

## Punto tres prueba de Grability

Preparado por: Camilo Andres Pulido Rodriguez  
correo: camilonosdocs@gmail.com

### 1. ¿ En que consiste el principio de responsabilidad única ?

Este principio se refiere a que un modulo , clase no debe tener mas de una razon para cambiar es decir que una clase u objeto debe tener un papel único al ser el responsable de ejecutar una actividad dentro del sistema de software.

### 2. ¿ Qué características tiene según tu opinión código limpio ?

Para mi un código limpio debe tener las siguientes características:

- Código ordenado
- Código claro para los demás programadores
- Código que sea apto para mantenimiento futuro, es decir que si hay que aplicar nuevas funcionalidades o actualizaciones no me toque hacer todo de nuevo poque este ligado a una característica que no lo permita
- Que por lo menos maneje POO
- Por ejemplo algo que profeso como lo dice el Framework Django es el don't repeat yourself con lo cual se debe tratar de no repetir o hacer cosas doble vez en código fuente que después lleguen a alargar los tiempos de desarrollo
- Que la estructura del código sea escalable y optimizada para soportar operaciones concurrentes ya que hoy en día las aplicaciones web deben pensarse para lograr soportar altos niveles de request

En este caso coloco un ejemplo con un código de un cliente que vi hace pocos días:

En si era un portal para realizar reservas, pero todo esta estructamente ligado al DOM de HTML y las operaciones se calculaban con los identificadores de los objetos DOM, clases dentro de los objetos DOM y eran trabajados por medio de JQuery , para mi esto es un mal cogido y no es nada limpio ya que cuando uno quiera modificar el diseño del portal , o modificar el css o si quiere integrar una App al portal no va a ser posible ya que esto ni es mantenible ni escalable.

En comparación a esto lo que yo haría es utilizar un framework como angularjs o ember si se requiere que sea muy dinámica la interfaz de usuario y no estoy tocando en ningún momento el dom como parte de la lógica de negocio. Ya que con el ejemplo anterior mi concepto si se quiere cambiar mucho la funcionalidad de reservas le diría al cliente que es mejor iniciar nuevamente el proyecto de ceros.