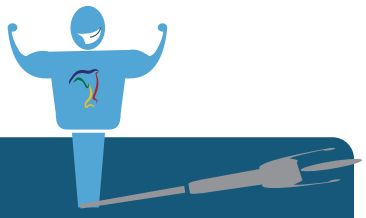
autentia

Avenida de Castilla,1 - Edificio Best Point - Oficina 21B
28830 San Fernando de Henares (Madrid)
tel./fax: +34 91 675 33 06
info@autentia.com - **www.autentia.com**

¿**Qué ofrece** Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
Ese apoyo que siempre quiso tener...

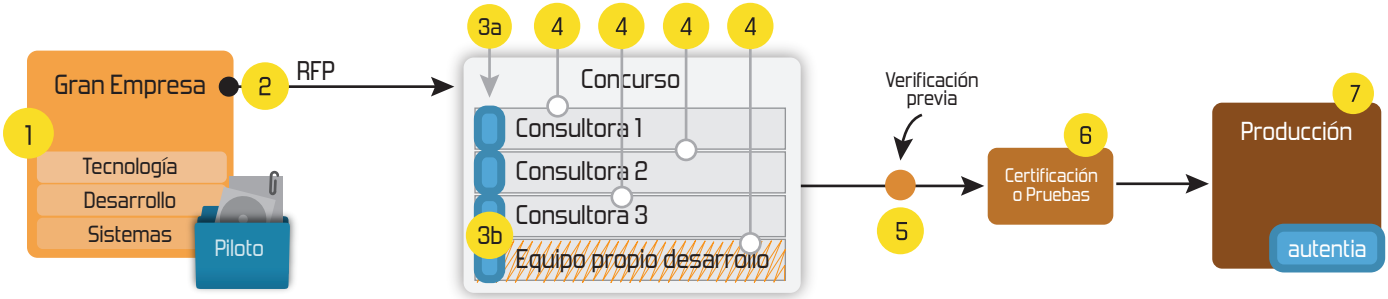
1. Desarrollo de componentes y proyectos a medida



2. Auditoría de código y recomendaciones de mejora

3. Arranque de proyectos basados en nuevas tecnologías

1. Definición de frameworks corporativos.
2. Transferencia de conocimiento de nuevas arquitecturas.
3. Soporte al arranque de proyectos.
4. Auditoría preventiva periódica de calidad.
5. Revisión previa a la certificación de proyectos.
6. Extensión de capacidad de equipos de calidad.
7. Identificación de problemas en producción.



4. Cursos de formación (impartidos por desarrolladores en activo)

Spring MVC, JSF-PrimeFaces /RichFaces, HTML5, CSS3, JavaScript-jQuery

Gestor portales (Liferay)
Gestor de contenidos (Alfresco)
Aplicaciones híbridas

Control de autenticación y acceso (Spring Security)
UDDI
Web Services
Rest Services
Social SSO
SSO (Cas)

JPA-Hibernate, MyBatis
Motor de búsqueda empresarial (Solr)
ETL (Talend)

Dirección de Proyectos Informáticos.
Metodologías ágiles
Patrones de diseño
TDD

Tareas programadas (Quartz)
Gestor documental (Alfresco)
Inversión de control (Spring)

BPM (jBPM o Bonita)
Generación de informes (JasperReport)
ESB (Open ESB)

Compartimos nuestro conocimiento en:
www.adictosaltrabajo.com

Para más información visítenos en:
www.autentia.com

DESARROLLADO POR:
Miguel Arlandy Rodríguez

Consultor tecnológico de desarrollo de proyectos informáticos.

Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/JEE

Fecha de publicación del tutorial: 2011-09-06

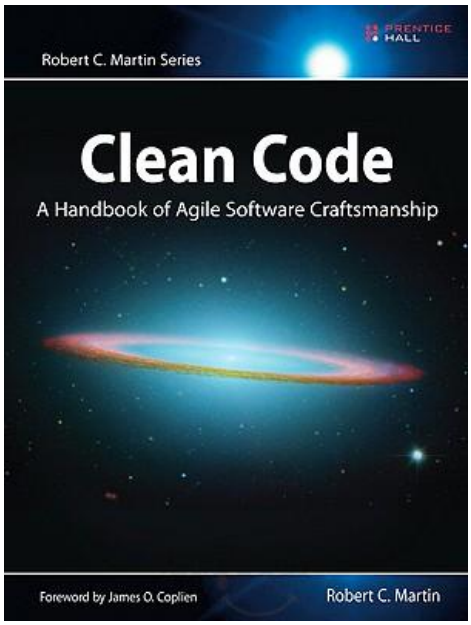
Clean Code: Impresiones.

0. Índice de contenidos.

1. Introducción

Clean Code es el título de un libro escrito por Robert C. Martin (Uncle Bob) donde nos habla de cómo escribir "código limpio", ese código bien estructurado, fácil de comprender, robusto y, a su vez fácil de mantener. Ese código donde no hay código duplicado. Código que se apoya en patrones de diseño.

En este artículo pretendo compartir mis impresiones sobre esta obra que he leido recientemente y que no me ha dejado, ni mucho menos, indiferente.



2. ¿A quién va dirigido?.

Clean Code está dirigido para todos aquellos desarrolladores de software que deseen mejorar el diseño y la calidad de su código. No importa el nivel que tengas. Estoy convencido de que, incluso los desarrolladores más experimentados, aprenderán algo nuevo en cada capítulo. Para los desarrolladores más noveles, será una ayuda más que interesante para que empiecen a forjar su estilo.

Considero que es un libro especialmente interesante para desarrolladores que trabajan habitualmente en equipo ya que, Uncle Bob, constantemente intenta sensibilizarnos acerca de la importancia de escribir código para que lo entiendan los demás, no para que lo comprendamos únicamente nosotros.

