

¿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)

	Hosting Patrocinado por enREDados.com 
---	--

[Home](#) | [Quienes Somos](#) | [Empleo](#) | [Foros](#) | [Tutoriales](#) | [Servicios Gratuitos](#) | [Contacte](#)

	Tutorial desarrollado por: Roberto Canales Mora 2003-2005 <u>Creador de AdictosAlTrabajo.com</u> y <u>Director General de Autentia S.L.</u> Recuerda que me puedes contratar para echarle una mano: Desarrollo y arquitectura Java/J2EE Asesoramiento tecnológico Web Formación / consultoría integrados en tu proyecto No te cortes y contacta: 655 99 11 72 rcanales@autentia.com.	
---	---	---

Descargar este documento en formato PDF [aspectj.pdf](#)

[Curso Web J2EE](#)

Curso Avanzado en Desarrollo Web con J2EE

[Royalty-Free JCA Adapters](#)

The critical last mile of EAI 50+ including SAP, Orcl, Psft, JDE

[NetBeans IDE 4.1](#)

Create EJBs & Web Services, ant-based project system and more.

[RPG apps to Java 2EE](#)

eCube Systems transforms AS/400 RPG legacy apps to Java and J2EE.

Anuncios Goooooogle

Anunciarse en este sitio

Programación con aspectos: AspectJ

Los que trabajamos con Java habitualmente nos encontramos un grave problema:

- La misma cosa se puede hacer de 20 modos distintos y nadie nos dice cual es el más adecuado. Además como evoluciona todo tan deprisa, lo que eran verdad (o podríamos pensar) hace unos meses, es posible que ahora no lo sea.

Hoy vamos a hablar de otro planteamiento de contrucción de aplicaciones, la denominada programación por aspectos.

Cuando construimos una aplicación, normalmente nuestros procesos de negocio realizan una serie de tareas:

1. Verificación de credenciales.
2. Verificación de precondiciones.
3. Comienzo de una transacción.
4. **Ejecución de nuestra función de negocio.**
5. Finalización de una transacción.
6. Verificación de post-condiciones.
7. Y además, sin un orden concreto, escritura de trazas, medida de la velocidad, etc.

De todos los puntos anteriores, solo el punto 4 es realmente nuestra función de negocio.

El problema ahora es que si queremos quitar, poner o modificar algunas de las otras funcionalidades colatorales, tenemos un grave problema... tocar mucho código.

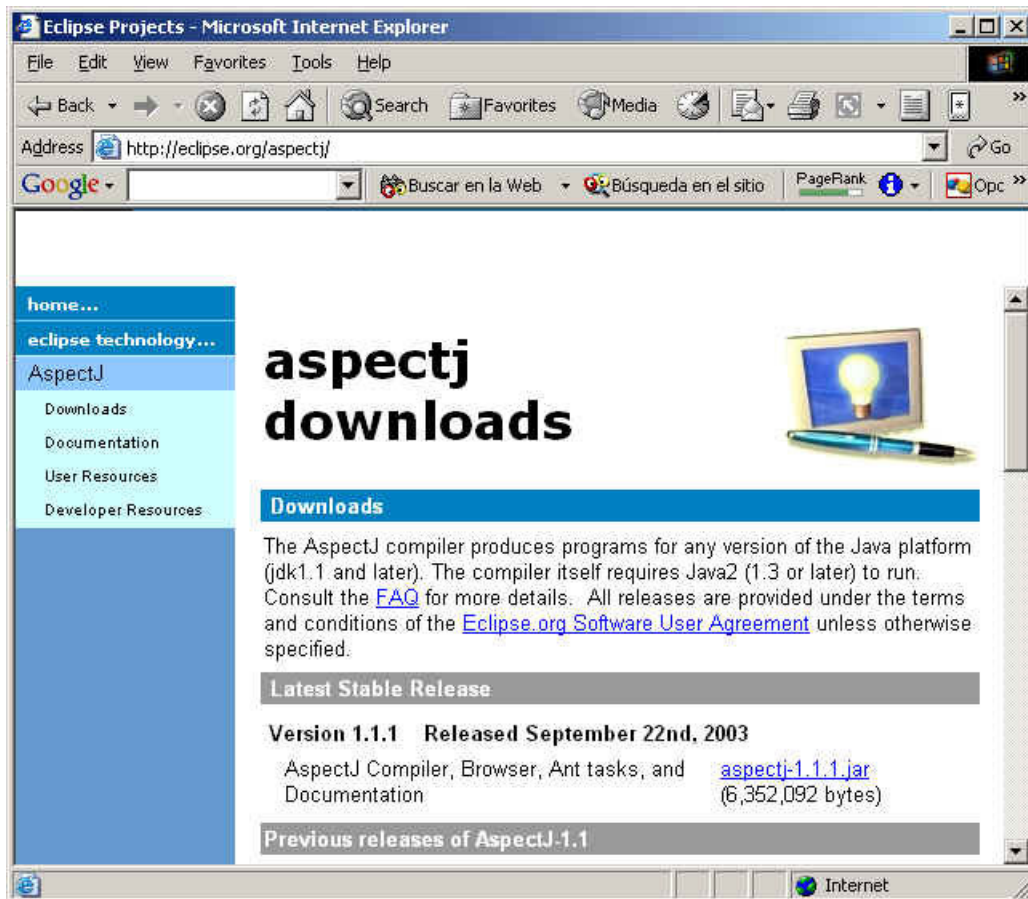
Quitar la funcionalidad o hacer que no se ejecute en un momento determinado (en base a un parámetro), puede ser relativamente fácil pero ¿cómo hacemos para introducir o cambiar esta funcionalidad sin tocar nuestros programas?... y para que nadie se lie con patrones de diseño (factorías y similares) pongamos que queremos calcular en tiempo de ejecución de todas las funciones en nuestro programa JEJEJE..... esto ya no es tan fácil ¿verdad?.

