

Manual de buenas prácticas de programación

- **Nombres descriptivos:** debes usar nombres de variables, funciones o clases que tengan sentido para facilitar su lectura a la hora de leer el código.

Ejemplo:

```
int p = 3; MAL
```

```
int precioEnEuros = 3;
```

- **Indentación y espaciado:** se debe aplicar una indentación consistente que permita mejorar la legibilidad del código, además, se debe emplear espacios de manera consistente en vez de tabulaciones.

Ejemplo:

```
if(precioEnEuros > 1600){  
    Console.WriteLine("Buen sueldo");  
}  
else{  
    Console.WriteLine("Mal sueldo");  
} MAL
```

```
if(precioEnEuros > 1600){  
    Console.WriteLine("Buen sueldo");  
}  
else {  
    Console.WriteLine("Mal sueldo");  
}
```

- **Comentarios claros:** Se deben poner comentarios cuando resulten útiles para explicar partes de código complicadas, evitando comentarios obvios e innecesarios.

Ejemplo:

```
public void TablaMultiplicar(int num){
    // Bucle de 0 a 10
    for(int i = 0; i < 10 ; i++){
        //Imprimimos i por el número que nos pasan por parámetro
        Console.WriteLine(i * num);
    }
} MAL
```

```
// Función que imprime la tabla de multiplicar de un número
Public void TablaMultiplicar(int num){
    // Bucle de 0 a 10
    for(int i = 0; i < 10 ; i++){
        //Imprimimos i por el número que nos pasan por parámetro
        Console.WriteLine(i * num);
    }
}
```

- **Modularizar el código:** se debe dividir el código en trozos más pequeños para tener una estructura más simplificada que no dificulte la lectura del mismo.
- **Evitar la duplicación de código:** no se debe repetir código innecesario, lo mejor es hacer una función con un trozo de código que puedas reutilizar en diferentes sitios.
- **Manejo de errores:** debes controlar los errores implementando excepciones concretas que permita identificar los problemas de código rápidamente.