



TECNOLÓGICO NACIONAL DE MÉXICO INSTITUTO TECNOLÓGICO DE TLAXIACO

CARRERA: INGENIERÍA EN SISTEMAS COMPUTACIONALES

TÓPICOS AVANZADOS DE PROGRAMACIÓN

**PRACTICA 6:
HILOS**

DOCENTE: ING. JOSÉ ALFREDO ROMAN CRUZ

ALUMNO:
Irving Fernando Reyes Pacheco

SEMESTRE: CUARTO

GRUPO: 4US

ENERO 2022- JULIO 2022

Heroica Ciudad de Tlaxiaco, Oaxaca, a 04 de abril del 2022.





Practica numero 5

Objetivo:

Realizar ejercicios sobre la utilización de hilos con C#, para comprender el funcionamiento y la forma en que se implementan dentro de un programa.

Materiales:

- Una computadora
- Internet
- Visual Studio.

Lista de figuras:

Ilustración 1
Ilustración 2
Ilustración 3
Ilustración 4
Ilustración 5
Ilustración 6
Ilustración 7
Ilustración 8
Ilustración 9
Ilustración 10
Ilustración 11
Ilustración 12
Ilustración 13
Ilustración 14
Ilustración 15
Ilustración 16
Ilustración 17
Ilustración 18
Ilustración 19
Ilustración 20



Procedimiento problema 1

Agregamos las librerías necesarias para la implementación del código con que se estará trabajando, las librerías permiten agregar más elementos a una clase, lo cual permite referenciar ciertas cualidades a nuestro programa.

```
using System;  
using System.Threading;
```

Ilustración 1

Nombre de nuestra clase, donde se implementará el código necesario para darle solución a cierto problema que sea el fin, del porque se realiza el código.

```
0 referencias  
class Hilos
```

Ilustración 2

Declaramos una variable de tipo booleana la cual tendrá un valor por defecto, el cual será verdadero.

```
private static bool ejecutar = true;
```

Ilustración 3

Creamos un método main el cual nos permite ingresar código el cual puede ejecutarse, este método puede decirse que es el método principal de la clase.

Además, inicializamos dos variables dándoles un tipo de dato entero, las cuales nos permitirán hacer referencia a los hilos que se ocuparan en el programa.

```
0 referencias  
static void Main(string[] args)  
{  
    int m = 0;  
    int x = 0;
```

Ilustración 4

Creamos un ciclo for para almacenar los valores de hilos, cada hilo tendrá almacenados otros 8 hilos dentro de él, dentro de este ciclo, mandamos a llamar al método 'MensajeM', dentro del cual se encuentran diferentes instrucciones que lleva a cabo el método.

```
for(x=0; x<8; x++) {  
    Thread miHilo = new Thread(MensajeM);  
    miHilo.Start();  
}
```

Ilustración 5

Creamos un while dentro del cual crearemos una instrucción la cual nos permitirá mostrar un texto, el cual por la instrucción que se le está dando se mostrará de color blanco, además agregamos una condición, la cual permitirá que se imprima este mensaje cincuenta veces, después de que ese número se supere, el valor de la variable ejecutar cambiará a falso y dejará de mostrar valores.

```
while (ejecutar)
{
    Console.ForegroundColor = ConsoleColor.White;
    Console.WriteLine("Saludos desde el hiloprincipal {0}", m);
    m++;
    if (m == 50)
        ejecutar = false;
}
```

Ilustración 6

Creamos el método MensajeM, dentro del cual agregaremos un conjunto de instrucciones, las cuales pertenecerán a un hilo secundario, el cual será mandado a llamar desde el método main.

```
1 referencia
static void MensajeM(object o)
{
    int n = 0;
    while (ejecutar)
    {
        Console.ForegroundColor = ConsoleColor.Green;
        Console.WriteLine("Saludos desde el hilo {0} - {1}", o, n);
        n++;
    }
}
```

Ilustración 7

Ejecución de la clase Hilos

```
Saludos desde el hiloprincipal 2
Saludos desde el hilo - 1
Saludos desde el hilo - 19
Saludos desde el hilo - 17
Saludos desde el hilo - 2
Saludos desde el hiloprincipal 3
Saludos desde el hilo - 18
Saludos desde el hilo - 18
Saludos desde el hilo - 3
Saludos desde el hiloprincipal 4
Saludos desde el hilo - 2
Saludos desde el hilo - 19
Saludos desde el hilo - 20
Saludos desde el hilo - 3
Saludos desde el hilo - 4
Saludos desde el hiloprincipal 5
Saludos desde el hilo - 20
```

