

¿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

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

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

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



Samuel Martín Gómez-Calcerrada

Consultor tecnológico de desarrollo de proyectos informáticos.

Ingeniero en Informática, especialidad en Ingeniería del Software.

 Puedes encontrarme en [Autentia](#): Ofrecemos servicios de soporte a desarrollo, factoría y formación

Somos expertos en Java/J2EE

[Ver todos los tutoriales del autor](#)

Catálogo de servicios Autentia



¿Quieres aprender inglés?

144 videoclases por 0€! Regístrate y aprende gratis


 Fecha de publicación del tutorial: **2015-01-26**

 Tutorial visitado 4 veces [Descargar en PDF](#)

Menos es más. Filosofía detrás de los principios YAGNI, KISS, DRY, DIE, hijos de la navaja de Ockham.

Índice de contenidos.

- Introducción
- Orígenes, la navaja de Ockham
- Keep It Simple, Stupid!
- You Aren't Gonna Need It
- Don't Repeat Yourself (DRY) / Duplication Is Evil (DIE)
- Conclusiones

Introducción

Este artículo pretende profundizar en el significado de todas las frases bonitas que utilizamos hoy en día y que quedan tan bien en las charlas de ponentes en ferias tecnológicas. Ahora que programar está a la moda, mucha gente influyente ha acuñado términos sencillos y directos con un acrónimo del que es fácil acordarse, como DRY "seco", KISS "beso" o DIE "morir".

Quizá KISS (Keep It Simple, Stupid!) es el mejor ejemplo de lo que estoy diciendo.

Esto busca viralizar la frase, lo cual no es malo, ya que el mensaje de fondo es algo que todos deberíamos aplicar en nuestro día a día como programadores, e incluso en nuestra vida diaria.

Orígenes, la navaja de Ockham

Todo comienza con William de Ockham, un fraile franciscano que vivió entre el 1287 y el 1347. Propuso que cuando había que elegir entre varias hipótesis, lo mejor era considerar como cierta la que menos asunciones hiciera siempre que el resto de parámetros se mantuvieran constantes.

"All things being equal, the simplest solution tends to be the best one"

"Mientras todo se mantenga igual, la solución más simple tiende a ser la mejor"

Como curiosidad, el término Navaja de Ockham apareció siglos mas tarde, acuñado por Sir William Hamilton en 1852.

Mucha gente puede intentar radicalizar esto, siendo demasiado simplistas, podemos decir que todo es obra de la magia.

Para evitar estas evidentes falacias, me gustaría acotar el principio de Ockham con una frase acuñada por Albert Einstein

"Everything Should Be Made as Simple as Possible, But Not Simpler"

"Todo debería ser tan simple como sea posible, pero nunca mas simple que eso"



Síguenos a través de:



Últimas Noticias

» [2015: ¡Volvemos a la oficina!](#)

» [Curso JBoss de Red Hat](#)

» [Si eres el responsable o líder técnico, considérate desafortunado. No puedes culpar a nadie por ser gris](#)

» [Portales, gestores de contenidos documentales y desarrollos a medida](#)

» [Comentando el libro Start-up Nation, La historia del milagro económico de Israel, de Dan Senor & Salu Singer](#)

[Histórico de noticias](#)

Últimos Tutoriales

» [Monitorización y alertas en aplicaciones java con AppDynamics](#)

» [Utilización de OAuth2 con Github](#)

» [Introducción a Spring Social](#)

» [AngularJS y los tests](#)

Últimos Tutoriales del Autor

- » [Utilización de OAuth2 con Github](#)
- » [\[S.O.L.I.D.\] Dependency inversion principle / Principio de inversión de dependencias](#)
- » [\[S.O.L.I.D.\] Interface Segregation Principle / Principio de segregación de interfaz](#)
- » [\[S.O.L.I.D.\] Liskov substitution](#)
- » [\[S.O.L.I.D.\] Open-Closed Principle / Principio Abierto-Cerrado](#)

Categorías del Tutorial

-  [Cultura de Empresa / Efectividad personal](#)

Keep It Simple, Stupid!

