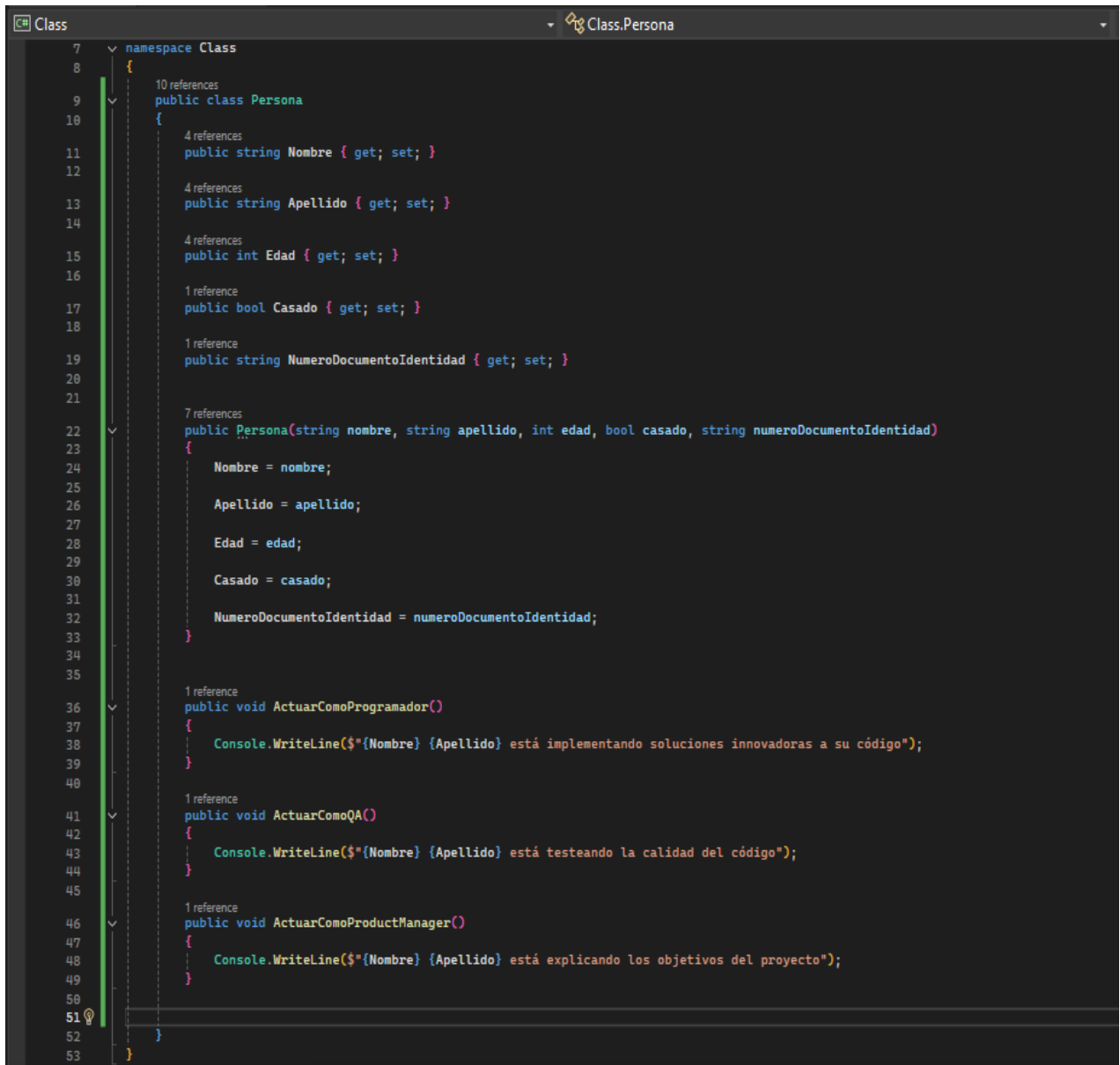


## Emanuel De La Cruz Balbuena, 2024-1720, Viernes.

1. Considera estás desarrollando un programa donde necesitas trabajar con objetos de tipo Persona. Define una clase Persona, pero en este caso considerando los siguientes atributos de clase: nombre (String), apellidos (String), edad (int), casado (boolean), numeroDocumentoIdentidad(String) y 3 metodos como acciones diferentes por persona de acuerdo a una profesión. Define un constructor y los métodos para poder establecer y obtener los valores de los atributos. Mínimo 7 personas diferentes con acciones diferentes.



```
7 namespace Class
8 {
9     10 references
10     public class Persona
11     {
12         4 references
13         public string Nombre { get; set; }
14
15         4 references
16         public string Apellido { get; set; }
17
18         4 references
19         public int Edad { get; set; }
20
21         1 reference
22         public bool Casado { get; set; }
23
24         1 reference
25         public string NumeroDocumentoIdentidad { get; set; }
26
27         7 references
28         public Persona(string nombre, string apellido, int edad, bool casado, string numeroDocumentoIdentidad)
29         {
30             Nombre = nombre;
31             Apellido = apellido;
32             Edad = edad;
33             Casado = casado;
34             NumeroDocumentoIdentidad = numeroDocumentoIdentidad;
35         }
36
37         1 reference
38         public void ActuarComoProgramador()
39         {
40             Console.WriteLine($"{Nombre} {Apellido} está implementando soluciones innovadoras a su código");
41         }
42
43         1 reference
44         public void ActuarComoQA()
45         {
46             Console.WriteLine($"{Nombre} {Apellido} está testeando la calidad del código");
