

Bernardo Amparo, 2024-1728, clase de viernes

1. Declarar variable de los diferentes tipos, asignarles valor e imprimir el valor.

```
namespace DeclararVariables
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            int num1 = 21;
            Console.WriteLine("Siglo: " + num1);

            double num2 = 33.19;
            Console.WriteLine("Peso: " + num2);

            string nombre = "Bernardo Amparo";
            Console.WriteLine("Nombre: " + nombre);

            bool tieneVehiculo = true;
            Console.WriteLine("¿Es rojo?: " + tieneVehiculo);

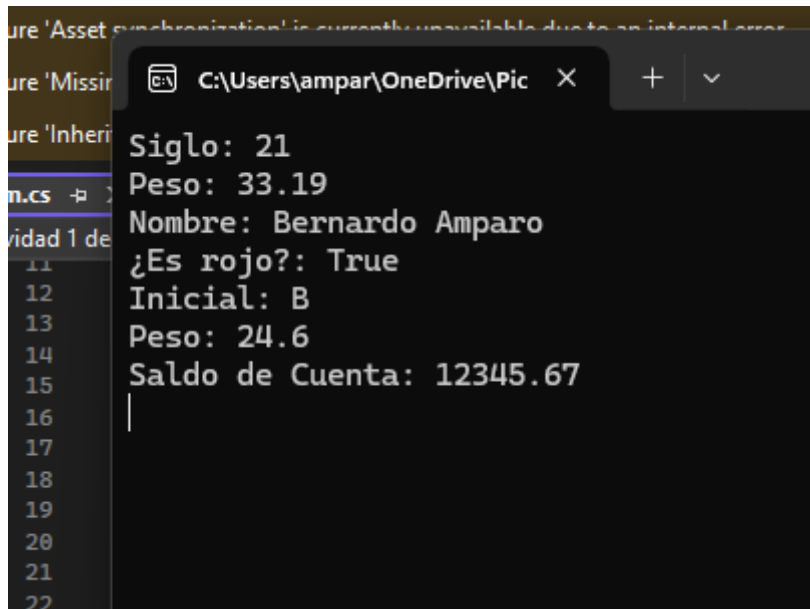
            char letra = 'B';
            Console.WriteLine("Inicial: " + letra);

            float num3 = 24.6f;
            Console.WriteLine("Peso: " + num3);
```

```
            decimal saldoCuenta = 12345.67m;
            Console.WriteLine("Saldo de Cuenta: " + saldoCuenta);

            Console.ReadKey();
        }
    }
}
```

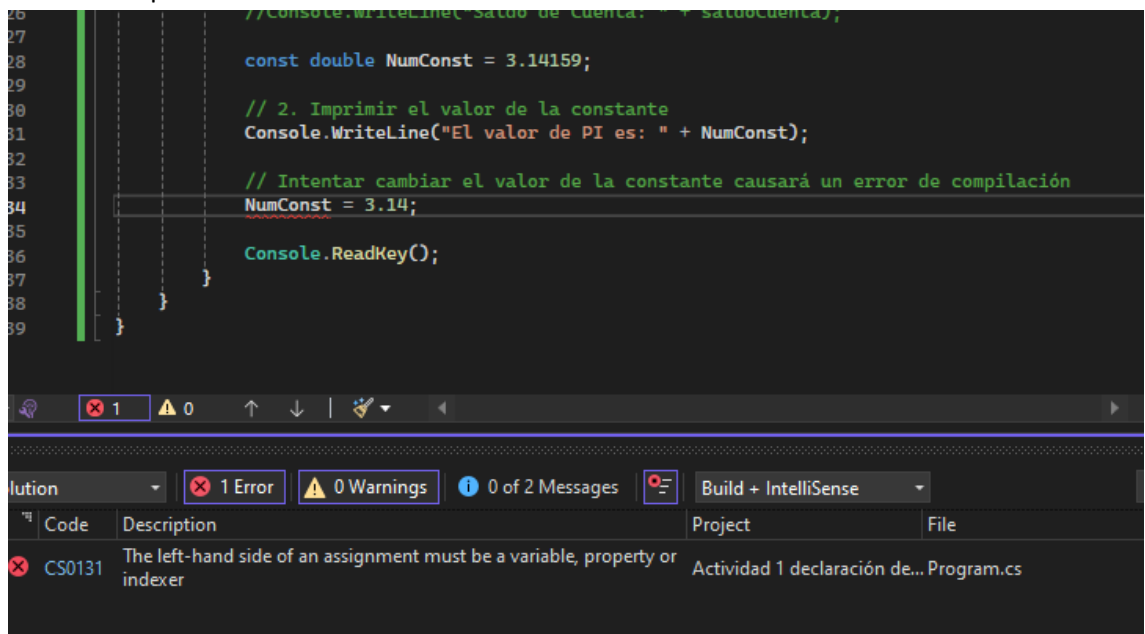
C# tarea inicial



```
Siglo: 21
Peso: 33.19
Nombre: Bernardo Amparo
¿Es rojo?: True
Inicial: B
Peso: 24.6
Saldo de Cuenta: 12345.67
```

