**Principio de responsabilidad Única:**

Cada responsabilidad es un eje de cambio. Cuando los requisitos cambian ese cambio será manifestado a través de un cambio de responsabilidad entre clases. Si una clase asume más de una responsabilidad, entonces habría más de una razón para cambiar.

Los cambios en una responsabilidad pueden perjudicar o inhibir la capacidad de la clase para cumplir con las otras.

**Principio Abierto – Cerrado:**

Se dice que se deben diseñar módulos que nunca cambien. Cuando cambian los requisitos, se amplía el comportamiento de dichos módulos agregando código nuevo, no modificando el código antiguo que ya funciona.

**Principio Inversión de Dependencia:**

Establece que toda función que opera sobre un puntero a una clase base, debería poder operar sobre derivadas de esa clase base.

También significa que las funciones miembro virtuales que están presentes en las clases base también deben estar presentes en las clases derivadas; y deben hacer un trabajo útil.