47         }
48
49         1 reference
50         public void ActuarComoProductManager()
51         {
52             Console.WriteLine($"{Nombre} {Apellido} está explicando los objetivos del proyecto");
53         }
54     }
55 }
```

```
Persona.cs Program.cs X
Class
1
2
3 using Class;
4
5 List<Persona> listaPersonas = new List<Persona>
6 {
7     new Persona("Emanuel", "De La Cruz", 30, false, "922892838"),
8     new Persona("Eutilio", "Mendoza", 28, true, "363625252"),
9     new Persona("Pedro", "Martinez", 40, false, "334343434"),
10    new Persona("Starling", "Germosen", 22, true, "4343434"),
11    new Persona("Fernanda", "Arias", 45, false, "234343455"),
12    new Persona("Josh", "Rodriguez", 32, true, "343456776"),
13    new Persona("Santiago", "Mazza", 50, false, "37636626")
14 };
15
16
17
18 foreach (var persona in listaPersonas)
19 {
20     if (persona.Edad < 30)
21     {
22         persona.ActuarComoProgramador();
23     }
24
25     else if (persona.Edad >= 30 && persona.Edad <= 40)
26     {
27         persona.ActuarComoProductManager();
28     }
29
30     else
31     {
32         persona.ActuarComoQA();
33     }
34 }
35
```

```
Microsoft Visual Studio Debug X + v
Emanuel De La Cruz está explicando los objetivos del proyecto
Eutilio Mendoza está implementando soluciones innovadoras a su código
Pedro Martinez está explicando los objetivos del proyecto
Starling Germosen está implementando soluciones innovadoras a su código
Fernanda Arias está testeando la calidad del código
Josh Rodriguez está explicando los objetivos del proyecto
Santiago Mazza está testeando la calidad del código

C:\Users\Emanu\source\repos\Class\bin\Debug\net8.0\Class.exe (process 31580) 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 . . .|
```

