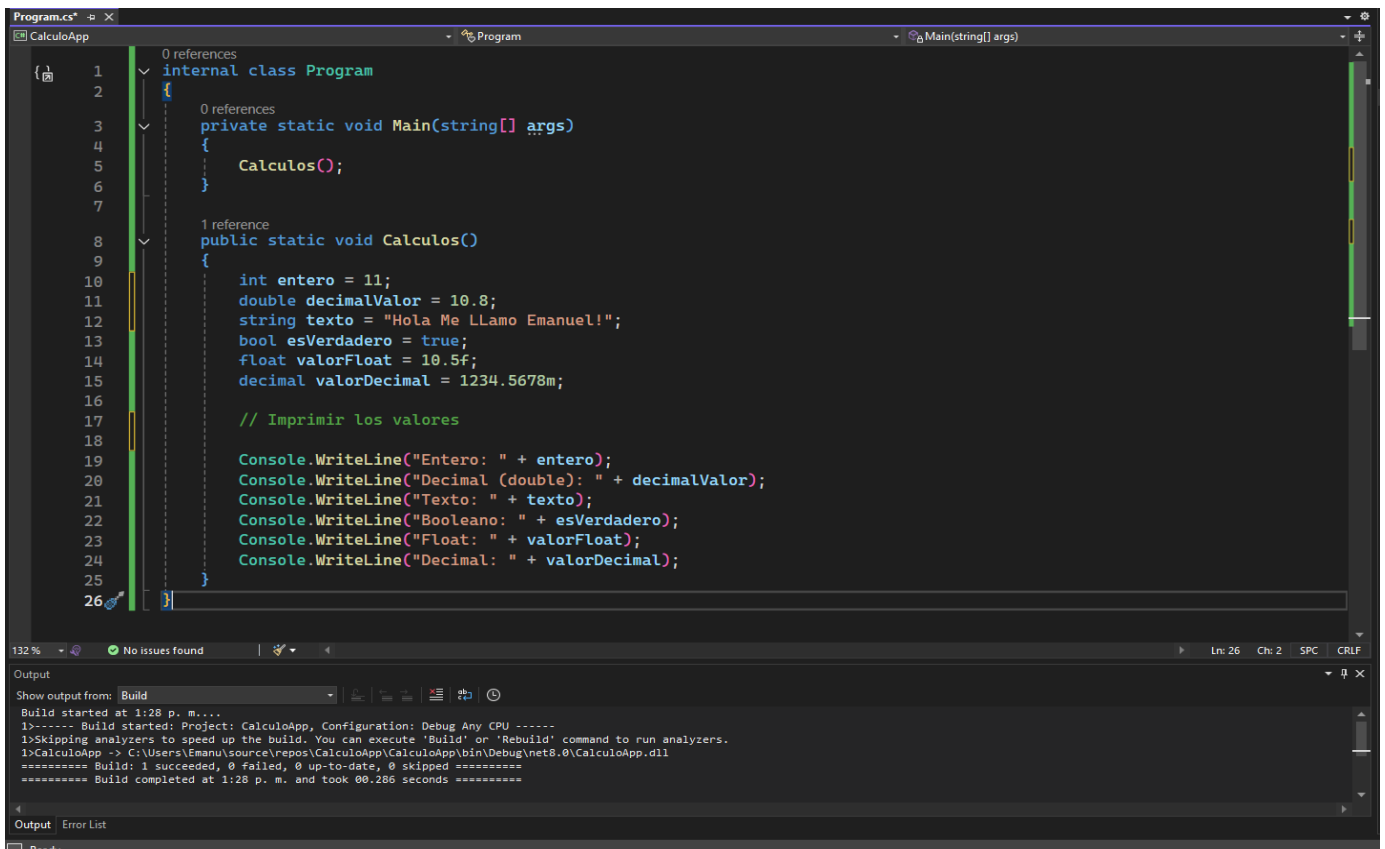


# C# tarea inicial

Emanuel De La Cruz Balbuena, 2024-1720, Viernes.

