

# Clean Code

---

## 5 -Artesanía, La mejora constante

Profesionalismo, pragmatismo y orgullo.



Para TrainingIT

**Por Alberto Basalo**



## 5 -Artesanía, La mejora constante

### Profesionalismo, pragmatismo y orgullo.

*"Primero hazlo, luego hazlo bien, luego hazlo mejor."*

--  **Addy Osmani.**

Imprime esta frase y cuélgala en un sitio bien visible. Contiene tres enseñanzas, todas rodeando al verbo imperativo **hazlo**. Porque esa es la base fundamental del trabajo, **hacer algo productivo**, algo que aporte valor.

Y esa es la primera enseñanza, **haz las cosas**, evita la parálisis por análisis, evita la procrastinación, las excusas. Siempre hay algo que hacer.. Primero hazlo.



### Hacer las cosas bien

Así que dan por hecho que no hacemos las cosas bien a la primera. Al menos en mi caso es rotundamente cierto. Ni siquiera a la segunda suelo acertar. Y tampoco me vienen nadie infalible a la mente... así que **es probable que tú tampoco hagas las cosas bien a la primera**.

Hacer las cosas bien implica que hay un criterio para validar lo hecho. **¿Qué está bien? ¿Quién dice que algo está bien?** Coloquialmente se dice que algo está bien si cumple con lo esperado. Si el que lo encarga está satisfecho. Si pasa la prueba del usuario.

Formalmente, diremos que **algo está bien si cumple unos requisitos**. Si pasa las pruebas de software que lo validen. En definitiva, las pruebas del software son el primer check point que demuestra que lo hecho está bien. Si no las tienes, la demostración es más costosa o inexistente.

Del manifiesto ágil al artesano.



La artesanía se base en la mejora continua, por el orgullo y el saber hacer del artesano. Y para un artesano del software el código correcto es mejorable en su diseño interno. **El refactoring**. Hazlo cada vez mejor.



### Disciplina y buenos hábitos.

Para iniciar la transformación que te llevará de la programación a la artesanía del software vas a necesitar algo más conocimientos. Vas a necesitar disciplina para aplicar lo aprendido. Y para ello has de incorporar buenos hábitos y desterrar los malos.

Es un viaje, no un salto. Tómate tu tiempo y entrena tu paciencia porque merece la pena dejar de ser un operario sin valor diferencial.

Ni artista  ni operario/a 

### Todo está en los libros:

Obviamente este cambio de actitud es mejor hacerlo acompañado. En persona y en conocimiento. Para lo segundo te dejo asuí una lista de libros que te ayudarán a enfocar tu carrera en el desarrollo de software.

## Programación

- Clean Code (*Robert C. Martin.*)
- Extreme Programming (*Kent Beck*)
- Refactoring (*Martin Fowler*)
- The pragmatic programmer (*Hunt & Thomas*)
- The Software Craftsman (*Sandro Mancuso*)

## Gestión

Y un par de ellos que te pueden servir para organizarte y mejorar en otros aspectos de la vida.

- Getting things done (*David Allen*)
- Atomic habits (*James Clear*)

Bienvenidas y bienvenidos al gremio de los artesanos y artesanas del software.

## 5.1 - Hacer las cosas bien

---

### Del manifiesto ágil al artesano.

*"Deja siempre el campamento más limpio de lo que estaba."*

--  **La regla del BoyScout**

Otra de las frases míticas que partiendo de un contexto alejado de la tecnología es una enseñanza fundamental para la artesanía del software.

El resumen para campistas sería que ir al bosque a limpiarlo es un trabajo, no un placer. Ir a ensuciarlo es incívico. Ir y recoger lo ensuciado es neutro. Pero el verdadero valor está en ir, disfrutar, recoger y si te encuentras algo que no es tuyo... recogerlo también. **Enfádate si quieres, pero recógelo.**

Aplicada al software querría decir algo así:

- No te plantees como trabajo el limpiar todo un programa.
- A menos, que ese sea tu trabajo. Revisor o limpiador de código.
- Cuando programes algo, déjalo limpio al salir. No agregues mierda.
- Si, de paso que visitas un método o una clase, ves algo sucio; límpialo.

Ya está, si **tú haces esto todos los días**, te garantizo que en unos meses el código estará mucho más limpio. Y lo que es mejor, casi sin esfuerzo.

Porque hagamos lo que hagamos debemos de hacerlo rentable y ágil. Rápido, repetible, medible, predecible. Ágil.

### Manifiesto ágil

**Individuos e interacciones sobre procesos y herramientas**

**Software funcionando sobre documentación extensiva**

**Colaboración con el cliente sobre negociación contractual**

**Respuesta ante el cambio sobre seguir un plan.**

Traigo aquí el manifiesto ágil porque es una gran influencia en otros movimientos como *extreme programming* y técnicas de desarrollo dirigida por los tests *TDD* que surgen de la aplicación de otro manifiesto: **el artesano**.

### Manifiesto artesano

**No sólo individuos e interacciones, también una comunidad de profesionales**

**No sólo software que funciona, también software bien diseñado**



**No sólo colaboración con el cliente, también asociaciones productivas**

**No sólo respondo al cambio, también agrego valor**

Y hacer las cosas bien, es **hacerlo todos los días un poco mejor**. Y mucho mejor si es con pruebas.



## 5.2 - Disciplina y buenos hábitos.

Ni artista  ni operario/a 

*"Voy a hacer un buen trabajo.*

*No tengo prisa.*

*Escribiré tests.*

*Iré rápido a base de hacerlo bien.*

*No escribiré mierda.*

*Voy a practicar, practicar y practicar*

*porque eso es lo que hace un profesional."*

--  **Sandro Mancuso**

Estas dos frases encierran toda la sabiduría que necesitas para **desarrollar tu carrera en el mundo de la programación**.

Esta profesión es **creativa y técnica a la vez**. Pero a la vez, no por separado. Esa intersección entre artista y operario es la figura del **artesano o artesana**. Alguien que ama lo que hace, pero buscando un resultado repetible y práctico.

Y uno de los signos de artesanía es la escritura de código limpio. No es un fin en si mismo, es el resultado de **alguien que decide** hacer las cosas bien, **que se preocupa** por sus compañeros y **que se compromete** en un proceso de **mejora y aprendizaje constante**.

Para adquirir esos valores se necesita disciplina, y la disciplina es cuestión de hábitos. Espero que este tutorial te haya inspirado para **adquirir buenos hábitos de programación**.

*"El **código limpio** no se escribe siguiendo un conjunto de reglas.*

*No te conviertes en un artesano de software al aprender una lista de heurísticas.*

*El profesionalismo y la artesanía provienen de valores que impulsa la **DISCIPLINA**."*

--  **Robert C. Martin**