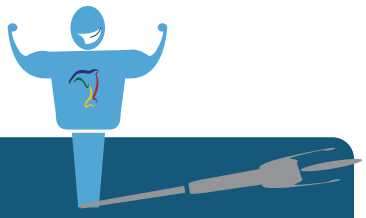
autentia

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...

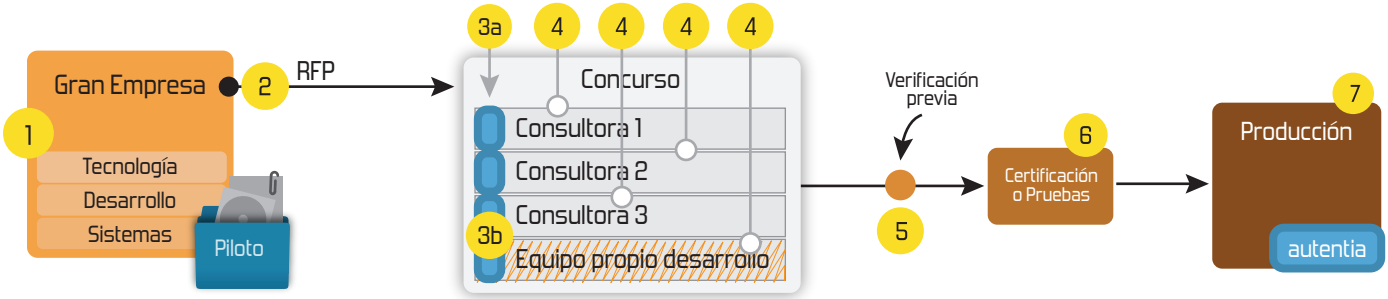
1. Desarrollo de componentes y proyectos a medida



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:
Alejandro Pérez García

Alejandro es socio fundador de Autentia y nuestro experto en J2EE, Linux y optimización de aplicaciones empresariales.

Ingeniero en Informática y Certified ScrumMaster

Si te gusta lo que ves, puedes contratarle para darte ayuda con soporte experto, impartir **cursos presenciales** en tu empresa o para que **realicemos tus proyectos como factoría** (Madrid). Puedes encontrarme en Autentia: Ofrecemos servicios de soporte a desarrollo, factoría y formación

Fecha de publicación del tutorial: 2009-02-26

Reunión Madrid Ágil 02-11-2010: DDD (Domain Driven Design)

Creación: 03-11-2010

Índice de contenidos

Histórico de NOTICIAS

1. Introducción

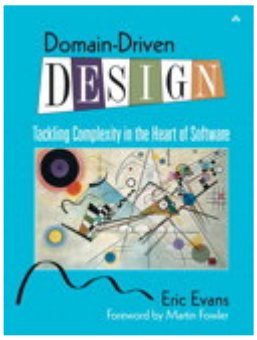
El 03-10-2010 tuvo lugar una de las reuniones de Madrid Ágil, en la que se habló de DDD (Domain Driven Design). En esta reunión también se tenía que haber hablado de BDD y de TDD, pero estos temas quedaron marginados y relegados a otra próxima reunión ya que Alfredo Casado (@AlfredoCasado) se curró una presentación impresionante de DDD con ejemplo de código incluido!!! Y estuvimos liados con esto durante toda la reunión, que por cierto, fue muy muy muy interesante.

En este artículo voy a contar mis impresiones después de la reunión.

2. DDD (Domain Driven Design)

Bueno y qué es eso del "Domain Driven Design". Una traducción más o menos literal podría ser "Diseño Dirigido por el Dominio", pero ¿y esto qué es?

El termino viene acuñado por Eric Evans en su libro de igual nombre: http://domaindrivendesign.org/books/evans_2003



Lo que nos ha quedado claro que es DDD no es una método, ni una técnica, ni una tecnología. Podríamos decir que es simplemente un conjunto de buenas prácticas especialmente pensadas para lidiar con dominios complejos o múltiples dominios, de forma que estas prácticas nos guíen en la toma de decisiones para elaborar nuestro diseño.