2. Crea una clase Cuenta con los métodos ingreso, reintegro y transferencia. La clase contendrá un constructor por defecto, un constructor con parámetros y los métodos getters y setters para mostrar e ingresar.

```
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Class
8  {
9      7 references
10     public class Cuenta
11     {
12         8 references
13         public string Titular { get; set; }
14
15         12 references
16         public decimal Saldo { get; set; }
17
18         0 references
19         public Cuenta()
20         {
21             Titular = "Alguien...";
22             Saldo = 0;
23         }
24
25         2 references
26         public Cuenta(string titular, decimal saldoInicial)
27         {
28             Titular = titular;
29             Saldo = saldoInicial;
30         }
31
32         1 reference
33         public string GetTitular()
34         {
35             return Titular;
36         }
37
38         0 references
39         public void SetTitular(string nuevoTitular)
40         {
41             Titular = nuevoTitular;
42         }
43
44         // Metodo Get y Set para el saldo (para mostrar e ingresar)
45         1 reference
46         public decimal GetSaldo()
47         {
48             return Saldo;
49         }
50
51         0 references
52         public void SetSaldo(decimal nuevoSaldo)
53         {
54             if (nuevoSaldo >= 0)
55             {
56                 // ...
57             }
58         }
59     }
60 }
```

```
Class
7 namespace Class
9 public class Cuenta
40 public decimal GetSaldo()
44
45 0 references
46 public void SetSaldo(decimal nuevoSaldo)
47 {
48     if (nuevoSaldo >= 0)
49     {
50         Saldo = nuevoSaldo;
51     }
52     else
53     {
54         Console.WriteLine("El saldo no puede ser negativo.");
55     }
56 }
57
58 4 references
59 public void MostrarDetallesCuenta()
60 {
61     Console.WriteLine($"Titular: {GetTitular()}");
62     Console.WriteLine($"Saldo actual: {GetSaldo():C}");
63 }
64
65 3 references
66 public void Ingreso(decimal cantidad)
67 {
68     if (cantidad > 0)
69     {
70         Saldo += cantidad;
71         Console.WriteLine($"Se ha ingresado {cantidad:C} a la cuenta de {Titular}. Saldo actual: {Saldo:C}");
72     }
73     else
74     {
75         Console.WriteLine("El monto de ingreso debe ser positivo.");
76     }
77 }
78
79 2 references
80 public void Reintegro(decimal cantidad)
81 {
82     if (cantidad > 0 && cantidad <= Saldo)
83     {
84         Saldo -= cantidad;
85         Console.WriteLine($"Se ha retirado {cantidad:C} de la cuenta de {Titular}. Saldo actual: {Saldo:C}");
86     }
87     else
88     {
89         Console.WriteLine("Saldo insuficiente o monto no válido.");
90     }
91 }
92
93 1 reference
94 public void Transferencia(Cuenta cuentaDestino, decimal cantidad)
95 {
96     if (cantidad > 0 && cantidad <= Saldo)
97     {
98         Reintegro(cantidad);
99         cuentaDestino.Ingreso(cantidad);
100         Console.WriteLine($"Se han transferido {cantidad:C} de {Titular} a {cuentaDestino.Titular}. Saldo origen: {Saldo:C}. Saldo destino: {cuentaDestino.Saldo:C}");
101     }
102     else
103     {
104         Console.WriteLine("Transferencia no válida: saldo insuficiente o monto no válido.");
105     }
106 }
```

```
1 reference
90 public void Transferencia(Cuenta cuentaDestino, decimal cantidad)
91 {
92     if (cantidad > 0 && cantidad <= Saldo)
93     {
94         Reintegro(cantidad);
95         cuentaDestino.Ingreso(cantidad);
96         Console.WriteLine($"Se han transferido {cantidad:C} de {Titular} a {cuentaDestino.Titular}. Saldo origen: {Saldo:C}. Saldo destino: {cuentaDestino.Saldo:C}");
97     }
98     else
99     {
100         Console.WriteLine("Transferencia no válida: saldo insuficiente o monto no válido.");
101     }
102 }
103
104 }
105
106 }
```

```
Microsoft Visual Studio Debug Console
Titular: Emanuel De La Cruz
Saldo actual: $1,500.00
Titular: Starling Germosen
Saldo actual: $700.00
Se ha ingresado $200.00 a la cuenta de Emanuel De La Cruz. Saldo actual: $1,700.00
Se ha ingresado $300.00 a la cuenta de Starling Germosen. Saldo actual: $1,000.00
Se ha retirado $100.00 de la cuenta de Emanuel De La Cruz. Saldo actual: $1,600.00
Se ha retirado $150.00 de la cuenta de Emanuel De La Cruz. Saldo actual: $1,450.00
Se ha ingresado $150.00 a la cuenta de Starling Germosen. Saldo actual: $1,150.00
Se han transferido $150.00 de Emanuel De La Cruz a Starling Germosen. Saldo origen: $1,450.00. Saldo destino: $1,150.00
Titular: Emanuel De La Cruz
Saldo actual: $1,450.00
Titular: Starling Germosen
Saldo actual: $1,150.00

C:\Users\Emanu\source\repos\Class\bin\Debug\net8.0\Class.exe (process 21452) 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 . . .|
```

