

## Ambito de Variables

```
public class AmbitoVariables
{
    // Variable Global
    static string nombre = "Jorge";
    2 referencias
    public static void Local()
    {
        string nombre = "Juan";
        Console.WriteLine(nombre);
        int suma = 0;

        for(int i=0; i < 5; i++)
        {
            suma = suma + i;
            Console.WriteLine($"Hola {nombre}; i {i}; suma {suma} ");
        }
        Console.WriteLine($"Termina Hola suma: {suma}");

        //No se puede acceder a la variable i
        Console.WriteLine($"Termina Hola i: {i}");
    }
}
```

CS0103: El nombre 'i' no existe en el contexto actual

[Mostrar posibles correcciones \(Alt+Entrar o Ctrl+.\)](#)

```
Juan
Hola Juan; i 0; suma 0
Hola Juan; i 1; suma 1
Hola Juan; i 2; suma 3
Hola Juan; i 3; suma 6
Hola Juan; i 4; suma 10
Termina Hola suma: 10
```

```
public class AmbitoVariables
{
    // Variable Global
    static string nombre = "Jorge";
    2 referencias
    public static void Local()
    {
        Console.WriteLine($"Hola: {nombre}");
        // No se pueden Acceder a las variables suma e i
        Console.WriteLine($"Termina Hola suma: {suma}");
        Console.WriteLine($"Termina Hola i: {i}");
    }
}
```

```
Hola: Jorge
```

## Cuerpo de Expresión

Una definición de cuerpo de expresión tiene la siguiente sintaxis general:

member => expresión;

```
private static string NombreCompletoDetallado()
{
    string nombreCompleto;
    nombreCompleto = $"{nombre} {apellidos}".Trim();
    return nombreCompleto;
}
```

```
private static string NombreCompletoSemidetallado()
{
    return $"{nombre} {apellidos}".Trim();
}
```

```
private static string NombreCompletoCuerpoExpresion() => $"{nombre} {apellidos}".Trim();
```

1 referencia

```
private static string NombreCompletoCuerpoExpresion(string nom, string app) => $"{nom} {app}".Trim();
```

```
public static void LLamaExpresion()
{
    nombre = "Jorge";
    apellidos = "Valdivia";
    string nombreCompleto;
    nombreCompleto = NombreCompletoDetallado();
    Console.WriteLine(nombreCompleto);
    nombreCompleto = NombreCompletoSemidetallado();
    Console.WriteLine(nombreCompleto);
    nombreCompleto = NombreCompletoCuerpoExpresion();
    Console.WriteLine(nombreCompleto);
    nombreCompleto = NombreCompletoCuerpoExpresion("Jorge", "Rosas");
    Console.WriteLine(nombreCompleto);
}
```

```
Jorge Valdivia
Jorge Valdivia
Jorge Valdivia
Jorge Rosas
```

