

Laboratorio # 5

Introducción

Objetivos



Entender el proceso de compilación y sus fases en C#.



Familiarizar con el IDE de Visual Studio.



Estandarizar la forma de envío de laboratorios.

Práctica en laboratorio

1.

Proyectos en Visual Studio

A.

Abra Visual Studio y cree un proyecto de tipo Visual C#, Console Application. El nombre del proyecto debe seguir la siguiente sintaxis ***L1+-<iniciales>+<carne>***. Guárdelo en el almacenamiento interno de la computadora.

B.

Agregue la sentencia:

i. Para mostrar el mensaje:

```
class Program
{
    0 references
    static void Main(string[] args)
    {
        Console.WriteLine("Hola Mundo soy NOMBRE");
    }
}
```

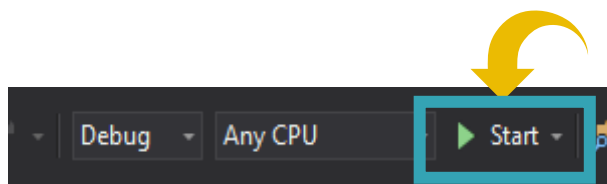
Nota: reemplace la palabra 'NOMBRE' con su nombre.

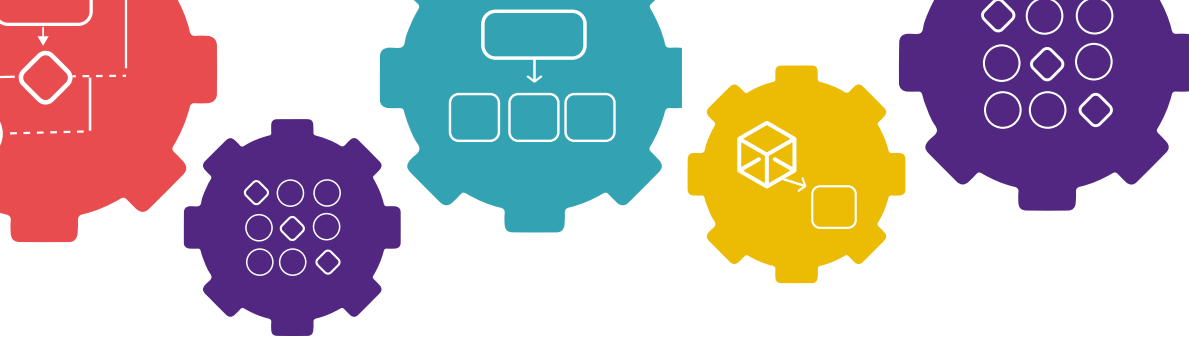
ii. Para que haga pausa en la ejecución:

```
class Program
{
    0 references
    static void Main(string[] args)
    {
        Console.WriteLine("Hola Mundo soy NOMBRE");
        Console.ReadKey();
    }
}
```

C.

Ejecute el programa presionando la tecla F5 o el *botón verde*:





D.

Agregue, ejecute y describa las diferencias entre las instrucciones:

WriteLine y *Write*

```
static void Main(string[] args)
{
    Console.WriteLine("Hola Mundo");
    Console.WriteLine("soy NOMBRE");

    Console.Write("Hola Mundo");
    Console.Write("soy NOMBRE");
    Console.ReadKey();
}
```

v

E.

Coloque las diferencias como comentarios entre los `/* */`:

```
static void Main(string[] args)
{
    Console.WriteLine("Hola Mundo");
    Console.WriteLine("soy NOMBRE");

    /* COMENTARIOS */

    Console.Write("Hola Mundo");
    Console.Write("soy NOMBRE");
    Console.ReadKey();
}
```

F.

Agregue la instrucción “Ingrese su nombre”, lea el texto ingresado y almacénelo en una variable:

```
static void Main(string[] args)
{
    Console.WriteLine("Ingrese su nombre: ");
    string Nombre = Console.ReadLine();

    Console.WriteLine("Hola Mundo");
    Console.WriteLine("soy " + Nombre);

    /* COMENTARIOS */

    Console.Write("Hola Mundo ");
    Console.Write("soy " + Nombre);
    Console.ReadKey();
}
```

G.

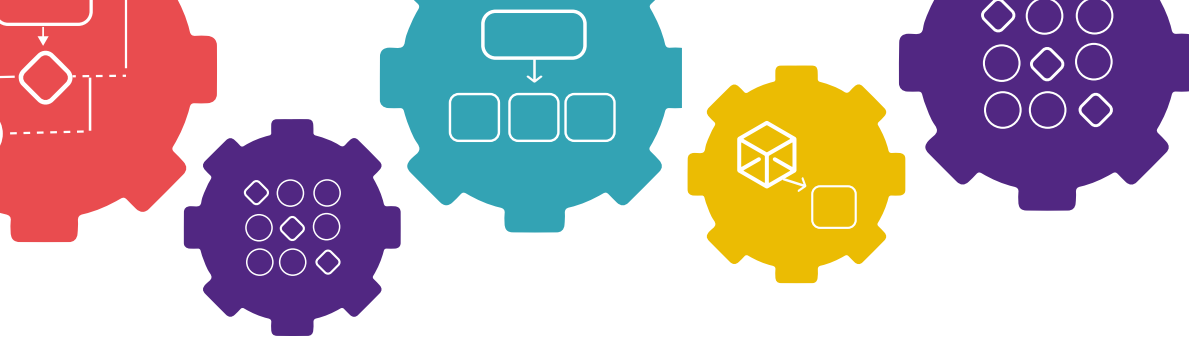
Ejecute el programa presionando la tecla F5 o el botón verde.

