

Seminario de Lenguajes (.NET)

Práctica 1

1. Consultar en la documentación de Microsoft y responder cuál es la diferencia entre los métodos `WriteLine()` y `Write()` de la clase `System.Console` ¿Cómo funciona el método `ReadKey()` de la misma clase? Escribir un programa que imprima en la consola la frase “Hola Mundo” haciendo una pausa entre palabra y palabra esperando a que el usuario presione una tecla para continuar **Tip: usar los métodos `ReadKey()` y `Write()` de la clase `System.Console`.**
2. Investigar por las secuencias de escape `\n`, `\t`, `\"` y `\\`. Escribir un programa que las utilice para imprimir distintos mensajes en la consola.
3. El carácter `@` delante de un `string` desactiva los códigos de escape. Probar el siguiente fragmento de código, eliminar el carácter `@` y utilizar las secuencias de escape necesarias para que el programa siga funcionando de igual manera

```
string st = @"c:\windows\system";  
Console.WriteLine(st);
```

4. Escribir un programa que solicite al usuario ingresar su nombre e imprima en la consola un saludo personalizado utilizando ese nombre o la frase “Hola mundo” si el usuario ingresó una línea en blanco. Para ingresar un `string` desde el teclado utilizar `Console.ReadLine()`
5. Idem. al ejercicio anterior salvo que se imprimirá un mensaje de saludo diferente según sea el nombre ingresado por el usuario. Así para “Juan” debe imprimir “¡Hola amigo!”, para “María” debe imprimir “Buen día señora”, para “Alberto” debe imprimir “Hola Alberto”. En otro caso, debe imprimir “Buen día ” seguido del nombre ingresado o “¡Buen día mundo!” si se ha ingresado una línea vacía.
 - a) utilizando `if ... else if`
 - b) utilizando `switch`
6. Utilizar `Console.ReadLine()` para leer líneas de texto (secuencia de caracteres que finaliza al presionar `<ENTER>`) por la consola. Por cada línea leída se debe imprimir la cantidad de caracteres de la misma. El programa termina al ingresar la cadena vacía. (si `st` es una variable de tipo `string`, entonces `st.Length` devuelve la cantidad de caracteres del `string`).
7. Qué hace la instrucción `Console.WriteLine("100".Length);` ?
8. Sea `st` una variable de tipo `string` correctamente declarada. ¿Es válida la siguiente instrucción: `Console.WriteLine(st=Console.ReadLine());`?
9. Escribir un programa que lea dos palabras separadas por un blanco que terminan con `<ENTER>`, y determinar si son simétricas (Ej: 'abbccd' y 'dccbba' son simétricas).
Tip: si `st` es un `string`, entonces `st[0]` devuelve el primer carácter de `st`, y `st[st.Length-1]` devuelve el último carácter de `st`.
10. Escribir un programa que imprima en la consola todos los números que sean múltiplos de 17 o de 29 comprendidos entre 1 y 1000
11. Comprobar el funcionamiento del siguiente fragmento de código, analizar el resultado y contestar las preguntas.

```

Console.WriteLine("10/3 = " + 10 / 3);
Console.WriteLine("10.0/3 = " + 10.0 / 3);
Console.WriteLine("10/3.0 = " + 10 / 3.0);
int a = 10, b = 3;
Console.WriteLine("Si a y b son variables enteras, si a=10 y b=3");
Console.WriteLine("entonces a/b = " + a / b);
double c = 3;
Console.WriteLine("Si c es una variable double, c=3");
Console.WriteLine("entonces a/c = " + a / c);

```

¿Qué se puede concluir respecto del operador de división “/” ?

¿Cómo funciona el operador + entre un **string** y un dato numérico?

12. Escribir un programa que imprima todos los divisores de un número entero ingresado desde la consola. Para obtener el entero desde un **string st** utilizar **int.Parse(st)**.
13. Escribir un programa que calcule la suma de dos números reales introducidos por teclado y muestre el resultado en la consola (utilizar **double.Parse(st)** para obtener el valor **double** a partir del **string st**).
14. Escribir un programa que multiplique por 365 el número entero ingresado por el usuario desde la consola. El resultado debe imprimirse en la consola dígito por dígito, separado por un espacio comenzando por el dígito menos significativo al más significativo.
15. Escribir un programa que solicite un año por pantalla y diga si es bisiesto. Un año es bisiesto si es divisible por 4 pero no por 100. Si es divisible por 100, para ser bisiesto debe ser divisible por 400.
16. Si **a** y **b** son variables enteras, identificar el problema (y la forma de resolverlo) de la siguiente expresión (tip: observar qué pasa cuando **b = 0**):


```
if ((b != 0) & (a/b > 5)) Console.WriteLine(a/b);
```
17. Utilizar el operador ternario condicional para establecer el contenido de una variable entera con el menor valor de otras dos variables enteras.
18. ¿Cuál es la salida por consola del siguiente código?

```

for (int i = 0; i <= 4; i++)
{
    string st = i < 3 ? i == 2 ? "dos" : i == 1 ? "uno" : "< 1" : i < 4 ? "tres" : "> 3";
    Console.WriteLine(st);
}

```

19. Para cada una de las siguientes líneas de código verificar cuáles son permitidas por el compilador y en tal caso en qué estado quedan las variables involucradas en la declaración al ejecutarse:

```
1.   int a, b, c;
2.   int a; int b; int c, d;
3.   int a = 1; int b = 2; int c = 3;
4.   int b; int c; int a = b = c = 1;
5.   int c; int a, b = c = 1;
6.   int c; int a = 2, b = c = 1;
7.   int a = 2, b, c, d = 2 * a;
8.   int a = 2, int b = 3, int c = 4;
9.   int a = 2; b = 3; c = 4;
10.  int a; int c = a;
11.  char c = 'A', string st = "Hola";
12.  char c = 'A'; string st = "Hola";
13.  char c = 'A', st = "Hola";
```

20. ¿Cuál es el problema del código siguiente y cómo se soluciona?

```
int i = 0;
for (int i = 1; i <= 10;)
{
    Console.WriteLine(i++);
}
```

21. Analizar el siguiente código. ¿Cuál es la salida por consola?

```
int i = 1;
if (--i == 0)
{
    Console.WriteLine("cero");
}
if (i++ == 0)
{
    Console.WriteLine("cero");
}
Console.WriteLine(i);
```