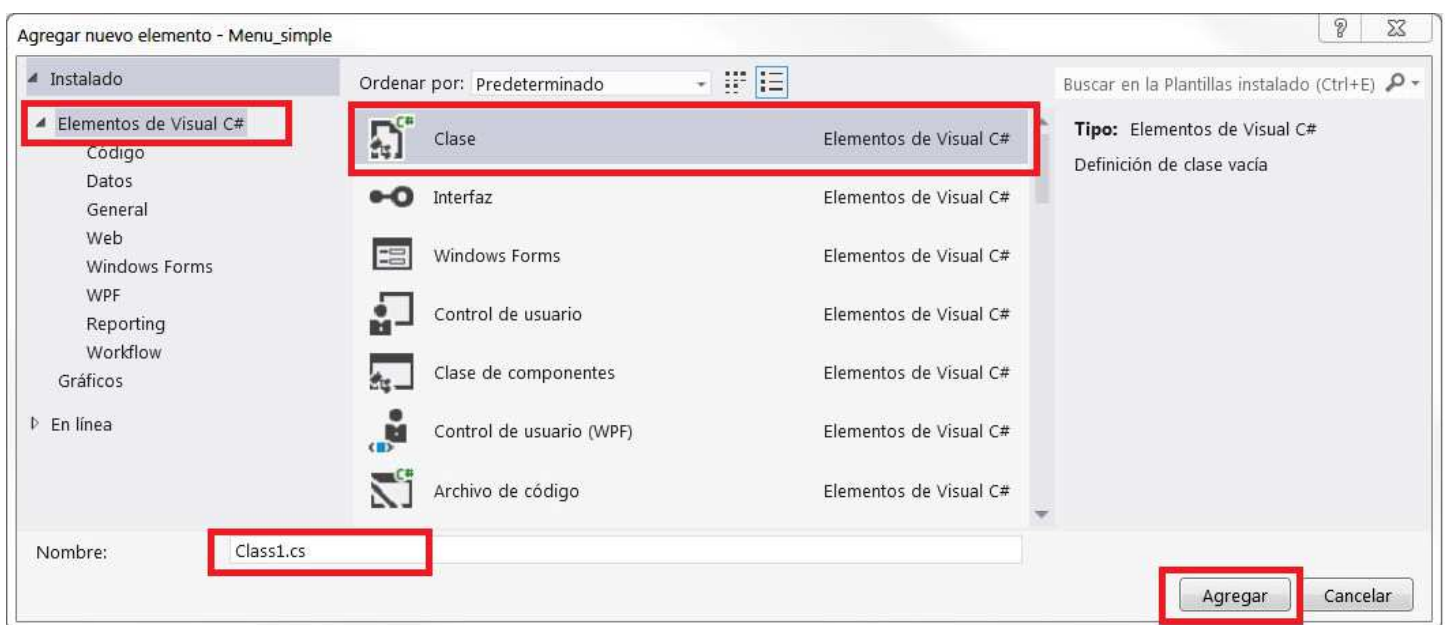


Menú Simple llamando a otras clases en C#

1. Cree un nuevo proyecto por consola y crear un método Menú en Program.cs
2. Una vez creado el proyecto Agregar las clases necesarias para el menú, para ello:



3. Se crea el método Menú en Program.cs e instanciar las clases a usar

```
1 using System;
2 using System.Collections.Generic;
3 using System.Linq;
4 using System.Text;
5 using System.Threading.Tasks;
6
7 namespace Menu_Simple
8 {
9     class Program
10     {
11         static void Main(string[] args)
12         {
13             Menu(); // llama a Menu
14         }
15
16         static void Menu() {
17             Class1 cla1 = new Class1();
18             Class2 cla2 = new Class2();
19             Class3 cla3 = new Class3();
20         }
21     }
22 }
```

4. En el código de la Clase 1 (Class1.cs), se crea el método público Mostrar:

```
namespace Menu_Simple
{
    class Class1
    {
        public void Mostrar() {
            Console.WriteLine("Opción 1 del Menú Principal");
            Console.Read();
        }
    }
}
```

