**Módulo: Características del Lenguaje C#**

**Tutorial: Estructuras**

**Ejemplo 1**

1. Definición de una estructura.

//Definición de una estructura

public struct Punto

{

public int x;

public int y;

}

1. Declarar la variable del tipo estructura. Instanciación de la estructura.

//Instanciacion

Punto p ;

p.x = 10;

p.y = 10;

1. Crear una nueva estructura del mismo tipo hacer la asignación .

Punto p1 = new Punto(0,0);

Console.WriteLine("Antes");

Console.WriteLine(p.ToString());

Console.WriteLine(p1.ToString());

p1 = p;

Console.WriteLine("Despues");

Console.WriteLine(p.ToString());

Console.WriteLine(p1.ToString());

Console.ReadLine();

1. Compilar y ejecutar
2. Uso de ReferenceEquals para comprobar que son diferentes direcciones de memoria.

if (ReferenceEquals(p, p1))

Console.WriteLine("Tienen la misma dirección de memoria");

else

Console.WriteLine("No tiene las misma dirección de memoria");

Console.ReadLine();

1. Crear una clase

class ClaseEjemplo

{

}

1. Verificar que los objetos de la clase cuando se asigna uno a otro tiene la misma dirección de memoria.

//Ejecicio de ReferenceEquals - Clase

ClaseEjemplo ClaseA = new ClaseEjemplo();

ClaseEjemplo ClaseB = new ClaseEjemplo();

ClaseA = ClaseB;

if (ReferenceEquals(ClaseA, ClaseB))

Console.WriteLine("Tienen la misma dirección de memoria");

else

Console.WriteLine("No tiene las misma dirección de memoria");

Console.ReadLine();

1. Sobrecargar el constructor de la estructura definida

//Nuevo Constructor

public Punto(int xPunto, int yPunto)

{

this.x = xPunto;

this.y = yPunto;

}

1. Además generar el override del método toString

public override string ToString()

{

return (String.Format("({0},{1})", x, y));

}