Aunque se ha traducido normalmente por Keep it simple, stupid! (¡manténlo simple, idiota!), esto es simplemente por dar mas juego al acrónimo, ya que su autor, Kelly Johnson lo ha traducido sin coma, lo que no insulta al ingeniero encargado si no que refuerza la idea de sencillez.

Esta frase goza de mucha popularidad entre los conferenciantes y evangelistas de las buenas prácticas porque, como se puede apreciar, es una frase con mucha fuerza, que hace despertar a los oyentes y les insta a twittear y difundir el mensaje (y al mensajero).

Todo lo que sea mejorar la calidad de nuestros programas lleva a una mejor y más barata mantenibilidad, y por lo tanto estoy a favor :)

You Aren't Gonna Need It

Literalmente traducido por "No vas a necesitarlo". Este principio esta detrás de otro mucho mas largo y feo como es DTSTTCPW utilizado en extreme programming que viene a decir "haz la cosa mas simple que pueda funcionar" (Do The Simplest Thing That Could Possibly Work).

Está pensado para utilizarse en la metodología de trabajo de extreme programming junto con las pruebas y la refactorización. Ayuda a no intentar hacer una mega-solución que solucione los problemas del mañana porque estos pueden cambiar, y, en lugar de esto, centrarse en únicamente lo necesario, que por otra parte es por lo que nos pagan.

Don't Repeat Yourself (DRY) / Duplication Is Evil (DIE)

Agrupo estos dos acrónimos porque su significado es el mismo, "no te repitas" y la "duplicación es el mal".

Si ya tienes una funcionalidad existente, es mejor reutilizarla que duplicarla, porque en este caso tendrías que mantener ambos algoritmos. Esto suena demasiado sencillo (y lo es), pero el mayor problema que he observado, en el que se escuda la gente, para no duplicar código es que dos funcionalidades no suelen ser exactamente iguales, siempre hay sutiles diferencias que hacen que sea mas fácil copiar lo que tenemos y pegarlo con pequeñas modificaciones.

El "truco" para evitar esto es la refactorización. Extraer las partes comunes es esencial para no duplicar funcionalidad, es una de las partes fundamentales del TDD y del extreme programming.

Si has seguido el principio YAGNI probablemente tengas código duplicado con otras funcionalidades parecidas pero no iguales.

Para aplicar además los principios DRY y DIE, ahora es el turno de refactorizar extrayendo partes comunes para eliminar duplicidad en tu código.

Una vez hecho esto consigues contentar a todos los principios y puedes empezar con la siguiente funcionalidad.

Conclusiones

Podríamos decir que hemos "refactorizado" estos principios para extraer la parte común de ellos, sería curiosa la reacción de la gente que los propuso ver que se a aplicado su principio a si mismo ;)

Espero que con este artículo ahora poseáis una vision común de todo esto y que tengáis un criterio propio para defender su uso frente a otros trabajadores que prefieren mirar a corto plazo y copiar/pegar trocitos para sobrellevar el día a día o crear una super arquitectura que pueda resolver cualquier posible cambio que se produzca de aquí a años vista (porque probablemente eso no pase).

Como siempre os digo, utilizad el sentido común, nada es blanco o negro. Espero que os haya gustado y haberos aclarado dudas que pudierais tener.

Samuel Martín.

A continuación puedes evaluarlo:

[Regístrate para evaluarlo](#)



Por favor, vota +1 o compártelo si te pareció interesante

 Share |        0   0

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

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



Esta obra está licenciada bajo licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5

PUSH THIS

Page Pushers

Community

Help?

no clicks

0 people brought clicks to this page



powered by [karmacracy](#)

Copyright 2003-2015 © All Rights Reserved | [Texto legal y condiciones de uso](#) | [Banners](#) | [Powered by Autentia](#) | [Contacto](#)

