

# Clase 8

Martes, 26 Septiembre 2023

---

[Indice](#)

## Getters

- Funciones creadas explícitamente para 'recoger' o conseguir el valor de un atributo. Esto sucede porque es muy posible que otros programadores quieran utilizar los valores que hemos creado en nuestras funciones, a las cuales habitualmente no tendrán acceso directo, sino a través de los getters.

```
,
public class Dolphin
{
    private double life;

    public double GetLife()
    {
        return life;
    }
},
```

Habitualmente *no reciben ningún parámetro y siempre devuelven un valor*

## Setters

- Funciones creadas explícitamente para 'establecer' el valor de un atributo. Se debe realizar la *validación* del parámetro, para que realice un correcto funcionamiento de nuestro programa.

```
,
public class Dolphin
{
    private double life;

    public void SetLife(double value)
    {
        this.life = value;
    }
},
```

Habitualmente *no devuelven nada y siempre reciben un parámetro*

- La validación de los setters puede implicar tres sistemas diferentes:
  - 1.- "Clamppear" o "Saturar" los parámetros de entrada. Esto significa llevar los parámetros a los niveles máximos y mínimos (Clamppear) o simplemente a los niveles máximos (Saturar).
  - 2.- Comprobar que los valores son correctos y en este caso, el programa funciona normalmente
  - 3.- Darse cuenta que los valores son incorrectos y en este caso, lanzar un error de advertencia.

---

1.- "Clamppear" o "Saturar" los parámetros de entrada.

```
,
public class Dolphin
{
    private double life;

    public void SetLife(double value)
    {
        if (value < 0)
            this.life = 0;
        else if (value > 100)
            this.life = 100;
        else
            this.life = value;
    }
},
```

```
    }  
    }  
    ,
```

2.- Comprobar que los valores son correctos y en este caso, el programa funciona normalmente

```
,  
public class Dolphin  
{  
    private double life;  
  
    public void SetLife(double value)  
    {  
        if (value > 0 && value < 100)  
            this.life = value;  
        // else el programa no realiza ninguna acción  
    }  
}  
,
```

3.- Darse cuenta que los valores son incorrectos y en este caso, lanzar un error de advertencia.

```
,  
public class Dolphin  
{  
    private double life;  
  
    public void SetLife(double value)  
    {  
        if (value < 0 || value > 100)  
            throw new Exception("Error de validacion de parametros");  
        this.life = value  
    }  
}  
,
```