Repetir en cada Clase creada

5. Se crea el método Menú de forma completa en Program.cs

```
static void Menu() {
    Class1 cla1 = new Class1();
    Class2 cla2 = new Class2();
    Class3 cla3 = new Class3();

    int valor;
    bool esValido = false;
    Console.WriteLine("1. Tarea 1");
    Console.WriteLine("2. Tarea 2");
    Console.WriteLine("3. Tarea 3");
    Console.WriteLine("4. Salir");
    Console.WriteLine("Seleccione una opción [1,2,3,4]: ");
    esValido = int.TryParse(Console.ReadLine(), out valor);

    while (!esValido) {
        Console.Clear();
        Console.WriteLine("\n Seleccione una opción [1,2,3,4]: ");
        esValido = int.TryParse(Console.ReadLine(), out valor);
        // tryparse automáticamente devuelve un valor booleano (true, false)
    }

    switch (valor)
    {
        case 1:
            cla1.Mostrar();
            Menu();
            break;
        case 2:
            cla2.Mostrar();
            Menu();
            break;
        case 3:
            cla3.Mostrar();
            Menu();
            break;
        default:
            Console.WriteLine("Ingresa una Opción");
            break;
    }
}
```

Nota: De ahora en adelante debe Crear un menú simple solo para las tareas y otro para los ejercicios, en cada uno de ellos debe crear una clase por cada Tarea y/o ejercicio.

Instrucción While

Una estructura repetitiva permite ejecutar una instrucción o un conjunto de instrucciones varias veces. Una ejecución repetitiva de sentencias se caracteriza por:

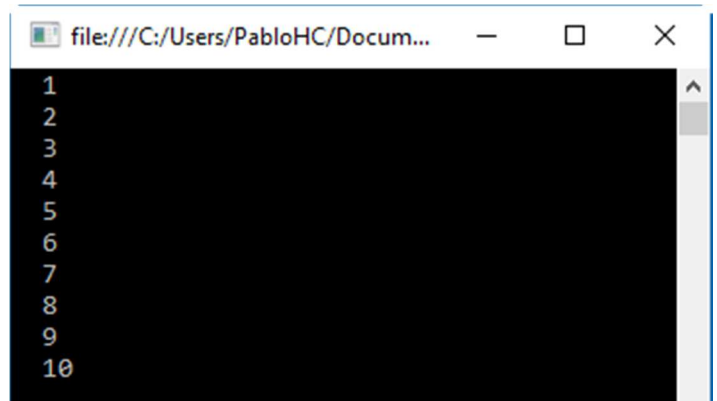
- La(s) sentencia(s) que se repiten.
- La condición antes de cada repetición, que motivará que se repitan o no las sentencias.

```
while (<condición>)  
{  
    <instrucciones>  
}
```

Ejercicio 8: Decremento de 10 a 1

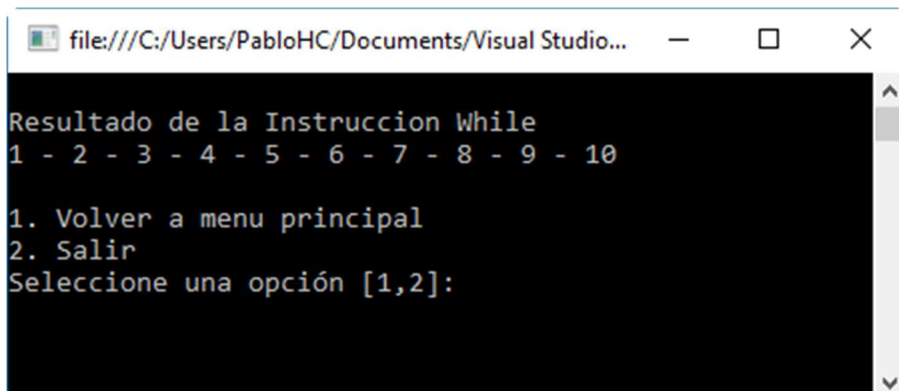
En una clase de su menú, realice la siguiente instrucción:

```
int x;  
x = 1;  
while (x <= 10)  
{  
    Console.WriteLine(x);  
    x = x + 1;  
}  
Console.Read();
```



```
file:///C:/Users/PabloHC/Docum...  
1  
2  
3  
4  
5  
6  
7  
8  
9  
10
```

