1. El principio de responsabilidad única expone que una clase solo debe de cambiar por una razón, esto significa que de existir otra, la funcionalidad debe de ser encapuslada por otra clase. El propósito de esto es el de facilitar la metodología de programación ágil, de esta manera 2 personas pueden estar trabajando en funcionalidades distintas sin preocuparse por que esos cambios estén interfiriendo con los cambios de la otra persona.
2. En mi opinión el código limpio tiene las siguientes características:

* Organizado: Es decir, saber donde están las cosas. Esto va desde como se estructura la carpeta del proyecto hasta las convenciones de nombres de los archivos.
* Limpio: Valga la redundancia, no tener comentarios, con funcionalidad vieja o “deseada” o variables auxiliares que no son usadas.
* Estructurado: Es decir, definir un estilo de código, una convención de nombres constante y algún paradigma de programación a usar.(Funcional, Orientado a Objetos, a Componentes, Imperativo, etc)
* Regular: Debe de mantener todas estas características a lo largo de toda la base de código, sin importar la persona(s) involucrada(s) o la necesidad de lanzamiento.