1. ¿En qué consiste el principio de responsabilidad única?

El principio de responsabilidad única u SRP (siglas del inglés (Single Responsibility Principle) en ingeniería de software establece que cada módulo o clase debe tener responsabilidad sobre una sola parte de la funcionalidad proporcionada por el software y esta responsabilidad debe estar encapsulada en su totalidad por la clase. Todos sus servicios deben estar estrechamente alineados con esa responsabilidad.

Una clase debe tener solo una razón para cambiar.

En programación orientada a objetos, se suele definir como principio de diseño que cada clase debe tener una única responsabilidad, y que esta debe estar contenida únicamente en la clase. Así:

- Una clase debería tener sólo una razón para cambiar
- Cada responsabilidad es el eje del cambio
- Para contener la propagación del cambio, debemos separar las responsabilidades.
- Si una clase asume más de una responsabilidad, será más sensible al cambio.
- Si una clase asume más de una responsabilidad, las responsabilidades se acoplan.

¿Cuál es su propósito?

Que cada clase tenga su propia tarea, cumpliendo así el principio de la única responsabilidad y dándonos una mayor flexibilidad para extender los módulos.

¿Qué características tiene un buen código o código limpio?

El lenguaje con el que se escribió el código debería parecer que fue hecho para el problema. No es el lenguaje lo que hace parecer simple a un problema, sino que es el desarrollador que hace que el lenguaje parezca simple.

El código no debe ser redundante.

Puede ser extendido fácilmente por cualquier otro desarrollador.

Debe tener dependencias mínimas

Mientras más dependencias tenga, más difícil va a ser de mantener y cambiar en el futuro.

El código debería ser mínimo. Tanto las clases como los métodos deberían ser cortos, preferentemente con pocas líneas de código. Debe estar bien dividido. Mientras mejor dividamos el código, más fácil se vuelve leerlo.

El proyecto debe tener pruebas unitarias.

El código debe ser expresivo, la expresividad del código significa que tiene nombres significativos. Estos nombres deben expresar la intención. No tienen que resultar engañosos. Tienen que ser distintivos.

En fin un buen código debería cumplir con las siguientes características:

Legible , Simple, Elegancia, Eficaz, Óptimo, con pruebas, etc.