

La procedencia del código es un ejercicio hecho en clase de M3.

Los archivos están la pertinente carpeta "ex1".

Para resumir el extenso tema de la *refactorización*, puedo concluir con que *refactorizar* consiste en tener un código más limpio y fácil de entender por todas las personas, más allá de poner unos cuantos comentarios.

Es importante entender que el código que iremos generando será por partes, de forma que en cada parte nos aseguraremos que a parte de que funcione, el código sea legible y entendible, para futuros mantenimientos o ampliaciones.

Por lo tanto no *refactorizar* no es añadir más funcionalidades, ni tampoco es necesariamente más eficiente.

"Ex1\_sin\_refactoring" es el ejercicio sin aplicar ninguna regla de *refactorización* ni reusabilidad, por lo tanto todo el código esa en única función y dificulta su futura ampliación de funcionalidades.

"Ex1\_con\_refactoring" es el mismo ejercicio anterior, solamente que se le ha aplicado una simple regla de *refactoring*, al haber sacado el código en las funciones, aunque pueda ser algo menos eficiente en tiempos de ejecución, es mucho mejor para entender el código y además permite ampliar las diferentes funciones o simplemente corregir posibles errores.

Que precisamente hablando de errores, uno de los pilares importantes del *refactoring* es la prueba unitaria, ya que con ello puedes probar todas las líneas de código que desees y comprobar que hacen lo esperado.