Tarea 5 Modifique el código del Ejercicio 8 para que la salida sea vista así:



```
file:///C:/Users/PabloHC/Documents/Visual Studio...  
Resultado de la Instruccion While  
1 - 2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10  
  
1. Volver a menu principal  
2. Salir  
Seleccione una opción [1,2]:
```

Para salir directamente de la consola puede usar

```
Environment.Exit(0);
```

Ejercicio 9: Números mayores o menores que cero

```
int numero;
Console.WriteLine("Ingresa un Número (0 para salir): ");
numero = Convert.ToInt32(Console.ReadLine());

while (numero != 0)
{
    if (numero > 0) Console.WriteLine("Es positivo");
    else Console.WriteLine("Es negativo");

    Console.WriteLine("Teclea otro número (0 para salir): ");
    numero = Convert.ToInt32(Console.ReadLine());
}
```

Ejercicio 10 Ingreso de Números y su promedio

```
int i = 0;
float num, acum = 0;

while (i < 5)
{
    Console.Write("Ingrese {0}º número:", i + 1);
    num = float.Parse(Console.ReadLine());
    acum += num;
    i++;
}
Console.Write("El promedio de los 5 números es {0}", acum / 5);
Console.Read();
```

Ejercicio 11 Ingreso de Números e indicar cuantos son > 0 y < 0

```
float numero;
uint cont = 0, menor = 0, mayor = 0;

while (cont < 10)
{
    Console.Write("Ingresa número:");
    numero = float.Parse(Console.ReadLine());

    if (numero > 0)
        mayor++;
    if (numero < 0)
        menor++;

    cont++;
}
Console.WriteLine("De 10 números son ... \n Menores a 0 = {0} y Mayores a 0 = {1}", menor, mayor);
Console.ReadLine();
```

Instrucción Do While

```
do {  
    <instrucciones>  
} while(<condición>);
```

La única diferencia del significado de **do...while** respecto al de **while** es que en vez de evaluar primero la condición y ejecutar <instrucciones> sólo si es cierta, **do...while** primero ejecuta las <instrucciones> y luego mira la <condición> para ver si se ha de repetir la ejecución de las mismas

Ejercicio 12 Números entre 0 y 999

```
int valor;  
string linea;  
do  
{  
    Console.Write("Ingrese un valor entre 0 y 999 (0 finaliza):");  
    linea = Console.ReadLine();  
    valor = int.Parse(linea);  
    if (valor >= 100)  
    {  
        Console.WriteLine("Tiene 3 dígitos.");  
    }  
    else  
    {  
        if (valor >= 10)  
        {  
            Console.WriteLine("Tiene 2 dígitos.");  
        }  
        else  
        {  
            Console.WriteLine("Tiene 1 dígito.");  
        }  
    }  
} while (valor != 0);
```

Tarea 6 Realice un sistema de ingreso simple, declare una variable de tipo string, y solicite por teclado dicha contraseña hasta que sea la correcta

Ayudita:

```
do  
{  
    Leer palabra introducida  
    Condición if (si la clave i= a la introducida);  
}  
while (clave != a la introducida);  
Mostrar un OK
```

Ciclo For

La estructura for se usa en aquellas situaciones en las cuales CONOCEMOS la cantidad de veces que queremos que se ejecute el bloque de instrucciones. Ejemplo: cargar 10 números, ingresar 5 notas de alumnos, etc.

```
for (<inicialización>; <condición>; <modificación>){  
    <instrucciones>  
}
```

Ejercicio 13 la siguiente instrucción, cuenta del 1 al 10

```
int suma, i, valor, promedio;  
string linea;  
suma = 0;  
for (i = 1; i <= 5; i++)  
{  
    Console.Write("Ingrese valor:");  
    linea = Console.ReadLine();  
    valor = int.Parse(linea);  
    suma = suma + valor;  
}  
Console.Write("La suma es:");  
Console.WriteLine(suma);  
promedio = suma / 5;  
Console.Write("El promedio es:");  
Console.Write(promedio);  
Console.ReadKey();
```