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



```
Program.cs* x
CalculoApp
0 references
internal class Program
{
0 references
private static void Main(string[] args)
{
    Calculos();
}

1 reference
public static void Calculos()
{
    int entero = 11;
    double decimalValor = 10.8;
    string texto = "Hola Me LLamo Emanuel!";
    bool esVerdadero = true;
    float valorFloat = 10.5f;
    decimal valorDecimal = 1234.5678m;

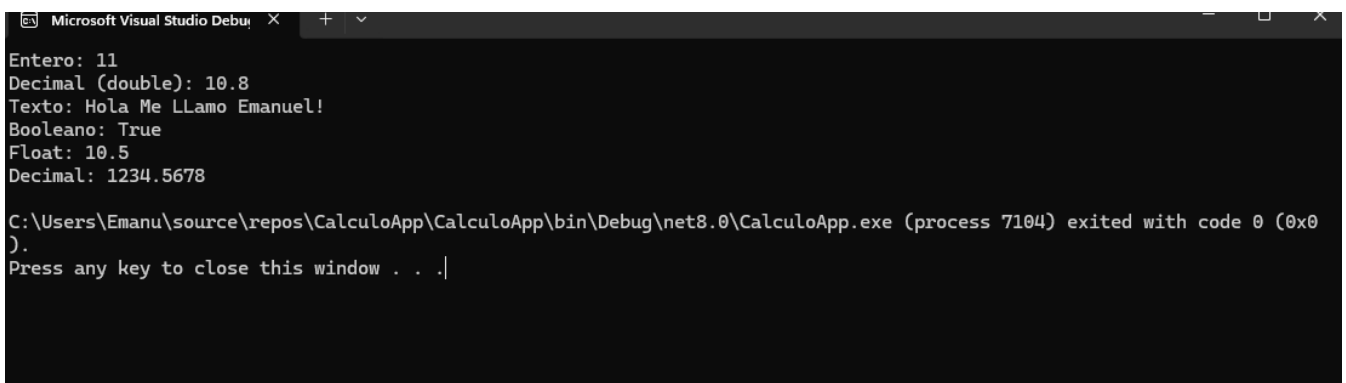
    // Imprimir los valores

    Console.WriteLine("Entero: " + entero);
    Console.WriteLine("Decimal (double): " + decimalValor);
    Console.WriteLine("Texto: " + texto);
    Console.WriteLine("Booleano: " + esVerdadero);
    Console.WriteLine("Float: " + valorFloat);
    Console.WriteLine("Decimal: " + valorDecimal);
}
}

132% No issues found Ln: 26 Ch: 2 SPC CRLF

Output
Show output from: Build
Build started at 1:28 p. m....
1>----- Build started: Project: CalculoApp, Configuration: Debug Any CPU -----
1>Skipping analyzers to speed up the build. You can execute 'Build' or 'Rebuild' command to run analyzers.
1>CalculoApp -> C:\Users\Emanu\source\repos\CalculoApp\CalculoApp\bin\Debug\net8.0\CalculoApp.dll
***** Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped *****
***** Build completed at 1:28 p. m. and took 00.286 seconds *****

Output Error List
Ready
```



```
Microsoft Visual Studio Debug Console
Entero: 11
Decimal (double): 10.8
Texto: Hola Me LLamo Emanuel!
Booleano: True
Float: 10.5
Decimal: 1234.5678

C:\Users\Emanu\source\repos\CalculoApp\CalculoApp\bin\Debug\net8.0\CalculoApp.exe (process 7104) exited with code 0 (0x0).
Press any key to close this window . . .|
```

## 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.

Una constante en C# se declara como una variable cuyo valor no puede cambiar después de ser asignado. Es decir, una vez que le asignas un valor a una constante, no puedes modificarlo más adelante en el código. Esto garantiza que el valor permanezca constante durante toda la ejecución del programa. Para declarar una constante, se hace lo siguiente:

# C# tarea inicial

1. Usar la palabra clave **const** antes del tipo de dato.
2. Asignarle un valor al momento de la declaración. Este valor debe ser fijo, ya que no se podrá modificar después.
3. Normalmente, las constantes se nombran usando letras mayúsculas, por convención, para diferenciarlas de las variables normales.

Cuando intentamos cambiar el valor de una constante el programa no compila ya que las constantes no pueden ser modificadas una vez se ha asignado su valor.