2. Buscar cómo se declara una constante en C# e imprimir el valor. Probar de cambiar su valor luego y ver que es lo que pasa.

En C#, una constante se declara utilizando la palabra clave `const` seguida del tipo de dato, el nombre de la constante y su valor. Es fundamental que la constante sea inicializada en el mismo momento de su declaración, ya que su valor no podrá ser modificado posteriormente.



```
//Console.WriteLine("Saldo de Cuenta: " + SaldoCuenta);

const double NumConst = 3.14159;

// 2. Imprimir el valor de la constante
Console.WriteLine("El valor de PI es: " + NumConst);

// Intentar cambiar el valor de la constante causará un error de compilación
NumConst = 3.14;

Console.ReadKey();
}
```

Build + IntelliSense

Code	Description	Project	File
CS0131	The left-hand side of an assignment must be a variable, property or indexer	Actividad 1 declaración de...	Program.cs

3. Declara un entero, incrementarlo, decrementarlo, hacer operación con el.

C# tarea inicial

```
namespace DeclararVariables
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            int number = 19;
            Console.WriteLine($"Valor inicial del número: {number}");

            number++;
            Console.WriteLine($"Después de incrementar (numero++): {number}");

            number--; // Esto es equivalente a numero = numero - 1;
            Console.WriteLine($"Después de decrementar (numero--): {number}");

            // Multiplicación
            int Multiplicacion = number * 5;
            Console.WriteLine($"Multiplicar por 5: {number} * 5 = {Multiplicacion}");

            // División
            int resultadoDivision = number / 2;
            Console.WriteLine($"Dividir por 2: {number} / 2 = {resultadoDivision}");
        }
    }
}
```

```
// Módulo (resto de la división)
int resto = number % 3;
Console.WriteLine($"Resto de la división por 3: {number} % 3 = {resto}");

// Combinación de operaciones
int otroNumero = 43;
int operacionCompleja = (number + otroNumero) * 2 - (resto / 1);
Console.WriteLine($"Operación compleja: ({number} + {otroNumero}) * 2 - ({resto} / 1) = {operacionCompleja}");

Console.ReadKey();
}
```

```
C:\Users\ampar\OneDrive\Pic X + v

Valor inicial del número: 19
Después de incrementar (numero++): 20
Después de decrementar (numero--): 19
Multiplicar por 5: 19 * 5 = 95
Dividir por 2: 19 / 2 = 9
Resto de la división por 3: 19 % 3 = 1
Operación compleja: (19 + 43) * 2 - (1 / 1) = 123
```

4. Declarar un float con valor=10152466.25. Declara un byte que es igual a 5 + el float.

C# tarea inicial

5. Adjuntar comentario de una y de varias líneas un su código. Imprimir la fecha y hora d el sistema.

```
namespace DeclararVariables
{
    0 references
    class Program
    {
        0 references
        static void Main(string[] args)
        {
            /*
             * Comentario de varias líneas:
             * imprimiremos la fecha y hora actuales del sistema.
             * Esto es útil para logs, marcas de tiempo,
             * o simplemente para mostrar la hora al usuario.
             */

            // Obtener la fecha y hora actuales del sistema
            // DateTime.Now devuelve un objeto DateTime que representa la fecha y hora actual.
            DateTime fechaHoraActual = DateTime.Now;

            // Imprimir la fecha y hora completas
            Console.WriteLine("\nFecha y Hora actuales del sistema: " + fechaHoraActual);

            Console.ReadKey();
        }
    }
}
```

```
C:\Users\ampar\OneDrive\Pic X + v
Fecha y Hora actuales del sistema: 30/5/2025 6:07:48 p. m.
```