Principios de código limpio

Hector Vanegas Solis

Claudia Gil Sanchez

Politécnico Colombiano Jaime Isaza Cadavid

Además de esto, el código envejece, y Clean Code se encaja perfectamente en este caso.

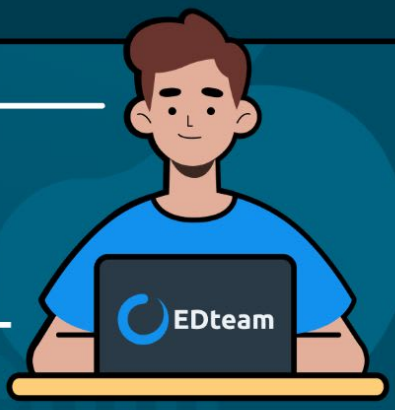
La iniciativa de los inicios de Clean Code es transformar el desarrollo web y el mantenimiento del código cada vez más fácil.

Un código con varios ajustes a lo largo de un largo tiempo, se vuelve imposible de conservar, por lo cual es mejor producir un código nuevo y no avanzar con una versión problemática.

El término clean code se atribuye al ingeniero de software Robert Cecil Martin, que lo utilizó en su libro Clean Code: Refactoring, Patterns, Testing and Techniques for Clean Codepara referirse al código limpio. Sin embargo, sus principios son mucho más antiguos y no provienen del campo de la programación. Te explicamos en qué consiste el clean code, cuáles son sus ventajas y cómo desarrollarlo.

4 CARACTERÍSTICAS DEL CÓDIGO LIMPIO

Los desarrolladores invierten **la mayor parte de su tiempo leyendo código.**



FUENTE: **Código limpio**
Robert C. Martin

1. NOMBRES QUE SE PUEDAN BUSCAR

Si nombras tus variables con una sola letra, **será difícil hacer Ctrl+f y buscar.**

```
let dayOfMonth = 4
```

2. NOMBRES CON SENTIDO

El nombre de tus variables, constantes y funciones **deben indicar claramente por qué existen y qué hacen.**

```
let paymentInUSD
```

3. ESCRIBE TU CÓDIGO COMO UN LIBRO

Tu código debe tener estructura y orden **para que sea fácil de entenderlo**, como cuando lees un libro.

4. COMENTARIOS

Debes usar **los comentarios para agregar valor al código.** No comentes nombres obvios.

Si quieres aprender más de programación mira nuestros cursos

ed.team/cursos

EDteam

1. Clean Code

Además de esto, el código envejece, y Clean Code se encaja perfectamente en este caso.

La iniciativa de los inicios de Clean Code es transformar el desarrollo web y el mantenimiento del código cada vez más sencilla.

Un código con varios ajustes a lo largo de un largo tiempo, se vuelve imposible de conservar, por lo cual es mejor producir un código nuevo y no avanzar con una versión problemática.

7 reglas principales del Clean Code

¿Cómo redactar código limpio? Por esto, su sentido despierta discusión en la sociedad de desarrolladores: lo cual unos piensan "limpio", podría ser "sucio" para los demás, en consecuencia el aseo del código termina siendo algo personal. En seguida, te presentamos ciertos inicios del código limpio bastante extendidos y que casi todos los desarrolladores piensan útiles.

1 – Los nombres son importantes

- Variable
- Función
- Parámetro
- Clase
- Método

Al definir un nombre, es necesario tener en mente dos aspectos principales:

Debe ser preciso y entregar la idea central. Es decir, debe ir directo al punto; No te preocupes por los nombres grandes. Si la función o parámetro necesita de un nombre extenso para demostrar lo que realmente representa, es lo que debes hacer.

2 – Regla del boy scout

Sabías qué? Si traemos esta regla al mundo de la programación, la tenemos la posibilidad de adaptar como dejar el código más limpio de lo cual estaba antecedente de editarlo.

3 – Debes ser el verdadero autor del código

Sabías qué? La gente está acostumbrado a pensar de forma narrativa, de esta forma que el código funciona del mismo modo. Resumiendo, para estructurar un código limpio, se necesita producir funcionalidades primordiales, claras y pequeñas. Estas tienen que ser todavía más pequeñas. Como lo hemos dicho en el primer inicio, los nombres monumentales no son un problema, empero las funcionalidades sí.

4 – DRY (Don't Repeat Yourself)

Sabías qué? Una expresión, que ha sido descrita por primera ocasión en un libro denominado The Pragmatic Programmer y se aplica a superficies de desarrollo, como:

Codificaciones.

DRY defiende que cada parte del entendimiento de un sistema debería tener representación exclusiva y ser plenamente independiente de ambigüedades.

5 – Comentar solamente lo necesario

Sabías qué? Este comienzo asegura que los comentarios tienen la posibilidad de hacerse; no obstante, dichos tienen que ser necesarios. Según Uncle Bob, los comentarios mienten; y esto tiene una especificación lógica.

Entonces, ya sabes; si vas a comentar el código, que sea únicamente lo primordial y que sea inspeccionado en grupo con la versión del código que lo sigue.

6 – Tratamiento de errores

Sabías qué? Hay una frase del autor Michael Feathers, bastante conocida en el área de desarrollo web, que afirma que las cosas pueden salir más; pero cuando esto ocurre, los programadores son los responsables por garantizar que el código continúe realizando lo que necesita.

Es decir, tratar las excepciones de forma correcta es un gran paso para el programador en desarrollo.

7 – Tests limpios

Sabías qué?

- Fast: El examen debería ser veloz, permitiendo que sea llevado a cabo muchísimas veces y en cualquier instante
- Repeatable: debería permitir la repetición del examen, muchísimas veces y en diferentes ambientes
- Self-Validation: Los exámenes bien escritos retornan con las respuestas true o false para que el error no sea personal
- Timely: Los exámenes tienen que continuar estrictamente el criterio de puntualidad. Además de esto, lo ideal es que sean escritos previamente del propio código, puesto que previene que sea bastante complejo para hacer el examen.

Por todo ello, el código limpio es bastante simple de conservar y muestra las próximas características:

Las clases y los procedimientos son predecibles, funcionan como se espera y son de ingreso público por medio de API (interfaces) bien documentadas.

Los resultados positivos de esta clase de programación son obvias: el clean code se vuelve sin dependencia del creador que lo ha desarrollado. En inicio, cualquier programador puede laborar con él, lo cual previene inconvenientes como los que conlleva el código heredado.

Para tener en cuenta
Anterior a redactar un comentario se debe pensar si existe otra forma de expresarse en el código. Una vez que haya que aumentar un comentario, se debe aseverarse que este describa el código que lo circunda. Es preferible describir nuestra intención en el código Anteriormente que recurrir a un comentario. Algunas veces se puede producir una funcionalidad o una variable que dicte lo mismo que el comentario.

4 características del código limpio. (n.d.). ..Ed.Team. Retrieved November 17, 2021, from https://ed.team/comunidad/4-caracteristicas-del-codigo-limpio.

Clean code: principios, ventajas y ejemplos. (n.d.). Ionos.Es. Retrieved November 17, 2021, from https://www.ionos.es/digitalguide/paginas-web/desarrollo-web/clean-code-que-es-el-codigo-limpio/.

Cómo conseguir un código limpio. (n.d.). Sdos.Es. Retrieved November 17, 2021, from https://www.sdos.es/blog/como-conseguir-un-codigo-limpio.

(México, H. (2020, April 14). Clean Code: Código Limpio, ¿qué es? ¿cómo funciona? Hostgator.mx. https://www.hostgator.mx/blog/clean-code-codigo-limpio/.