The screenshot shows a C# program in Visual Studio. The code defines an internal class `Program` with a `Main` method that calls `Calculos()`. The `Calculos` method contains a constant `PI = 3.14159` and an attempt to reassign it to `4.12`. This causes a compilation error: `CS0131: The left-hand side of an assignment must be a variable, property or indexer`. The error message is displayed in a tooltip and the Output window.

```
1 internal class Program
2 {
3     private static void Main(string[] args)
4     {
5         Calculos();
6     }
7
8     public static void Calculos()
9     {
10        //Constante
11        const double PI = 3.14159;
12
13
14        PI = 4.12;
15
16    }
17
18 }
```

Output:

```
Build started: Project: CalculoApp, Configuration: Debug Any CPU -----
1>Skipping analyzers to speed up the build. You can execute 'Build' or 'Rebuild' command to run analyzers.
1>C:\Users\Emanu\source\repos\CalculoApp\CalculoApp\Program.cs(14,10,14,12): error CS0131: The left-hand side of an assignment must be a variable, property or indexer
1>Done building project "CalculoApp.csproj" -- FAILED.
===== Build: 0 succeeded, 1 failed, 0 up-to-date, 0 skipped =====
===== Build completed at 1:46 p. m. and took 00.198 seconds =====
```

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

The screenshot shows a C# program in Visual Studio. The code defines an internal class `Program` with a `Main` method that calls `Calculos()`. The `Calculos` method contains a constant `numero = 14`, increments it, decrements it, and performs arithmetic operations (sum, difference, multiplication). The code is compiled successfully, and the Output window shows the build output.

```
1 internal class Program
2 {
3     private static void Main(string[] args)
4     {
5         Calculos();
6     }
7
8     public static void Calculos()
9     {
10        int numero = 14;
11
12        // Incrementar
13        numero++;
14        Console.WriteLine("Incrementado: " + numero);
15
16        // Decrementar
17        numero--;
18        Console.WriteLine("Decrementado: " + numero);
19
20        // Operaciones
21        int suma = numero + 7;
22        int resta = numero - 4;
23        int multiplicacion = numero * 2;
24
25        Console.WriteLine("Suma: " + suma);
26        Console.WriteLine("Resta: " + resta);
27        Console.WriteLine("Multiplicación: " + multiplicacion);
28    }
29 }
```

Output:

```
Build started at 1:54 p. m. ....
1>----- Build started: Project: CalculoApp, Configuration: Debug Any CPU -----
1>Skipping analyzers to speed up the build. You can execute 'Build' or 'Rebuild' command to run analyzers.
1>CalculoApp -> C:\Users\Emanu\source\repos\CalculoApp\CalculoApp\bin\Debug\net8.0\CalculoApp.dll
===== Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped =====
===== Build completed at 1:54 p. m. and took 00.204 seconds =====
```

## C# tarea inicial

```
Microsoft Visual Studio Debug Console

Incrementado: 15
Decrementado: 14
Suma: 21
Resta: 10
Multiplicación: 28

C:\Users\Emanu\source\repos\CalculoApp\CalculoApp\bin\Debug\net8.0\CalculoApp.exe (process 22660) exited with code 0 (0x0).
Press any key to close this window . . .|
```

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

```
Program.cs
1 internal class Program
2 {
3     0 references
4     private static void Main(string[] args)
5     {
6         Calculos();
7     }
8     1 reference
9     public static void Calculos()
10    {
11        float valorDelFloat = 10152466.25f;
12        byte valorDelByte = (byte)(5 + valorDelFloat);
13
14        Console.WriteLine("El Valor en float es: " + valorDelFloat);
15        Console.WriteLine("El Valor en byte es: " + valorDelByte);
16    }
17 }
```

Output

```
Show output from: Build
Build started at 2:01 p. m....
1>----- Build started: Project: CalculoApp, Configuration: Debug Any CPU -----
1>Skipping analyzers to speed up the build. You can execute 'Build' or 'Rebuild' command to run analyzers.
1>CalculoApp -> C:\Users\Emanu\source\repos\CalculoApp\CalculoApp\bin\Debug\net8.0\CalculoApp.dll
***** Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped *****
***** Build completed at 2:01 p. m. and took 00.263 seconds *****
```