Algunos de los pilares de DDD son:

- Potenciar la colaboración con los interesados y expertos del dominio. Todo el mundo tiene que ayudar a definir el modelo (el modelo es la representación del dominio, el modelo son los elementos de tu dominio y sus relaciones).
- Tiene que existir un leguaje ubícuo, es decir, tiene que haber un mismo lenguaje que esté en todas partes, tanto en los expertos del dominio, como en los técnicos, como en el modelo, como en el código !!!
- Cualquier técnico tiene que picar. No hay gente que sólo pinte y no hay gente que sólo codifique, todo el mundo participa en todo.
- Cuando escribes código estás participando en el modelo. El código es la implementación del modelo, pero es todo lo mismo. No tiene sentido definir un modelo y luego implementar otra cosa, en tal caso el modelo ha perdido todo su valor y hemos perdido el tiempo.
- Como el código es el modelo, si cambiamos código habrá que avisar a los interesados y expertos del dominio.

Algunos conceptos definidos por DDD:

- **Entidad**: Es un objeto que tiene un identificador único en el contexto que estamos tratando. En el ejemplo que preparó Alfredo teníamos al "Alumno".

Últimos Tutoriales

DataTable con paginación en base de datos con Primefaces

Resolviendo el cubo de Rubik (II) - ayudas, ejemplo Wink, extensiones y encuesta

Configuración de jCaptcha en Struts

MySQL - Sensibilidad a mayúsculas/minúsculas de los nombres de las tablas

Cómo evitar tener más de dos cabezas en Mercurial

Últimos Tutoriales del Autor

Cómo evitar tener más de dos cabezas en Mercurial

Reunión Madrid Ágil 14 -10-2010: Equipos autogestionados, y motivación del individuo y del equipo

Spring + REST + JSON = SOAUI

Redescubriendo el Agilismo

Reunión Madrid Ágil 21 -09-2010: Estrategias de Branches, y

- **Value Object**: Es un objeto que tiene atributos, pero no tiene identificador en el contexto. Pueden tener comportamiento y suelen ser inmutables. Sirven para describir cosas. En el ejemplo de Alfredo eran los "DatosDeContacto" del "Alumno".
- **Servicio**: Son los verbos del lenguaje ubícuo. Un servicio aparece cuando tenemos una funcionalidad que no pertenece a ningún objeto del dominio, en tal caso creamos un nuevo objeto que no es del dominio y que tendrá la responsabilidad de realizar esta funcionalidad. Esto es el patrón Fabricación Pura de GRASP. No tienen estado. Tienen que tener poco código, si hay mucho es que no se está haciendo bien la Orientación a Objetos. Normalmente se limitan a "sincronizar" operaciones entre varios objetos del dominio.
- **Agregado**: Es una composición de objetos. Tiene una raíz que es una entidad. Lo normal es no poder acceder al contenido del Agregado, y todas las operaciones que se quieran hacer sobre el contenido siempre se harán a través de la entidad raíz. En el ejemplo de Alfredo el "Alumno" es la entidad raíz, y esta contienen a los "DatosDeContacto".
- **Factoría**: Crea objetos complejos (en muchos casos agregados). Crea objetos válidos (no devolverá nulos). Se parece mucho a los distintos patrones de creación de GOF.
- **Repositorio**: Básicamente es el patrón DAO. Persiste entidades y agregados, no Value Objects (Los Value Object se persisten al persistir el Agregado correspondiente). La interfaz está en el dominio pero la implementación esta en la capa de infraestructura. Es muy importante el ver como en DDD hay que ser muy consciente de que los objetos es necesario guardarlos en algún sitio, y por ello la interfaz del repositorio está dentro del dominio.
- **Módulos**: Paquetes, namespaces, ... Lo importante es que tienen que formar parte del lenguaje ubícuo (agrupación lógica). Es decir no tiene sentido hacer un módulo con todas las entidades, otro módulo con todos los repositorios, ...
- **Modelos anémicos**. DDD no habla expresamente de modelos anémicos, pero sí lo hace de modelos "ricos" que sería el opuesto. Lo que está claro es que DDD intenta centrarse en el modelo de dominio para precisamente crear modelos ricos y evitar los modelos anémicos.

3. Conclusiones

La reunión de ayer fue realmente muy interesante. Está claro que DDD es algo que hay que tener muy presente si queremos hacer sistemas que se puedan mantener y evolucionar en el futuro. En este artículo no hemos hecho más que rascar un poquito los principios de DDD, y por eso recomiendo leer el libro y en general toda la información que podás encontrar sobre el tema.

4. Sobre el autor

Alejandro Pérez García, Ingeniero en Informática (especialidad de Ingeniería del Software) y Certified ScrumMaster

Socio fundador de Autentia (Formación, Consultoría, Desarrollo de sistemas transaccionales)

mailto:alejandropg@autentia.com

Autentia Real Business Solutions S.L. - "Soporte a Desarrollo"

http://www.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