

BLOQUE 1 : NOMBRES

-Para tener un código de lectura rápida, sencilla y fácil de leer necesitamos usar **nombres con significado** tanto en variables, clases, funciones, métodos... etc.

-Al mismo tiempo estos nombres tienen que ser **fácil de pronunciar**

-**Usar nombres que puedan buscarse**, por ejemplo si tenemos constantes, las podemos extraer fácilmente con un nombre que tenga significado.

-Los nombres de las clases deben ser **nombres** mientras que las funciones o métodos deben ser **verbos**.

-**No usar sinónimos y ponerles la misma funcionalidad** a otras funciones o conceptos ya creados.

BLOQUE 2 : FUNCIONES

-**Funciones pequeñas:** en el libro se recomienda que sean muy cortas (3-4 líneas aprox.) pero esto es inviable en la mayoría de casos así que un límite más razonable serían unas **50-60 líneas aprox.**

-**Deben realizar solo una cosa.** En caso contrario es mejor dividir las.

-**No abusar de los switch/when.** Se puede usar poliformismo para ocultar esto.

-Una función no debe tener mas de **3 argumentos** y lo ideal es que tenga **0**.

-Llevado de la mano con lo anterior debemos evitar los **“flags arguments”** que suelen ser booleanos.

-Una función debe hacer algo o devolver algo **nunca ambas cosas**.

-**No repetir funciones (básicamente no repetir código).**

BLOQUE 3 : COMENTARIOS

-Los comentarios **no compilan** porque lo que si el código asociado se modifica el comentario se queda obsoleto.

-En vez de usar comentarios para explicar parte del código, intentar mejorar el código para que se explique a si mismo.

-Cabe recalcar que los comentarios no son inútiles. En muchas ocasiones (mas bien cuando trabajas con algún framework o API) vas a necesitar usarlos.

-Los comentarios dicen **que hace el código** no como lo hace