## Pase por Valor y Pase por Referencia

```
private static int suma(int a, int b)
{
    a += b;
    Console.WriteLine("Dentro de Sumar");
    Console.WriteLine($"a: {a}; b: {b}");
    return a;
}
```

```
int a, b, c;
Console.WriteLine("Llamada por Valor");
a = 3; b = 5; c = 0;
Console.WriteLine("Antes de LLamar Sumar");
Console.WriteLine($"a: {a}; b: {b}; c:{c}");
c = suma(a, b);
Console.WriteLine("Después de LLamar Sumar");
Console.WriteLine($"a: {a}; b: {b}; c: {c}");
```

```
Llamada por Valor
Antes de LLamar Sumar
a: 3; b: 5; c:0
Dentro de Sumar
a: 8; b: 5
Después de LLamar Sumar
a: 3; b: 5; c: 8
```

```
private static int suma(ref int a, int b)
{
    a += b;
    Console.WriteLine("Dentro de Sumar");
    Console.WriteLine($"a: {a}; b: {b}");
    return a;
}

Console.WriteLine("Llamada por Referencia");
a = 3; b = 5; c = 0;
Console.WriteLine("Antes de LLamar Sumar");
Console.WriteLine($"a: {a}; b: {b}; c:{c}");
c = suma(ref a, b);
Console.WriteLine("Después de LLamar Sumar");
Console.WriteLine($"a: {a}; b: {b}; c: {c}");
```

```
Llamada por Referencia
Antes de LLamar Sumar
a: 3; b: 5; c:0
Dentro de Sumar
a: 8; b: 5
Después de LLamar Sumar
a: 8; b: 5; c: 8
```

```
private static void suma(int a, int b, out int d)
{
    a += b;
    d = a;
    Console.WriteLine("Dentro de Sumar");
    Console.WriteLine($"a: {a}; b: {b}; c: {d}");
}
```

```
Console.WriteLine("Llamada variable de Salida");
a = 3; b = 5; c = 0;
Console.WriteLine("Antes de LLamar Sumar");
Console.WriteLine($"a: {a}; b: {b}; c:{c}");
suma(a, b, out c);
Console.WriteLine("Después de LLamar Sumar");
Console.WriteLine($"a: {a}; b: {b}; c: {c}");
```

```
Llamada variable de Salida
Antes de LLamar Sumar
a: 3; b: 5; c:0
Dentro de Sumar
a: 8; b: 5; c: 8
Después de LLamar Sumar
a: 3; b: 5; c: 8
```

Pase por Valor y Referencia Estructuras

```
public struct CoordenadaStruct
{
    10 referencias
    public CoordenadaStruct(double x, double y)
    {
        X = x;
        Y = y;
    }

    34 referencias
    public double X { get; set; }
    3 referencias
    public double Y { get; set; }
    1 referencia
    public string Valores() => $"({X}, {Y})";
}
```

```

private static CoordenadaStruct suma(CoordenadaStruct a, CoordenadaStruct b)
{
    a.X += b.X;
    Console.WriteLine("Dentro de Sumar");
    Console.WriteLine($"a.X: {a.X}; b.X: {b.X}");
    return a;
}

CoordenadaStruct a, b, c;
Console.WriteLine("Llamada por Valor");
a = new CoordenadaStruct(1, 2); b = new CoordenadaStruct(3, 4); c = new CoordenadaStruct(0, 0);
Console.WriteLine("Antes de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");
c = suma(a, b);
Console.WriteLine("Después de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");

```

```

Llamada por Valor
Antes de LLamar Sumar
a.X: 1; b.X: 3; c.X: 0
Dentro de Sumar
a.X: 4; b.X: 3
Después de LLamar Sumar
a.X: 1; b.X: 3; c.X: 4

```

```

private static CoordenadaStruct suma(ref CoordenadaStruct a, CoordenadaStruct b)
{
    a.X += b.X;
    Console.WriteLine("Dentro de Sumar");
    Console.WriteLine($"a.X: {a.X}; b.X: {b.X}");
    return a;
}

```

```

Console.WriteLine("Llamada por Referencia");
a = new CoordenadaStruct(1, 2); b = new CoordenadaStruct(3, 4); c = new CoordenadaStruct(0, 0);
Console.WriteLine("Antes de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");
c = suma(ref a, b);
Console.WriteLine("Después de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");

```

```

Llamada por Referencia
Antes de LLamar Sumar
a.X: 1; b.X: 3; c.X: 0
Dentro de Sumar
a.X: 4; b.X: 3
Después de LLamar Sumar
a.X: 4; b.X: 3; c.X: 4

```

```

private static void suma(CoordenadaStruct a, CoordenadaStruct b, out CoordenadaStruct c)
{
    a.X += b.X;
    c = a;
    Console.WriteLine("Dentro de Sumar");
    Console.WriteLine($"a.X: {a.X}; b: {b.X}; c: {c.X}");
}

Console.WriteLine("Llamada variable de Salida");
a = new CoordenadaStruct(1, 2); b = new CoordenadaStruct(3, 4); c = new CoordenadaStruct(0, 0);
Console.WriteLine("Antes de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");
suma(a, b, out c);
Console.WriteLine("Después de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");