3. Crea una clase Contador con los métodos para incrementar y decrementar el contador. La clase contendrá un constructor por defecto, un constructor con parámetros, y los métodos getters y setters.

```
Class
1  using System;
2  using System.Collections.Generic;
3  using System.Linq;
4  using System.Text;
5  using System.Threading.Tasks;
6
7  namespace Class
8  {
9      6 references
10     public class Contador
11     {
12         private int _value;
13
14         1 reference
15         public Contador()
16         {
17             _value = 0;
18         }
19
20         1 reference
21         public Contador(int initialValue)
22         {
23             _value = initialValue;
24         }
25
26         4 references
27         public int GetValue()
28         {
29             return _value;
30         }
31
32         0 references
33         public void SetValue(int newValue)
34         {
35             _value = newValue ;
36         }
37
38         2 references
39         public void Incrementar()
40         {
41             _value++;
42             Console.WriteLine($"El contador se ha incrementado {GetValue()}");
43         }
44
45         2 references
46         public void Decrementar()
47         {
48             _value--;
49             Console.WriteLine($"El contador se ha decrementado {GetValue()}");
50         }
51     }
52 }
```

```
Contador contador1 = new Contador();  
Console.WriteLine($"Valor inicial de contador1: {contador1.GetValue()}");  
  
contador1.Incrementar();  
contador1.Decrementar();  
  
Contador contador2 = new Contador(14);  
Console.WriteLine($"Valor inicial de contador2: {contador2.GetValue()}");  
  
contador2.Incrementar();  
contador2.Decrementar();
```

Microsoft Visual Studio Debug Console

Valor inicial de contador1: 0  
El contador se ha incrementado 1  
El contador se ha decrementado 0  
Valor inicial de contador2: 14  
El contador se ha incrementado 15  
El contador se ha decrementado 14

C:\Users\Emanu\source\repos\Class\bin\Debug\net8.0\Class.exe (process 33652) 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 . . .