H.

Identifique la ubicación donde está guardado el proyecto y envíelo a una carpeta comprimida (**.zip**).

I.

Suba al portal la carpeta comprimida, en el espacio para entrega del laboratorio.



2.

Proceso de compilación

A.

El proyecto que se acaba de crear, en Visual Studio C# para Consola, ¿utiliza un compilador o un intérprete para ejecutarse? ¿Por qué?

B.

Para el proyecto que se acaba de crear, identifique:

ENTRADAS	
PROCESOS	
SALIDAS	



Tarea – Problema No. 1

- Abra Visual Studio y cree un proyecto de tipo Visual C#, Console Applications. El nombre del proyecto debe seguir la siguiente sintaxis **T1+-+<iniciales>+<carné>**.
- Agregue una instrucción que muestre en pantalla

“Mi segundo programa”

- Cree las variables ‘String’ para ingresar los siguientes datos:
 - **sNombre**
 - **sEdad**
 - **sCarrera**
 - **sCarné**
- Utilice la instrucción `Console.ReadLine()`; para solicitar al usuario que ingrese los datos mencionados en el punto C.
- En Pantalla se debe mostrar el siguiente mensaje con los datos ingresados por el usuario:

Mi segundo programa

Nombre: + ‘NOMBRE’

Edad: + ‘EDAD’

Carrera: + ‘CARRERA’

Carné: + ‘CARNE’

- Agregue las instrucciones necesarias para mostrar en pantalla el siguiente mensaje:

Soy + ‘NOMBRE’ +, tengo + ‘EDAD’ + años y estudio la carrera de + ‘CARRERA’ +.

Mi número de carné es; ‘CARNE’

- g. Recuerde que al finalizar debe colocar la instrucción de ***Console.ReadKey()***; para que permita hacer una pausa antes que se cierre el programa.
- h. Valide el funcionamiento del proyecto ejecutándolo (F5 o *botón verde*).
- i. Localice la ubicación donde está guardado el proyecto y enviarlo a una carpeta comprimida (.zip).
- j. Suba, al portal, la carpeta comprimida, en el espacio para entrega del laboratorio.



Instrucciones y componentes de C# utilizados en esta práctica:

1.

Consola o línea de comandos:

Es la ventana con fondo negro que te aparece al ejecutar el programa. Permite a los usuarios dar instrucciones a algún programa informático por medio de una línea de texto simple. Con ella puedes ingresar las entradas y mostrar las salidas de tu programa.

2.

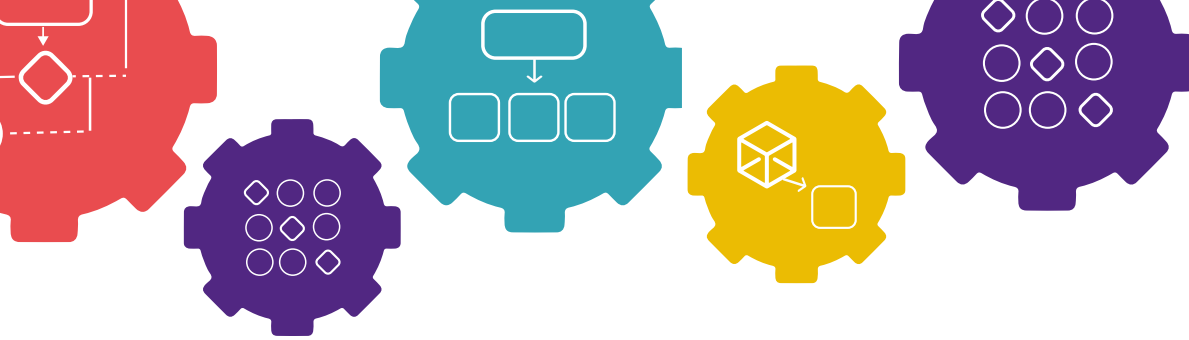
Sentencia **Console.WriteLine()** y **Console.Write()**

Es un método que permite mostrar (escribir) un texto en la consola.

3.

Sentencia **Console.ReadLine()**

Le permite al programa leer el texto que el usuario ha ingresado en la consola, para posteriormente almacenarlo en alguna variable.



4

Sentencia **Console.ReadKey()**

Esta sentencia hace que la consola permanezca abierta en espera de que se presione alguna tecla. Permite también capturar el carácter de dicha tecla al ser presionada.