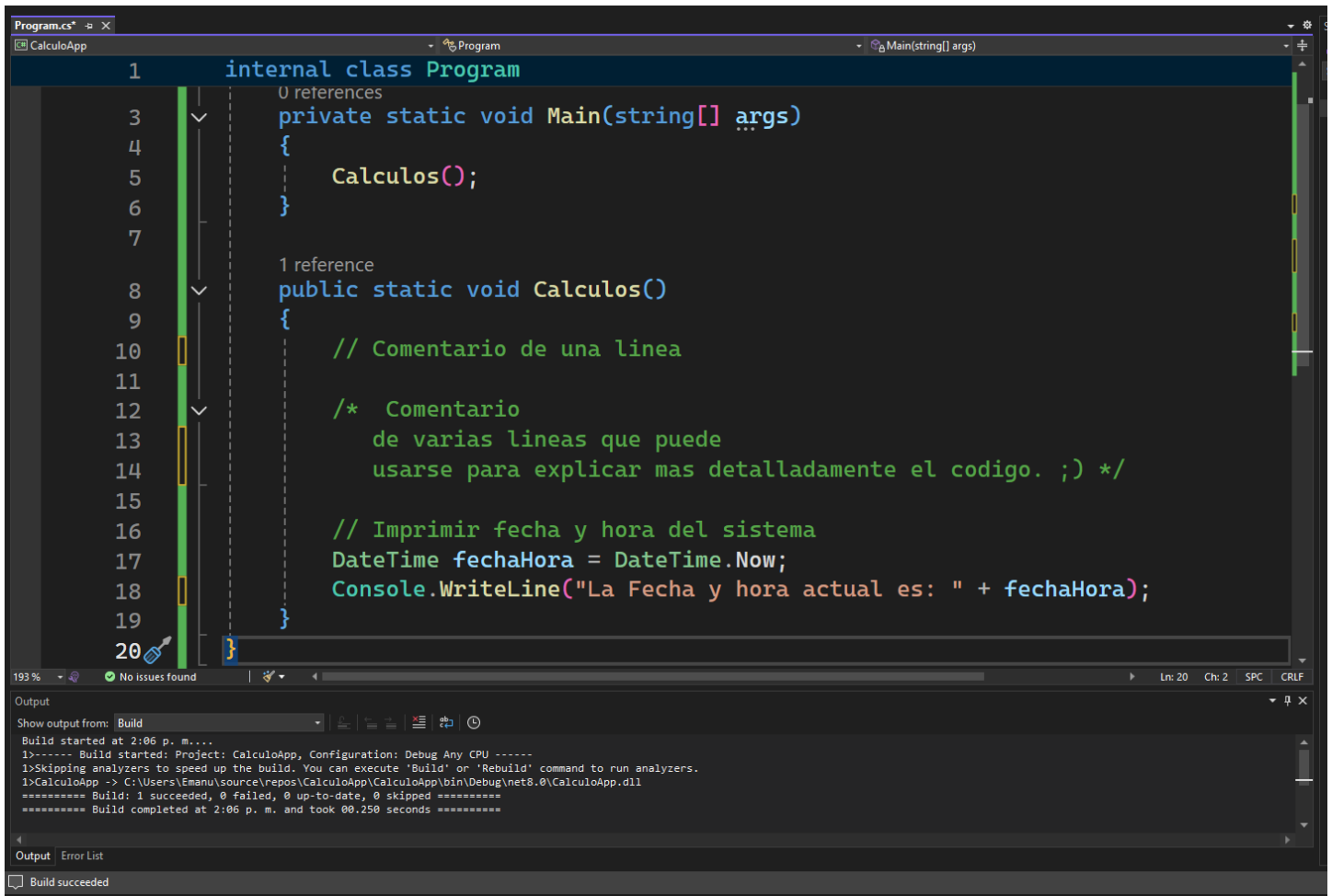
```
Microsoft Visual Studio Debug Console

El Valor en float es: 10152466
El Valor en byte es: 23

C:\Users\Emanu\source\repos\CalculoApp\CalculoApp\bin\Debug\net8.0\CalculoApp.exe (process 28072) exited with code 0 (0x0).
Press any key to close this window . . .|
```

