

EVAGDstacruztf



Página Principal ► Institutos de Educación Secundaria (CIFP, EASD, IES) ► CIFP César Manrique ► Informática ►  
Turno de Mañana ► DEW\_B ► UD2. Estructura del lenguaje JavaScript ► U02-C01: Introducción al "Code Clean"

# Introducción al "Code Clean"

## Qué es el código limpio según seis de las grandes figuras del software

**Bjarne Stroustrup, inventor de C++ y autor de The C++ Programming Language.**

*Me gusta que el código sea elegante y eficiente. La lógica debe ser sencilla para evitar que los errores se escondan u oculten con facilidad, las dependencias mínimas para facilitar el mantenimiento, la gestión de errores completa de acuerdo a una estrategia definida, y el rendimiento cercano a lo óptimo para no tentar a la gente a que ensucie el código tratando de llevar a cabo optimizaciones innecesarias.*

*El buen código es aquel que hace una única cosa y la hace bien.*

**Grady Booch, autor de Object Oriented Analysis and Design with Applications**

*El código limpio es simple y directo, se lee como la prosa bien escrita. El código limpio nunca oscurece u oculta la intención del diseñador sino que debe contener abstracciones nítidas y líneas de control directas y legibles.*

**“Big” Dave Thomas, fundador de OTI, padrino de Eclipse strategy.**

*Un código limpio puede ser leído y mejorado por otro programador además del autor original. Tiene pruebas de unidad y aceptación. Tiene nombres con significado. Proporciona una manera, en vez de muchas maneras para hacer una cosa.*

*Tiene mínimas dependencias, que son explícitamente definidas y proporciona una clara y mínima API.*

**Michael Feathers, autor de Working Effectively with Legacy Code.**

*Podría listar todas las cualidades que identifiqué en un código limpio, pero hay una cualidad general que conduce a todas ellas.*

*Un código limpio siempre parece que fue escrito por alguien que se preocupa.*

*No hay nada obvio que puedas hacer para hacerlo mejor. Todas esas cosas ya fueron pensadas por el autor del código, y si intentas imaginar mejoras, no las encontrarás, seguirás delante de código hecho por alguien que se preocupa profundamente del arte del propio código.*

**Ron Jeffries, autor de Extreme Programming Installed y Extreme Programming Adventures in C#.**

*En los últimos años siempre comienzo, y suelo terminar, con las reglas de código simple de Beck.*

*En orden de prioridad, el código simple:*

- Ejecuta todas las pruebas.
- No contiene duplicidad.

- Expresa todas las ideas de diseño que están en el sistema.
- Minimiza el número de entidades como clases, métodos, funciones y similares.

*Destaco la importancia de la no duplicidad. Cuando la misma cosa es hecha una y otra vez, es señal de que la idea que teníamos en mente no está bien representada en el código.*

*Utilizar nombres con significado también es muy importante y suelo cambiar los nombres de las cosas varias veces hasta dar con el adecuado.*

*Por último y no menos importante, utilizar el concepto de **abstracción** adecuadamente para resolver problemas que nos encontramos de manera recurrente.*

**Ward Cunningham**, inventor de Wiki, inventor de Fit, coinventor de eXtreme Programming. Fuerza motriz detrás de Design Patterns. Líder de pensamiento de SmallTalk y Orientación a Objetos. El padrino de todos aquellos que se preocupan por el código.

*Sabes que estás trabajando con código limpio cuando cada rutina que lees resulta ser más o menos lo que esperabas. Puedes decir que ese código es limpio y bonito cuando el código hace que parezca que el lenguaje se hizo específicamente para resolver el problema.*

##

Este artículo está basado en el libro Clean code: A Handbook of Agile Software Craftsmanship (Robert C. Martin)

##

## Las 9 cualidades del código limpio

- <http://www.dosideas.com/noticias/desarrollo-de-software/965-las-9-cualidades-del-codigo-limpio.html>

## 12 ideas de la filosofía Clean Code que no pueden faltar en tu código

- <http://www.genbetadev.com/metodologias-de-programacion/12-ideas-de-la-filosofia-clean-que-no-pueden-faltar-en-tu-codigo>

## Ver Presentación de Clean Code

Última modificación: miércoles, 20 de septiembre de 2017, 11:46

### NAVEGACIÓN



Página Principal

■ Área personal

Páginas del sitio

Curso actual

DEW\_B

Participantes

Insignias

General

UD1. Arquitecturas y lenguajes de programación en...

UD2. Estructura del lenguaje JavaScript

 **U02-C01: Introducción al "Code Clean"**

 U02-C02: Fundamentos de JavaScript

 U02-C03: Diferencias entre var, let y const

Mis cursos