4. Crea una clase Libro con los métodos préstamo, devolución y ToString. La clase contendrá un constructor por defecto, un constructor con parámetros y los métodos getters y setters.

```
PracticandoClasesPOO
PracticandoClasesPOO.Libro

7
8 namespace PracticandoClasesPOO
9 {
10     14 references
11     public class Libro
12     {
13         8 references
14         public string Titulo { get; set; }
15
16         4 references
17         public string Genero { get; set; }
18
19         4 references
20         public string Autor { get; set; }
21
22         7 references
23         public bool estaPrestado { get; set; }
24
25         0 references
26         public Libro()
27         {
28             Titulo = "Los libros son arte";
29             Genero = "Fantasia";
30             Autor = "Sirius Romaneconti";
31             estaPrestado = false;
32         }
33
34         5 references
35         public Libro(string titulo, string genero, string autor)
36         {
37             Titulo = titulo;
38             Genero = genero;
39             Autor = autor;
40         }
41
42         1 reference
43         public string GetLibroInfo()
44         {
45             return $"Titulo: {Titulo}, Genero: {Genero} , Autor: {Autor}" ;
46         }
47
48         0 references
49         public void SetLibro(string titulo, string genero, string autor, bool estaPrestado)
50         {
51             Titulo = titulo ;
52             Genero = genero;
53         }
54 }
```

```
PracticandoClasesPOO
PracticandoClasesPOO.Libro

8 namespace PracticandoClasesPOO
9 public class Libro
10 {
11     1 reference
12     public string GetLibroInfo()
13     {
14         return $"Titulo: {Titulo}, Genero: {Genero} , Autor: {Autor}" ;
15     }
16
17     0 references
18     public void SetLibro(string titulo, string genero, string autor, bool estaPrestado)
19     {
20         Titulo = titulo ;
21         Genero = genero;
22         Autor = autor;
23         this.estaPrestado = estaPrestado;
24     }
25
26     4 references
27     public void Prestar()
28     {
29         if (!estaPrestado)
30         {
31             estaPrestado = true;
32             Console.WriteLine($"El libro '{Titulo}' ha sido prestado.");
33         }
34         else
35         {
36             Console.WriteLine($"El libro '{Titulo}' ya está prestado.");
37         }
38     }
39
40     // Método para devolver el libro
41     2 references
42     public void Devolver()
43     {
44         if (estaPrestado)
45         {
46             estaPrestado = false;
47             Console.WriteLine($"El libro '{Titulo}' ha sido devuelto.");
48         }
49         else
50         {
51             Console.WriteLine($"El libro '{Titulo}' no está prestado.");
52         }
53     }
54
55     1 reference
56     public override string ToString()
57     {
58         string estadoPrestado = estaPrestado ? "Si" : "No";
59     }
60 }
```

```
PracticandoCLasesPOO
82
83 Libro libro1 = new Libro("El Principito", "Fantasia", "Antoine de Saint-Exupéry");
84 Libro libro2 = new Libro("Alicia en el País de las Maravillas", "Aventura", "Lewis Carroll");
85 Libro libro3 = new Libro("El Mago de Oz", "Aventura", "L. Frank Baum");
86 Libro libro4 = new Libro("El Caballero de la Armadura Oxidada", "Fábula", "Robert Fisher");
87 Libro libro5 = new Libro("Harry Potter y la piedra filosofal", "Acción", "J.K. Rowling");
88
89 List<Libro> librosList = new List<Libro> { libro1, libro2, libro3, libro4, libro5 };
90
91
92
93 int contador = 1;
94 Console.WriteLine("INFO LIBROS: ");
95
96 foreach (var libro in librosList)
97 {
98     Console.WriteLine($"Libro {contador} : {libro.GetLibroInfo()}");
99     contador++;
100 }
101
102
103
104 Console.WriteLine("\n");
105
106 Console.WriteLine("Registro de Prestamos y Devoluciones: ");
107 try
108 {
109     libro1.Prestar();
110     libro2.Prestar();
111     libro1.Devolver();
112     libro3.Prestar();
113     libro4.Prestar();
114     libro4.Devolver();
115 } catch (Exception ex)
116 {
117     Console.WriteLine($"Hubo un error al prestar o devolver un libro: {ex.Message}");
118 }
119
120 Console.WriteLine("\n");
121
122 int contador2 = 1;
123 Console.WriteLine("Estados de Prestamos: ");
124
125 foreach (var libro in librosList)
126 {
127     Console.WriteLine($"Libro {contador2} : {libro.ToString()}");
128     contador2++;
129 }
130
131
132
133
134
135
136
```

Microsoft Visual Studio Debug X + v - □ X

INFO LIBROS:  
Libro 1 : Titulo: El Principito, Genero: Fantasia , Autor: Antoine de Saint-Exupéry  
Libro 2 : Titulo: Alicia en el País de las Maravillas, Genero: Aventura , Autor: Lewis Carroll  
Libro 3 : Titulo: El Mago de Oz, Genero: Aventura , Autor: L. Frank Baum  
Libro 4 : Titulo: El Caballero de la Armadura Oxidada, Genero: Fábula , Autor: Robert Fisher  
Libro 5 : Titulo: Harry Potter y la piedra filosofal, Genero: Acción , Autor: J.K. Rowling

Registro de Prestamos y Devoluciones:  
El libro 'El Principito' ha sido prestado.  
El libro 'Alicia en el País de las Maravillas' ha sido prestado.  
El libro 'El Principito' ha sido devuelto.  
El libro 'El Mago de Oz' ha sido prestado.  
El libro 'El Caballero de la Armadura Oxidada' ha sido prestado.  
El libro 'El Caballero de la Armadura Oxidada' ha sido devuelto.