## C# tarea inicial

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



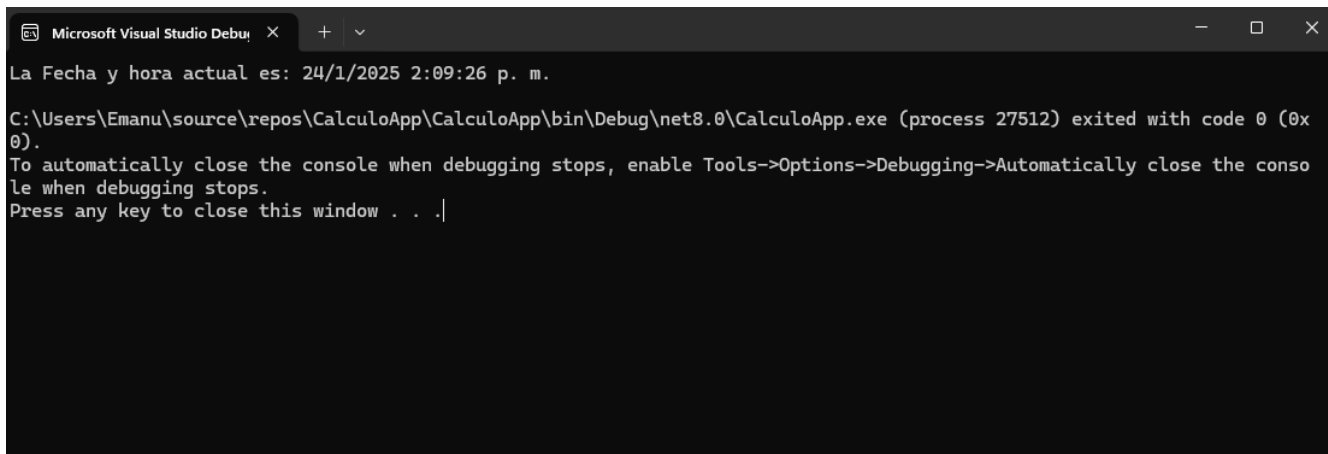
The screenshot shows the Visual Studio IDE with the file `Program.cs` open. The code defines an internal class `Program` with two methods: `Main` and `Calculos`. The `Calculos` method contains three types of comments: a single-line comment, a multi-line comment, and a comment that spans multiple lines. It also prints the current date and time to the console. The output window at the bottom shows the build process, which completed successfully at 2:06 p.m.

```
1 internal class Program
2 {
3     private static void Main(string[] args)
4     {
5         Calculos();
6     }
7
8     public static void Calculos()
9     {
10        // Comentario de una linea
11
12        /* Comentario
13           de varias lineas que puede
14           usarse para explicar mas detalladamente el codigo. ;) */
15
16        // Imprimir fecha y hora del sistema
17        DateTime fechaHora = DateTime.Now;
18        Console.WriteLine("La Fecha y hora actual es: " + fechaHora);
19    }
20 }
```

Output

```
Build started at 2:06 p. m....
1>----- Build started: Project: CalculoApp, Configuration: Debug Any CPU -----
1>Skipping analyzers to speed up the build. You can execute 'Build' or 'Rebuild' command to run analyzers.
1>CalculoApp -> C:\Users\Emanu\source\repos\CalculoApp\CalculoApp\bin\Debug\net8.0\CalculoApp.dll
***** Build: 1 succeeded, 0 failed, 0 up-to-date, 0 skipped *****
***** Build completed at 2:06 p. m. and took 00.250 seconds *****
```

Build succeeded



The screenshot shows the Visual Studio Debug Console. The first line is the output of the program: "La Fecha y hora actual es: 24/1/2025 2:09:26 p. m.". The second line is a message from the debugger indicating that the application has exited with code 0. The third line is a prompt from the debugger asking the user to press any key to close the window.

```
La Fecha y hora actual es: 24/1/2025 2:09:26 p. m.

C:\Users\Emanu\source\repos\CalculoApp\CalculoApp\bin\Debug\net8.0\CalculoApp.exe (process 27512) exited with code 0 (0x0).
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the console when debugging stops.
Press any key to close this window . . .|
```