3. Consejos para leer el libro.

Clean Code no es un "libro de playa" o de "cuarto de baño". Es un libro técnico y por tanto, tiene un relativo nivel de complejidad. Para sacarle el máximo partido hay que leerlo despacio, sin prisa por terminarlo.

- Párate de vez en cuando a reflexionar sobre los conceptos que nos trata de transmitir Uncle Bob. No quieras leerlo de carrerilla.
- Vuelve a releer determinados puntos si ves que no te han quedado muy claros. Repito, no tengas prisa por terminarlo. Lo importante es aprender de él.
- No intentes aprenderte las cosas de memoria, intenta comprender el por qué de ellas y contrástala con tus experiencias.

Además debes saber que Clean Code está escrito en inglés, por lo que si no tienes relativa soltura leyendo en este idioma se puede convertir en un problema.

4. Resumen de contenidos.

Aunque el objetivo de este artículo no pretende ser el de hacer un resumen del libro, aquí os dejo un pequeño resumen de contenidos.

- Capítulo 1: Clean Code. Introducción al "código limpio".
- Capítulo 2: Meaningful Names. Resalta la importancia de elegir nombres apropiados para nuestras clases, métodos, variables...
- Capítulo 3: Functions. Cómo definir funciones correctamente. Funciones cortas, atómicas y con el mismo nivel de abstracción.
- Capítulo 4: Comments. Distintos tipos de comentarios y cómo deberíamos utilizarlos.
- Capítulo 5: Formatting. Consejos sobre el formato que debería tener nuestro código.
- Capítulo 6: Objects and Data Structures. Qué uso hay que dar a los objetos y a las estructuras de datos.
- Capítulo 7: Error Handling. Relación entre excepciones y la lógica de negocio.
- Capítulo 8: Boundaries. Cómo elegir y utilizar el código de terceros.
- Capítulo 9: Unit Tests. Importancia de los tests y qué reglas deben cumplir.
- Capítulo 10: Classes. Cómo definir clases correctamente. Estructuración, tamaño, encapsulación...
- Capítulo 11: Systems. Cómo construir sistemas basados en POJO´s y sostenidos por tests.
- Capítulo 12: Emergence. Definición de reglas que nos ayudarán en nuestro diseño.
- Capítulo 13: Concurrency. Observaciones al diseño de sistemas concurrentes.
- Capítulo 14: Successive Refinement. Ejemplo de refactorización.
- Capítulo 15: JUnit Internals. Revisión al código del framework Junit.
- Capítulo 16: Refactoring SerialDate. Otro ejemplo de refactorización. En este caso de la clase SerialDate de la librería jcommon.
- Capítulo 17: Smells and Heuristics. Resumen de conceptos tratados durante el libro.
- Apéndice A: Concurrency II. Ampliación del capítulo 13.
- Apéndice B: org.jfree.date.SerialDate. Código fuente del paquete org.jfree.date.

Me parecieron especialmente interesantes los capítulos 2, 3, 5, 8 y 9.

5. La importancia de los tests.

Me llamó especialmente la atención la importancia que Robert C. Martin, da a los tests. No me estoy refiriendo únicamente al hecho de que nuestras apliaciones deban hacer uso de ellos, sino a cómo crearlos correctamente. Cuanquiera que sepa un poco de TDD sabrá a lo que me refiero. No es suficiente con hacer tests, hay que hacerlos correctamente (F.I.R.S.T) y ejecutarlos constantemente.

Además, me gustó mucho la parte de los "Learning Tests". Aquellos tests que surgen de la necesidad de comprobar si un framework o librería de terceros, hace lo que realmente nosotros esperamos que haga. Este tipo de tests son un tanto atípicos ya que no ayudan a desarrollar y mantener nuestro sistema. Son tests que nos ayudan a comprender el funcionamiento de un código de terceros y comprueban que haga lo que esperamos. Es interesante mantener este tipo de tests en nuestra aplicación ya que ayudarán al equipo de trabajo a comprender determinados aspectos del framework o librería sin necesidad de haberse tenido que pegar con ello.

En este punto simplemente quería avisar al futuro lector de que se va a encontrar constantemente referencias a la importancia de los tests.

6. Antes y después de leer el libro.

Estoy convencido de que hay un antes y un después en la forma de escribir código de todo aquel que haya leido este libro.

Notarás que no es suficiente con que el código haga lo que tiene que hacer. Además tiene que ser "limpio". Ver código duplicado será más "doloroso" de lo que era antes. Te volverás más exigente con tu estilo de desarrollo. Pero, sobre todo, tendrás la satisfacción personal de saber que estás haciendo las cosas bien.

7. Conclusiones.

Todos (o la mayoría) tenemos una lucha personal por ser mejores profesionales e intentar hacer las cosas cada vez mejor y, el mundo del desarrollo de software no es una excepción. Clean Code es un libro de gran ayuda para todo aquel que quiera mejorar su código.

En este artículo, totalmente subjetivo, he intentado plasmar algunas de mis impresiones sobre Clean Code e intentar motivar a los desarrolladores a que lo lean. Os aseguro que merece la pena ;).

Espero que os haya sido de ayuda. Un saludo.

Miguel Arlandy

marlandy@autentia.com

Anímate y coméntanos lo que pienses sobre este **TUTORIAL:**

Puedes opinar o comentar cualquier sugerencia que quieras comunicarnos sobre este tutorial; con tu ayuda, podemos ofrecerte un mejor servicio.

Enviar comentario

(Sólo para usuarios registrados)

» **Registrate** y accede a esta y otras ventajas «

COMENTARIOS