Ilustración 8

```
Saludos desde el hilo - 50
Saludos desde el hilo - 77
Saludos desde el hilo - 81
Saludos desde el hiloprincipal 48
Saludos desde el hilo - 54
Saludos desde el hilo - 81
Saludos desde el hilo - 59
Saludos desde el hilo - 44
Saludos desde el hilo - 82
Saludos desde el hilo - 36
Saludos desde el hilo - 82
Saludos desde el hilo - 60
Saludos desde el hilo - 55
Saludos desde el hilo - 37
Saludos desde el hilo - 61
Saludos desde el hiloprincipal 49
Saludos desde el hilo - 83
Saludos desde el hilo - 83
Saludos desde el hilo - 62
Saludos desde el hilo - 80
Saludos desde el hilo - 45
Saludos desde el hilo - 78
Saludos desde el hilo - 56
Saludos desde el hilo - 38
```

Ilustración 9

Procedimiento problema 2

Agregamos las librerías necesarias para la implementación del código con que se estará trabajando, las librerías permiten agregar más elementos a una clase, lo cual permite referenciar ciertas cualidades a nuestro programa.

```
using System;  
using System.Threading;  
using System.IO;
```

Ilustración 10

Nombre de nuestra clase, donde se implementará el código necesario para darle solución a cierto problema que sea el fin, del porque se realiza el código.

```
1 referencia  
public class Hilos2  
{
```

Ilustración 11

Creamos un método estático de tipo entero, en el cual crearemos los parámetro para la variable n, este método obtendrá el valor de n y le sumará 1 a ese valor, obteniendo así, el nuevo valor de n.

```
1 referencia  
public static int CalculoComplejo(int n)  
{  
    n = n + 1;  
    Thread.Sleep(5000);  
    return n;  
}
```

Ilustración 12

Creamos una nueva clase dentro de la cual implementamos una función callback, la cual nos sirve para pasar el valor de una función a otra función, pero este pasaría como argumento.

```
3 referencias  
public class HiloParaMates  
{  
    protected int n;  
    protected MatesCallback callback = null;  
    1 referencia  
    public HiloParaMates(int n, MatesCallback callback)  
    {  
        this.n = n;  
        this.callback = callback;  
    }  
}
```

Ilustración 13

Creamos un nuevo método dentro del cual capturamos el valor(argumento) que se extrajo de la función callback y si el valor es diferente a nulo, es decir si existe algún valor, entonces retornamos ese valor.

```
public void CalculoComplejo()
{
    int result = Hilos2.CalculoComplejo(n);
    if (callback != null)
        callback(result);
}
```

Ilustración 14

Creamos un delegado con la firma necesaria para capturar el valor devuelto por el método CalculoComplejo.

```
public delegate void MatesCallback(int n);
```

Ilustración 15

Creamos una nueva clase y creamos una instancia donde se agrega un valor, el cual puede ser modificado en el código, en esta instancia se mandará a llamar el argumento que esta guardado en el método 'CalculoComplejo'.

```
0 referencias
public class Ejemplo
{
    0 referencias
    public static void Main()
    {
        HiloParaMates rd = new HiloParaMates(2000, new MatesCallback(ResultCallback));
        Thread th = new Thread(new ThreadStart(rd.CalculoComplejo));

        th.Start();
        th.Join();
    }
}
```

Ilustración 16

Además de crear una salida de texto donde se mostrará el valor que se obtenga.

```
1 referencia
public static void ResultCallback(int n)
{
    Console.WriteLine("Resultado de la operación: " + n);
}
```

Ilustración 17

Ejecución de la clase 'Hilos2'

Selección de la consola de depuración de Microsoft Visual Studio

Resultado de la operación: 2001

C:\Users\ACER\source\repos\Hilos2\bin\Debug\net6.0\Hilos2.exe

Ilustración 18

Consola de depuración de Microsoft Visual Studio

Resultado de la operación: 1001

C:\Users\ACER\source\repos\Hilos2\bin\Debug\net6.0\Hilos2.exe

Ilustración 19

Consola de depuración de Microsoft Visual Studio

Resultado de la operación: 501

C:\Users\ACER\source\repos\Hilos2\bin\Debug\net6.0\Hilos2.exe

Ilustración 20

Conclusión

Después de realizar esta práctica puedo decir que Visual Studio es un programa muy completo en cuanto a lo que son los diferentes tipos de proyectos que podemos realizar en él, sin duda ofrece una gran cantidad de herramientas que permiten el diseño de un formulario, así como también permite realizar programas en consola, la cual es muy útil si solo se quieren crear programas que impliquen solamente una solución mediante código.

Además, durante la elaboración de los programas se pudo comprender que los hilos son secuencias de control dentro de un proceso que ejecuta sus instrucciones de forma independiente. En un sistema operativo puede ejecutar varios procesos al mismo tiempo bien sea por concurrencia o paralelismo, dentro de un proceso puede haber varios hilos de ejecución.

También se descubrió que existen 2 formas de trabajar con hilos en cuanto a su creación se refiere: Heredando de la clase Thread e Implementando la interfaz Runnable, en este caso como se realizaron los programas en consola se ocupó la clase Thread.

Los hilos son una parte importante dentro de la programación, conocer sus cualidades y saber implementarlos posibilita la obtención de mejores soluciones a la hora de implementar varios procesos dentro de un mismo programa.