10");
14             for(int i = 0; i <= 10; i++)
15             {
16                 Console.WriteLine("i= " + i);
17             }
18             Console.ReadKey();
19         }
20     }
21 }
```

Programa que imprime la numeración del 0 al 25, de 2 en 2:

```
Program.cs  X
For1 For1.Program Main(string[] args)

1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace For1
8  {
9      class Program
10     {
11         static void Main(string[] args)
12         {
13             Console.WriteLine("Imprime números del 1 al 25, de 2 en 2");
14             for(int i = 0; i <= 25; i+=2)
15             {
16                 Console.WriteLine("i= " + i);
17             }
18             Console.ReadKey();
19         }
20     }
21 }
```

Imprime de 10 en 10 partiendo del 100, llegando al -100, es decir, corre de manera negativa:

```
Program.cs
For1
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace For1
{
    class Program
    {
        static void Main(string[] args)
        {
            Console.WriteLine("Imprime números del 100 al -100, de 10 en 10");
            for(int i = 100; i >= -100; i-=10)
            {
                Console.WriteLine("i= " + i);
            }
            Console.ReadKey();
        }
    }
}
```

Realizar un programa que imprima la tabla de multiplicar deseada

```
Program.cs
For1
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

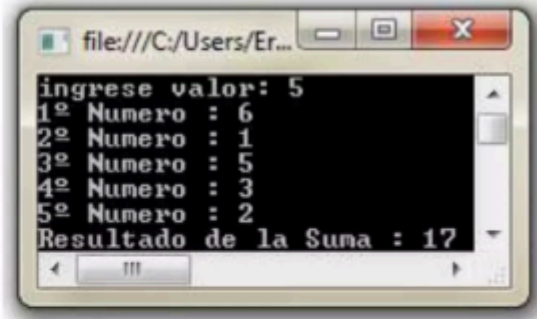
namespace For1
{
    class Program
    {
        static void Main(string[] args)
        {
            int tabla;
            Console.WriteLine("Que tabla de multiplicar deseas imprimir?");

            tabla = Convert.ToInt16(Console.ReadLine());
            for (int i = 1; i <= 10; i++)
            {
                Console.WriteLine(tabla+" x " + i+" = "+ tabla*i);
            }
            Console.ReadKey();
        }
    }
}
```

5. Aplicando for halla la suma de n numeros.

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

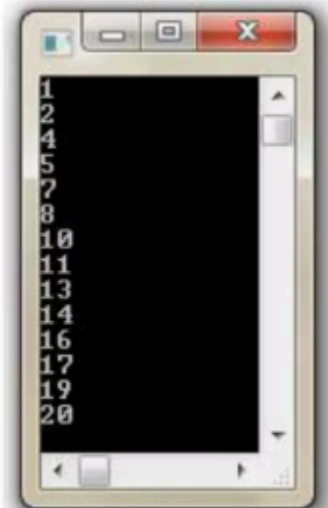
namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int i, n, suma, sumal;
            sumal = 0;
            Console.Write("ingrese valor:");
            n = int.Parse(Console.ReadLine());
            for (i = 1; i <= n; i++)
            {
                Console.Write("{0}° Numero :", i);
                suma = int.Parse(Console.ReadLine());
                sumal = suma + sumal;
            }
            Console.WriteLine("Resultado de la Suma : {0}", sumal);
            Console.ReadLine();
        }
    }
}
```



19. Mostrar los números del 1 al 20, en excepción de los múltiplos de tres

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            for (int i = 1; i <= 20; i++)
            {
                if (i % 3 == 0) continue;
                Console.WriteLine(i);
            }
            Console.ReadLine();
        }
    }
}
```



Ejercicio:

- Leer 7 números y hallar el mayor de ellos.
- Leer 10 números y decir cuántos son positivos, cuantos son negativos y cuantos ceros hay.
- Ingresar dos números y a partir del primer numero debe mostrar sucesivamente hasta llegar al segundo número. Nota: el primer numero debe ser menor que el segundo número.
- Realizar la serie de Fibonacci.
- Realizar el promedio de n estudiantes con n notas en la materia de algoritmos.

ciclo do..while:

1. Ingresar del Numero 1 al 10

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            i = 1;
            while (i <= 10)
            {
                Console.WriteLine("{0}", i);
                i = i + 1;
            }
            Console.ReadLine();
        }
    }
}
```