Ejercicio 14 Tabla de multiplicar de números entre 1 y 10

```
int num;  
do  
{  
    Console.Write("Ingresa un numero > 0 y <= 10:");  
    num = Int32.Parse(Console.ReadLine());  
  
} while (num < 1 || num > 10);  
  
for (int i = 1; i <= 10; i++)  
    Console.WriteLine("{0} x {1} = {2}", num, i, num * i);  
Console.Read();
```

Tarea 7 Ingrese por teclado 2 números para multiplicarlos, si el primer número ingresado es igual a cero, deberá mostrar por pantalla "Error la multiplicación x 0 = 0"; si el numero ingresado es distinto de cero debe pedir el siguiente número.

Sentencia break: termina el bucle

Podemos salir de un bucle "for" antes de tiempo con la orden "break", ejemplo:

```
int contador;  
  
for (contador=1; contador<=10; contador++)  
{  
    if (contador==5) break;  
    Console.Write("{0} ", contador);  
}
```

El resultado seria: 1 2 3 4

en cuanto se llega al valor 5, se interrumpe el "for", por lo que no se alcanza el valor 10

Sentencia continue: fuerza la siguiente iteración

Podemos saltar alguna repetición de un bucle con la orden "continue"

```
int contador;  
  
for (contador=1; contador<=10; contador++)  
{  
    if (contador==5) continue;  
    Console.Write("{0} ", contador);  
}
```

El resultado seria:

1 2 3 4 6 7 8 9 10

Se observar que no aparece el valor 5.

Sentencia goto, para hacer saltos incondicionales

Ejercicio 15 Realice el siguiente ejemplo y analise lo que hace

```
int i, j, c;  
Console.Write("ingrese un numero para el ciclo:");  
j= Convert.ToInt32(Console.ReadLine());  
Console.Write("ingrese un numero para el corte:");  
c = Convert.ToInt32(Console.ReadLine());  
  
for (i = 0; i <= j; i++)  
{  
    Console.WriteLine("numeros: {0}", i);  
    if (i == c) goto salida;  
}  
salida:  
    Console.Write("se corto en {0}, Fin del programa",c);  
    Console.ReadKey();
```

Ejercicio 16 (Mejorando el ejercicio 15).

Ingrese un numero de entrada y un numero de corte, si el numero de entrada es menor debe permitir volver a ingresar el numero, si el numero de entrada es mayor, debe mostrar los numeros correlativos y mostrar en que numero se corto el ciclo.

inicio:

```
int i, j, c;
Console.Clear();
Console.Write("ingrese un numero para el ciclo:");
j = Convert.ToInt32(Console.ReadLine());
Console.Write("ingrese un numero menor para el corte:");
c = Convert.ToInt32(Console.ReadLine());

if (j <= c)
{
    Console.WriteLine("Error su numero ingresado es menor");
    Console.ReadKey();
    goto inicio;
}
else
{
    for (i = 0; i <= j; i++)
    {
        Console.WriteLine("numeros: {0}", i);
        if (i == c) goto salida;
    }
}
salida:
    Console.Write("se corto en {0}, Fin del programa", c);
    Console.ReadKey();
}
```

Tarea 8 Solicite por teclado una contraseña (de tipo String) hasta que esta sea la correcta, la contraseña correcta debe ser: qwerty

Una vez ingresada la contraseña correcta, calcule el factorial de un número ingresado por teclado. Considere lo siguiente:

Un factorial es el producto de un número y sus antecesores empezando desde 1. Y se le reconoce por el símbolo "n!". Ejemplos

el factorial de 5

$5! = 5 \times 4 \times 3 \times 2 \times 1$

osea:

$5! = 120$

el factorial de 4

$4! = 4 \times 3 \times 2 \times 1$

$4! = 24$

Por ende ingrese un número por teclado, calcule su factorial y además divida ese factorial por el mismo número que ingreso, o sea si ingreso el 5, entonces $5! = 120$ y la división $120/5 = 24$