TECNOLÓGICO NACIONAL DE MÉXICO

SUBDIRECCION ACADÉMICA

DEPARTAMENTO DE SISTEMAS Y COMPUTACIÓN

SEMESTRE ENERO-JUNIO 2020

ING. SISTEMAS COMPUTACIONALES.

MATERIA: Patrones de Diseño

Guzmán Martínez Jesús Arturo 15210525

ACTIVIDAD: Practica 6

Estrategia

ASESOR: Martha Elena Pulido

Fecha de Entrega: 02 de Abril 2020

CODIGO

```
class Usuario
    {
       private string Profesor;
       private string Alumno;
       private string Socio;


public class DevolucionSinPrestamoException : System.Exception
{
       double total_prestado;
       double total_devuelto;
    public DevolucionSinPrestamoException() : base() { }
    public      DevolucionSinPrestamoException(double      total_prestado,      double
total_devuelto) : base()
       {
              this.total_prestado = total_prestado;
              this.total_devuelto = total_devuelto;
       }
public class Libro
{
       // Propiedades
       private List<double> prestamos;
       private List<double> devoluciones;
       private double total_prestado;
       private double total_devuelto;


public Libro()
       {
              // En el constructor default todo vale 0 y vale


              prestamos = new List<double>();
              devoluciones = new List<double>();
```

```
            total_prestado = 0.0;
            total_devuelto = 0.0;
      }
public Libro(List<double> prestamos, List<double> devoluciones)
      {

            // total_prestado >= total_devuelto
            // Asi que tenemos que chequearlo nosotros

            this.prestamos = new List<double>(prestamos);
            total_prestado = SumarLista(GetPrestamos());

            this.devoluciones = new List<double>(devoluciones);
            total_devuelto = SumarLista(GetDevoluciones());

            if (total_prestado < total_devuelto)
            {
                  throw      new      DevolucionSinPrestamoException(total_prestado,
total_devuelto);
            }
      }
public Libro(Libro libro)
      {

            double total_prestado_aux = SumarLista(libro.GetPrestamos());
            double total_devuelto_aux = SumarLista(libro.GetDevoluciones());

            if (total_prestado_aux < total_devuelto_aux)
            {
                  throw    new    DevolucionSinPrestamoException(total_prestado_aux,
total_devuelto_aux);
            }
```

```
            this.prestamos = new List<double>(libro.GetPrestamos());
            this.devoluciones = new List<double>(libro.GetDevoluciones());
            total_devuelto = total_devuelto_aux;
            total_prestado = total_prestado_aux;
      }

      //Getters y Setters por cada propiedad
      public List<double> GetPrestamos()
      {

            return new List<double>(prestamos);
      }

      public void SetPrestamos(List<double> prestamos)
      {



            double total_prestado_aux = SumarLista(prestamos);
            if (total_prestado_aux < total_devuelto)
            {
                  throw    new    DevolucionSinPrestamoException(total_prestado_aux,
total_devuelto);
            }

            this.prestamos = new List<double>(prestamos);
            total_prestado = total_prestado_aux;
      }

      public List<double> GetDevoluciones()
      {
                        return new List<double>(devoluciones);
```

```
	}

	public void SetDevoluciones(List<double> devoluciones)
	{
		double total_devuelto_aux = SumarLista(devoluciones);
		if (total_prestado < total_devuelto_aux)
		{

			throw	new	DevolucionSinPrestamoException(total_prestado,
total_devuelto_aux);
		}

		this.devoluciones = new List<double>(devoluciones);
		total_devuelto = total_devuelto_aux;
	}




	 static void Main(string[] args)
	 {
	int[] librosSeleccionados = int[cant];
		int cant=0;
		int val;
		int menu;

		Console.WriteLine("	Bienvenido a la biblioteca	");
		Console.WriteLine("Cuántos usuarios solicitan préstamo de libros");
		cant = int.Parse(Console.ReadLine());

		Console.WriteLine();
```

```
            string[] usuarios = new string[cant];
            int[] tel = new int[cant];

            for (int i = 0; i < cant; i++)//Iniciamos un ciclo for para que se ingrese
la cantidad de usuarios que se requieran
            {
                Console.WriteLine("Ingrese nombre de usuario " + (i + 1));
                usuarios[i] = Console.ReadLine();
                while(!int.TryParse(Console.ReadLine(), out val))
                Console.WriteLine("Ingrese número de teléfono de " + usuarios[i]);
                tel[i] = val;
                Console.Clear();

                Console.WriteLine("Elije una opción de libro para " + usuarios[i] +
                    "\n1. El Alquimista - Paulo Cohello" +
                    "\n2. Las aventuras de Tom Sawyer - Mark Twain" +
                    "\n3. El Castillo - Kafka" +
                    "\n4. La guerra y la paz - Leon Tolstoi" +
                    "\n5. La ciudad y los perros - Mario Vargas LLosa" +
                    "\n6. El Extranjero - Albert Camus" +
                    "\n7. Crimen y castigo - Fedor Dostoyevski" +
                    "\n8. Diario de Ana Frank - Ana Frank\n");

                    menu = Convert.ToInt32(Console.ReadLine());
                do
                {
                    switch (menu)
                    {
                        case 1:
                            Console.WriteLine("Su elección es: \n" + menu);
                            break;
                        case 2:
```

```
                        Console.WriteLine("Su elección es: \n" + menu);
                        break;
                    case 3:
                        Console.WriteLine("Su elección es: \n" + menu);
                        break;
                    case 4:
                        Console.WriteLine("Su elección es:\n" + menu);
                        break;
                    case 5:
                        Console.WriteLine("Su elección es:\n" + menu);
                        break;
                    case 6:
                        Console.WriteLine("Su elección es:\n" + menu);
                        break;
                    case 7:
                        Console.WriteLine("Su elección es: \n" + menu);
                        break;
                    case 8:
                        Console.WriteLine("Su elección es:\n" + menu);
                        break;
                    default:
                        Console.WriteLine("No es una opción correcta\n");
                        break;
                }
                librosSeleccionados[i] = menu;
                menu = Convert.ToInt32(Console.ReadLine());

if (!librosSeleccionados.Contains(menu))
{
      do
      {
            //...
```

```
        } while (Expresion);


        librosSeleccionados[i] = menu;
}
else
{
        Console.WriteLine("El libro ya a sido seleccionado");
}



                } while (menu>9);


            }
            Console.ReadKey();
```

Esta practica se me hizo un poco difícil de realizar, mas que nada por hacerle cambios al mismo objeto sin cambiar al objeto, dependiendo la situación solicitada.

Para entender un poco mas tuve que investigar un poco mas y ver ejemplos de implementación

El patrón Strategy tiene como objetivo adaptar el comportamiento y los algoritmos de un objeto en función de una necesidad sin cambiar las interacciones de este objeto