2. Ingresar del 1 al 10

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int i;
            i = 1;
            do
            {
                Console.WriteLine("{0}", i);
                i = i + 1;
            } while (i <= 10);
            Console.ReadLine();
        }
    }
}
```

7. Leer varios números, hasta que el usuario ingrese CERO. Muestre Cuantos números ingresó y su promedio.

```
using System;
using System.Collections.Generic;
using System.Text;

namespace ConsoleApplication1
{
    class Program
    {
        static void Main(string[] args)
        {
            int n, cont;
            double suma, prom;
            suma = 0; cont = 0;
            do
            {
                Console.Write("ingrese un numero:");
                n = int.Parse(Console.ReadLine());
                cont = cont + 1;
                suma = suma + n;
            } while (n != 0);

            prom = suma / (cont - 1);
            Console.WriteLine("cantidad de numeros: {0}", cont - 1);
            Console.WriteLine("promedio: {0}", prom);
            Console.ReadLine();
        }
    }
}
```

Ingrese número: 7
Ingrese número: 2
Ingrese número: 6
Ingresé número: 0
Cantidad de números:
3

Verificacion de una clave.

```
byte K, SW;
string CLAVE;
SW = 0; K = 0;
do
{
    Console.Write("DIGITE CLAVE: "); CLAVE = Console.ReadLine();
    if ((CLAVE.ToUpper() == "ARIEL"))
    {
        SW = 1;
    }
    else
    {
        K++;
    }
} while (((K < 3) & (SW == 0)));
if (SW == 1)
{
    Console.WriteLine("BIENVENIDO " + CLAVE);
}
else
{
    Console.WriteLine("OPORTUNIDADES TERMINADAS");
}
Console.Write("Pulse una Tecla:"); Console.ReadLine();
```

Ejercicio: Leer un numero y decir si es primo o no. El usuario decide cuando terminar.

Realizar el factorial de un numero x.

While

Tabla de multiplicar

```
static void Main(string[] args)
{
    int NUM, I, RESULT;
    string linea;
    Console.Write("DIGITE NÚMERO:"); linea = Console.ReadLine();
    NUM = int.Parse(linea);
    I = 1;
    while ((I <= 12))
    {
        RESULT = NUM * I;
        Console.WriteLine("{0} * {1} = {2}", NUM, I, RESULT);
        I = I + 1;
    }
    Console.Write("Pulse una Tecla:"); Console.ReadLine();
}
```

Numero primo:

```
static void Main(string[] args)
{
    int n, x, sw, resi;
    string linea;
    x = 2;
    sw = 0;
    Console.WriteLine("NUMERO PRIMO");
    Console.WriteLine();
    Console.Write("Ingrese el numero:");
    linea = Console.ReadLine();
    n = int.Parse(linea);
    while (x < n && sw == 0)
    {
        resi = n % x;
        if (resi == 0)
        {
            sw = 1;
        }
        else
        {
            x = x + 1;
        }
    }
    if (sw == 0)
    {
        Console.WriteLine();
        Console.WriteLine("El numero es PRIMO");
    }
    else
    {
        Console.WriteLine();
        Console.WriteLine("El numero no es PRIMO");
    }
    Console.ReadKey();
}
```

Arreglos:

También se le conoce como Matriz, la cual almacena una colección de datos del mismo tipo (int, float, double, String, etc.), con ello se evita declarar una infinidad de variables, del mismo tipo.

Una sola variable que puede almacenar muchas variables del mismo tipo, como por ejemplo, una agenda que almacena en el campo teléfono, muchos teléfonos en una sola variable.

Por regla, toda matriz comienza en la posición 0 y llega hasta n, accedemos a sus valores por medio del índice que se coloca en la posición del valor referido. Todas las matrices consisten en ubicaciones de memoria contiguas.

Para declarar una matriz seguimos la siguiente sintaxis:

```
datatype[] arrayName;
```

datatype: Se utiliza para especificar el tipo de elemento de la matriz.

[]: Especifica el rango de la matriz, es decir, su tamaño.

arrayName: Nombre del arreglo.

Es necesario inicializar a la matriz para crear una instancia a la matriz y poder asignar valores a la misma. Para crear la instancia de la matriz utilizamos la palabra reservada new.

```
double[] balance = new double[10];
```

Para asignar valores a la matriz podemos, asignar a elementos de matriz individuales, utilizando el número de índice.

```
double[] balance = new double[10];  
balance[0] = 4500.0;
```

Podemos asignar valores a la matriz en el momento de la declaración:

```
double[] balance = { 2340.0, 4523.69, 3421.0};
```

También podemos inicializar una matriz de la siguiente manera:

```
int [] marks = new int[5] { 99, 98, 92, 97, 95};
```

Se puede omitir el tamaño de la matriz:

```
int [] marks = new int[] { 99, 98, 92, 97, 95};
```

Es posible copiar los valores de una matriz origen a una matriz destino, ambas apuntan a la misma dirección de memoria:

```
int [] marks = new int[] { 99, 98, 92, 97, 95};  
int[] score = marks;
```

En caso de no inicializar la matriz, el valor que contiene es 0.

Ejercicio

Confeccionar un programa que permita obtener la suma de n números agregados en un vector. Nota. Si cumple con lo solicitado?

```
using System;  
public class Ejemplo36  
{  
    public static void Main()  
    {  
        int[] numero = new int[5]; /* Un array de 5 números enteros */  
        int suma; /* Un entero que será la suma */  
        int i; /* Para recorrer los elementos */  
        for (i = 0; i <= 4; i++) /* Pedimos los datos */  
        {  
            Console.Write("Introduce el dato numero {0}: ", i + 1);  
            numero[i] = Convert.ToInt32(Console.ReadLine());  
        }  
        suma = 0; /* Valor inicial de la suma */  
        for (i = 0; i <= 4; i++) /* Y hallamos la suma repetitiva */  
        {  
            suma += numero[i];  
            Console.WriteLine("Su suma es {0}", suma);  
        }  
        Console.ReadKey();  
    }  
}
```


Ejercicio

Desarrollar un programa que permita cargar 5 nombres de personas y sus edades respectivas. Luego de realizar la carga por teclado de todos los datos imprimir los nombres de las personas mayores de edad (mayores o iguales a 18 años)

```
namespace PruebaVector10
{
    class PruebaVector10
    {
        private string[] nombres;
        private int[] edades;

        public void Cargar()
        {
            nombres = new string[5];
            edades = new int[5];
            for (int f = 0; f < nombres.Length; f++)
            {
                Console.Write("Ingrese nombre:");
                nombres[f] = Console.ReadLine();
                Console.Write("Ingrese edad:");
                string linea;
                linea = Console.ReadLine();
                edades[f] = int.Parse(linea);
            }
        }

        public void MayoresEdad()
        {
            Console.WriteLine("Personas mayores de edad.");
            for (int f = 0; f < nombres.Length; f++)
            {
                if (edades[f] >= 18)
                {
                    Console.WriteLine(nombres[f]);
                }
            }
            Console.ReadKey();
        }

        static void Main(string[] args)
        {
            PruebaVector10 pv = new PruebaVector10();
            pv.Cargar();
            pv.MayoresEdad();
        }
    }
}
```

Ejercicio propuesto: Confeccionar un programa que permita cargar los nombres de 5 operarios y sus sueldos respectivos. Mostrar el sueldo mayor y el nombre del operario