

```
1 namespace ResumenFunciones
2 {
3     public enum ColorType
4     {
5         BLUE,
6         WHITE,
7         BLACK
8     }
9
10    public class Dolphin
11    {
12        // SECCION 1: Atributos.
13
14        private double _size;
15        private double _life;
16        private double _lifeCapacity;
17
18        private string _name = "";
19        private ColorType _Color;
20
21
22        // SECCION 2: Constructores.
23        public Dolphin()
24        {
25
26        }
27
28        public Dolphin(double currentLife, double maxLifeCapacity)
29        {
30            _life = currentLife;
31            _lifeCapacity = maxLifeCapacity;
32        }
33
34        public Dolphin(double currentLife, double maxLifeCapacity,
35                        double size)
36        {
37            _life = currentLife;
38            _lifeCapacity = maxLifeCapacity;
39            _size = size;
40        }
41
42        // SECCION 3: Getters & Setters.
43        public double GetLife()
44        {
45            return _life;
46        }
47
48        public void SetLife(int value)
49        {
50            _life = value;
51        }
52    }
```

```
53      /*
54          SETTERS CON VALIDACION DE PARAMETROS.
55
56          CASO 1: CLAMPEAR/SATURAR, llevamos los valores a sus límites posibles.
57          CASO 2: Comprobamos que los valores son CORRECTOS => establecemos el VALUE.
58          CASO 3: Comprobamos que los valores son INCORRECTOS => Lanzamos un error.
59      */
60      public void SetLifeCase1(int value)
61      {
62          if (value < 0.0)
63              _life = 0;
64          else if (value > _lifeCapacity)
65              _life = _lifeCapacity;
66          else
67              _life = value;
68      }
69
70      public void SetLifeCase2(int value)
71      {
72          if (0.0 <= value && value <= _lifeCapacity)
73              _life = value;
74          // else
75          // el programa no hace nada
76      }
77
78      public void SetLifeCase3(int value)
79      {
80          if (value < 0.0 || value > _lifeCapacity)
81              throw new Exception("Error de validacion de parametros");
82          _life = value;
83      }
84
85
86      // SECCION 4: Metodos.
87
88      // FUNCION que devuelve el porcentaje de vida.
89      // Tantos por 1.
90      public double GetLifePercentOne()
91      {
92          return (_life / _lifeCapacity);
93      }
94
95      // Tantos por 100.
96      public double GetLifePercent()
97      {
98          return (_life / _lifeCapacity) * 100;
99      }
100 }
101 }
```