Antes de que empecéis a picar como locos, me gustaría que os miréis esto en un minuto por el bien de todos.

Lo primero de todo, quiero que os deis cuenta que esto no es una práctica de TP, lo que tú escribes lo va a leer mucha más gente, y se va a modificar incluso más veces aún.

Así que como en cualquier proyecto de software medianamente decente, me gustaría que siguiéramos unos estándares para evitar perder el tiempo y los ojos en código que no se entiende.

Para ello os propongo seguir estas reglas:

**Toda variable debe tener la primera letra de cada palabra que compone su nombre en mayúscula.** Salvo la primera de todas.

boolean esBuenNombre;

**El nombre de los métodos debe ser o contener un verbo.**

**No programar en spanglish.** Me pareció ver el nombre de una clase en inglés. Entré a verla y los nombres estaban en español. Un ejemplo muy parecido es éste, cogido de nuestro propio código:

this.estado = SignalState.ROJO;

Si tenemos una variable estado, un estado que es rojo, entonces ¿por qué el enumerado en vez de EstadoSenal se llama SignalState?

La mezcla de idiomas en el mismo contexto no es buena. Además de que puede haber gente que no se maneje muy bien con el inglés (mal hecho a estas alturas).

**Nombrar con sentido.** Como tú has escrito el código, sabes que esa variable s es un argumento spliteado y en uppercase, pero el resto de la clase quizá no.

**Cada método tiene que tener, como máximo, una pantalla de extensión.**

**Cada clase debe tener una sola responsabilidad, y un motivo para cambiar.**

**No poner comentarios redundantes.** Este código también es nuestro:

//Para modificar el estado de la señal

public void setState(){ … }

Además viola el punto 3. Sería setEstado().

El set y el get se quedarían en inglés como excepción por ser un estándar de java, y no tener nombre en español.

El único comentario bueno es el que no se tiene que escribir.

**NO DEJÉIS BLOQUES NI CÓDIGO COMENTADO.** El típico síndrome de diógenes del programador. No comentéis código que no funciona, **reescribidlo**. Además de que el siguiente que vaya a tocarlo no sabrá si es suficientemente importante como para quitarlo, y puede sobrevivir hasta el fin de los tiempos.

Excepto Comentarios necesarios sobre algo que no pueda entender alguien más.

**Intentad que el código sea autoexplicativo.** Esto es bueno para poder entender lo que está escrito con un simple vistazo (además de ser más bonito).

Existe además un riesgo extra, y es que el código se mantiene (modifica), pero de los comentarios rara vez nos acordamos. Seguro que esa segunda vez que has modificado algo ya no encaja totalmente con el comentario que tenías arriba. Un ejemplo, qué prefieres ver, esto:

// Comprobar si un empleado reúne los requisitos.[[1]](#footnote-0)

if ((empleado.indicador & INDICADOR\_HORA) && (empleado.edad > 65))

o esto:

if (empleado.reuneLosRequisitos())

No se tarda nada en darle expresión al código, la mayoría de las veces sólo hay que hacer un método privado.

**NO PONER LLAVE DE ABERTURA EN LA SIGUIENTE LÍNEA.** Como alguien me ponga esto en java, lo mato.

public void modificarSenal (boolean estadoNuevo) // ASÍ NOOOO

{

this.estado = estadoNuevo;

}

**Seguid una estructura en la clase.** Poner primero los atributos, luego los constructores, luego los métodos públicos y abajo del todo los métodos privados.

Si tenéis alguna duda sobre el estilo de código, podéis ver el standard de java [aquí](http://www.oracle.com/technetwork/java/javase/documentation/codeconvtoc-136057.html).

Para cualquier duda me podéis preguntar. Héctor Laria

1. Ejemplo sacado de Clean code, de Robert C.Martin [↑](#footnote-ref-0)