Estados de Prestamos:  
Libro 1 : Titulo: El Principito, ¿Está prestado?: No  
Libro 2 : Titulo: Alicia en el País de las Maravillas, ¿Está prestado?: Sí  
Libro 3 : Titulo: El Mago de Oz, ¿Está prestado?: Sí  
Libro 4 : Titulo: El Caballero de la Armadura Oxidada, ¿Está prestado?: No  
Libro 5 : Titulo: Harry Potter y la piedra filosofal, ¿Está prestado?: No

C:\Users\Emanu\source\repos\Class\bin\Debug\net8.0\PracticandoCLasesPOO.exe (process 29752) 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 . . .|



5. Crea una clase Fracción con métodos para sumar, restar, multiplicar y dividir fracciones.

```
PracticandoClasesPOO
PracticandoClasesPOO.Fraccion

7 namespace PracticandoClasesPOO
8 {
9     32 references
10     public class Fraccion
11     {
12         15 references
13         public int Numerador { get; set; }
14         18 references
15         public int Denominador { get; set; }
16
17         0 references
18         public Fraccion()
19         {
20             Numerador = 0;
21             Denominador = 1;
22         }
23
24         7 references
25         public Fraccion(int numerador, int denominador)
26         {
27             if (denominador == 0)
28             {
29                 throw new ArgumentException("Lider El denominador no puede ser 0.");
30             }
31
32             Numerador = numerador;
33             Denominador = denominador;
34         }
35
36         1 reference
37         public static Fraccion Sumar(Fraccion f1, Fraccion f2)
38         {
39             int numerador = f1.Numerador * f2.Denominador + f2.Numerador * f1.Denominador;
40             int denominador = f1.Denominador * f2.Denominador;
41             return new Fraccion(numerador, denominador);
42         }
43
44         1 reference
45         public static Fraccion Restar(Fraccion f1, Fraccion f2)
46         {
47             int numerador = f1.Numerador * f2.Denominador - f2.Numerador * f1.Denominador;
48             int denominador = f1.Denominador * f2.Denominador;
49             return new Fraccion(numerador, denominador);
50         }
51
52         1 reference
53         public static Fraccion Multiplicar(Fraccion f1, Fraccion f2)
54         {
55             int numerador = f1.Numerador * f2.Numerador;
56             int denominador = f1.Denominador * f2.Denominador;
57             return new Fraccion(numerador, denominador);
58         }
59     }
60 }
```

```
46 public static Fraccion Multiplicar(Fraccion f1, Fraccion f2)
47 {
48     int numerador = f1.Numerador * f2.Numerador;
49     int denominador = f1.Denominador * f2.Denominador;
50     return new Fraccion(numerador, denominador);
51 }
52
53 1 reference
54 public static Fraccion Dividir(Fraccion f1, Fraccion f2)
55 {
56     if (f2.Numerador == 0)
57     {
58         throw new DivideByZeroException("Tonto No se puede dividir entre cero.");
59     }
60     int numerador = f1.Numerador * f2.Denominador;
61     int denominador = f1.Denominador * f2.Numerador;
62     return new Fraccion(numerador, denominador);
63 }
64 }
65 }
```

```

138
139 Fraccion fraccion1 = new Fraccion(2, 3);
140 Fraccion fraccion2 = new Fraccion(4, 5);
141 Fraccion fraccion3 = new Fraccion(6, 7);
142
143
144 Fraccion resultadoSuma = Fraccion.Sumar(fraccion1, fraccion2);
145 Console.WriteLine($"Suma: {resultadoSuma.Numerador}/{resultadoSuma.Denominador}");
146
147
148 Fraccion resultadoResta = Fraccion.Restar(fraccion1, fraccion2);
149 Console.WriteLine($"Resta: {resultadoResta.Numerador}/{resultadoResta.Denominador}");
150
151
152 Fraccion resultadoMultiplicacion = Fraccion.Multiplicar(fraccion1, fraccion3);
153 Console.WriteLine($"Multiplicación: {resultadoMultiplicacion.Numerador}/{resultadoMultiplicacion.Denominador}");
154
155
156 Fraccion resultadoDivision = Fraccion.Dividir(fraccion3, fraccion2);
157 Console.WriteLine($"División: {resultadoDivision.Numerador}/{resultadoDivision.Denominador}");
158
159

```

Microsoft Visual Studio Debug X + v - □ X

Suma: 22/15  
 Resta: -2/15  
 Multiplicación: 12/21  
 División: 30/28

C:\Users\Emanu\source\repos\Class\bin\Debug\net8.0\PracticandoCLasesPOO.exe (process 14924) 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 . . .