

PRUEBA TÉCNICA PARA EL BACKEND DE GRABILITY

Ernesto Soto Meléndez
CC 1.015.429.099

Bogotá D.C.
Abril 24 2017

Contents

Preguntas.....	2
Principio de responsabilidad única.....	2
“Buen” código o código limpio.....	2

Preguntas

Principio de responsabilidad única

El principio de responsabilidad única dicta que cada componente de programación debería tener una única responsabilidad. Su propósito principal es mantener un código fácil y entendible, desacoplado, y que facilite la reutilización y el mantenimiento del mismo.

De esta forma, el principio especifica que cada componente sólo debe tener una razón para cambiar. Si tiene más de una responsabilidad, las responsabilidades se acoplan y será más sensible a cambiar. Esto favorece la mantenibilidad de un software.

Además, aplicado en arquitecturas de microservicios, favorece la interoperabilidad y la escalabilidad.

“Buen” código o código limpio

Yo creo que cualquiera es capaz de escribir código funcional, es decir, que un compilador pueda entenderlo y ejecutarlo. Pero no cualquiera puede escribir código que otras personas puedan entender e interpretar.

Y con la alta rotación de personal en los equipos y en las empresas de software, especialmente en nuestro país, es necesario que cualquier nuevo integrante que llegue a un equipo pueda entender fácilmente el código para continuar con el proyecto.

Un código ordenado y descriptivo, que cualquier persona que conozca el lenguaje de programación puede entender. De esta forma, el código es mantenible, flexible, escalable y reutilizable.

Además, el código limpio igualmente debe cumplir con la calidad y asegurar las medidas de los atributos de calidad que requiera el software.

Finalmente, en un caso ideal para una compañía, los nombramientos y las estructuras del código deben ser estándar, de modo que, sin importar cuántos desarrolladores participen en un proyecto, sigue siendo mantenible y entendible.