Clean Code CODIGO LIMPIO





Agenda

- Buenas practicas para escribir código.
- · Clean Code.
- Craftmanship.
- Code review.
- Linters.





Any fool can write code that a computer can understand.

Good programmers write code that humans can
understand.

- Martin Fowler





Los 10 mandamientos de los desarrolladores en Jalasoft

- 1. Seguir un proceso iterativo e incremental.
- 2. Seguir convenciones para producir código legible (de preferencia CleanCode).
- 3. Eliminar redundancias y duplicaciones.
- 4. Uso adecuado de las estructuras de datos elegidas.
- 5. Hacer pruebas de unidad.





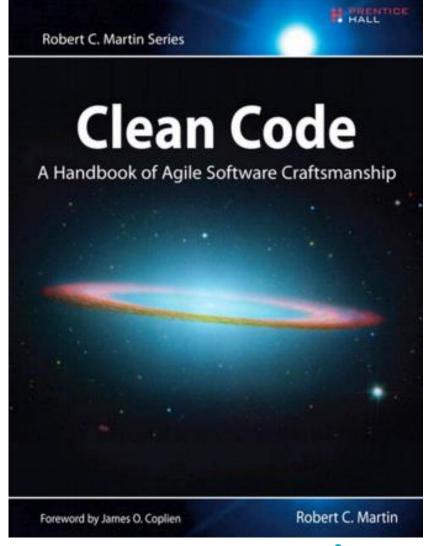
Los 10 mandamientos de los desarrolladores en Jalasoft

- 6. Refactorización constante.
- 7. Escribir código que sea fácil de extender.
- 8. Usar consistentemente manejo y recuperación de errores.
- 9. Reportar y documentar los defectos inmediatamente.
- 10. Trabajar con un ambiente de producción limpio.





CLEAN CODE







Resumen de Clean code

- Siga las convenciones estándar.
- Mantenlo simple y estúpido. Lo más simple siempre es mejor. Reduzca la complejidad tanto como sea posible.
- Regla de los boy scouts. Deje el campamento más limpio de lo que lo encontró.
- Encuentre siempre la causa raíz. Busque siempre la causa raíz de un problema.









No sólo responder al cambio,



No sólo individuos e interacciones, sino también una comunidad de profesionales No sólo colaboración de clientes, sino también asociaciones productivas Es decir, en la búsqueda de los elementos de la izquierda, hemos encontrado indispensables los elementos de la derecha. a declaración puede ser libremente copiada en cualquier forma,

sino también agregar valor constantemente

craftmanship.org/







DECEMBER 1, 2003 by JOEL SPOLSKY

Craftsmanship

■ ROCK STAR DEVELOPER, ARTICLES

Making software is not a manufacturing process. In the 1980s everyone was running around terrified that Japanese software companies were setting up "software factories" that could churn out high quality code on an assembly line. It didn't make any sense then and it doesn't make sense now. Shoving a lot of programmers into a room and lining them up in neat rows did not really help get the bug counts down.

If writing code is not assembly-line style production, what is it? Some have proposed the label *craftsmanship*. That's not quite right, either, because I don't care what you say: that dialog box in Windows that asks you how you want your help file indexed does not in any way, shape, or form resemble what any normal English speaker would refer to as "craftsmanship."

Craftmanship (Joel Spolsky)

https://www.joelonsoftware.com/2003/12/01/craftsmanship-2/

Code review

- Ayuda a verificar si se siguen las buenas practicas de codificación.
- El code review no garantiza que después de la revisión no hayan errores en la funcionalidad.
- Todo código que escribamos debería ser escrito pensando en que ese código sera revisado por otras personas y debe ser fácil de leer y entender.





Linters

- Son herramientas que automatizan la verificación de la calidad del código.
- Utilizar linters ayuda a reducir el esfuerzo de realizar code reviews, de esta forma los revisores pueden enfocarse en la lógica del código mas que en la forma.
- Es recomendable utilizar linters como parte de la integración del código fuente.





Resumen

- La principal característica del buen código es que puede ser entendido por cualquier desarrollador que tenga que mantenerlo.
- Es necesario aprender buenas practicas de codificación.
- El code review nos ayuda a reforzar el cumplimiento de las buenas practicas de codificación, esto ayuda a mantener la calidad del código.
- Es recomedable usar herramientas automatizadas para la verificación de la calidad del código, estas herramientas son conocidas como Linters.