```

```

Llamada variable de Salida
Antes de LLamar Sumar
a.X: 1; b.X: 3; c.X: 0
Dentro de Sumar
a.X: 4; b: 3; c: 4
Después de LLamar Sumar
a.X: 1; b.X: 3; c.X: 4

```

## Llamada por Referencia de Clases

```

public class CoordenadaClase
{
    12 referencias
    public CoordenadaClase(double x, double y)
    {
        X = x;
        Y = y;
    }

    43 referencias
    public double X { get; set; }
    2 referencias
    public double Y { get; set; }
    0 referencias
    public string Valores() => $"({X}, {Y})";
}

private static CoordenadaClase suma(CoordenadaClase a, CoordenadaClase b)
{
    a.X += b.X;
    Console.WriteLine("Dentro de Sumar");
    Console.WriteLine($"a.X: {a.X}; b.X: {b.X}");
    return a;
}

```

```

CoordenadaClase a, b, c;
Console.WriteLine("Llamada por Valor");
a = new CoordenadaClase(1, 2); b = new CoordenadaClase(3, 4); c = new CoordenadaClase(0, 0);
Console.WriteLine("Antes de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");
c = suma(a, b);
Console.WriteLine("Después de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");

```

```

Llamada por Valor
Antes de LLamar Sumar
a.X: 1; b.X: 3; c.X: 0
Dentro de Sumar
a.X: 4; b.X: 3
Después de LLamar Sumar
a.X: 4; b.X: 3; c.X: 4

```

```

private static CoordenadaClase suma(ref CoordenadaClase a, CoordenadaClase b)
{
    a.X += b.X;
    Console.WriteLine("Dentro de Sumar");
    Console.WriteLine($"a.X: {a.X}; b.X: {b.X}");
    return a;
}

```

```

Console.WriteLine("Llamada por Referencia");
a = new CoordenadaClase(1, 2); b = new CoordenadaClase(3, 4); c = new CoordenadaClase(0, 0);
Console.WriteLine("Antes de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");
c = suma(ref a, b);
Console.WriteLine("Después de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");

```

```

Llamada por Referencia
Antes de LLamar Sumar
a.X: 1; b.X: 3; c.X: 0
Dentro de Sumar
a.X: 4; b.X: 3
Después de LLamar Sumar
a.X: 4; b.X: 3; c.X: 4

```

```

private static void suma(CoordenadaClase a, CoordenadaClase b, out CoordenadaClase c)
{
    a.X += b.X;
    c = a;
    Console.WriteLine("Dentro de Sumar");
    Console.WriteLine($"a.X: {a.X}; b: {b.X}; c: {c.X}");
}

```

```

Console.WriteLine("Llamada variable de Salida");
a = new CoordenadaClase(1, 2); b = new CoordenadaClase(3, 4); c = new CoordenadaClase(0, 0);
Console.WriteLine("Antes de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");
suma(a, b, out c);
Console.WriteLine("Después de LLamar Sumar");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");

```

```

Llamada variable de Salida
Antes de LLamar Sumar
a.X: 1; b.X: 3; c.X: 0
Dentro de Sumar
a.X: 4; b.X: 3; c.X: 4
Después de LLamar Sumar
a.X: 4; b.X: 3; c.X: 4

```

```

Console.WriteLine("Asignacion de Clases");
a = new CoordenadaClase(1, 2); b = new CoordenadaClase(3, 4); c = new CoordenadaClase(0, 0);
Console.WriteLine("Antes de Asignar clase");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");
a = b;
Console.WriteLine("Después de Asignar clase");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");
b.X = 8;
Console.WriteLine("Después de Asignar propiedad");
Console.WriteLine($"a.X: {a.X}; b.X: {b.X}; c.X: {c.X}");

```

```

Asignacion de Clases
Antes de Asignar clase
a.X: 1; b.X: 3; c.X: 0
Después de Asignar clase
a.X: 3; b.X: 3; c.X: 0
Después de Asignar propiedad
a.X: 8; b.X: 8; c.X: 0

```