¿Cual es la solución? Una de ellas es separar los distintos **aspectos** de las aplicaciones.

Antes de entrar en teoría, vamos a instalar el software y verlo en un ejemplo.

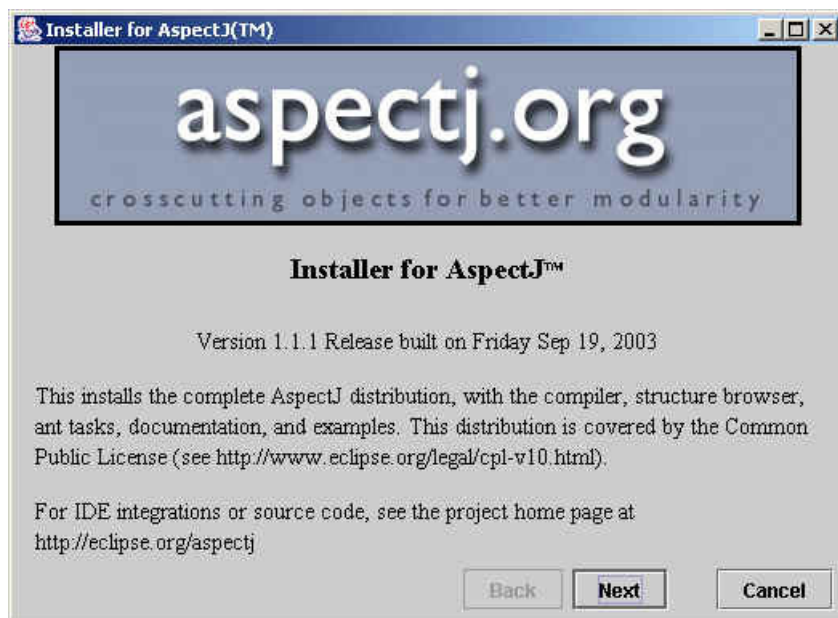
Instalación de AspectJ

AspectJ es una extensión de Java que podemos encontrar en Eclipse.org.

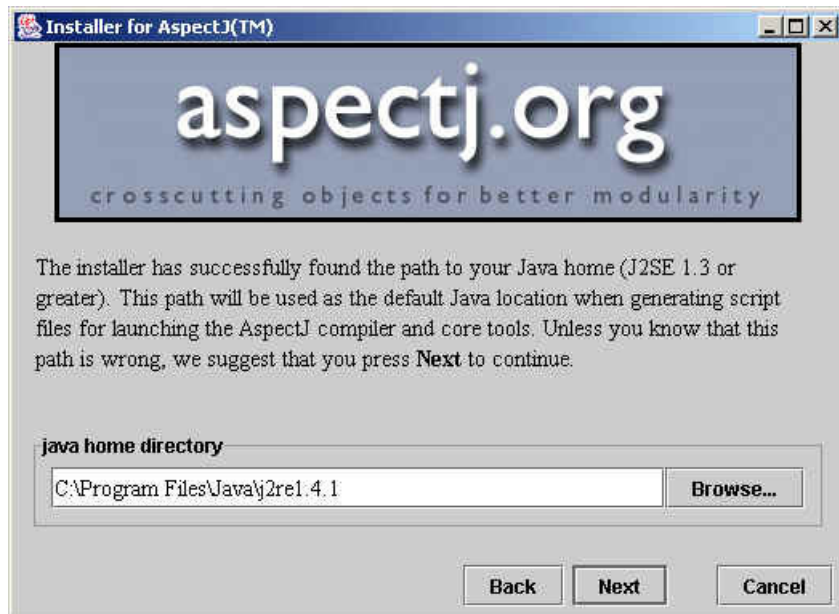


Nos descargamos el Jar y lo ejecutamos.

Nos aparece el instalador...



Seleccionamos la máquina virtual a ejecutar



Elegimos es directorio de trabajo



Y debemos tocar las variables de entorno de nuestro sistema ...

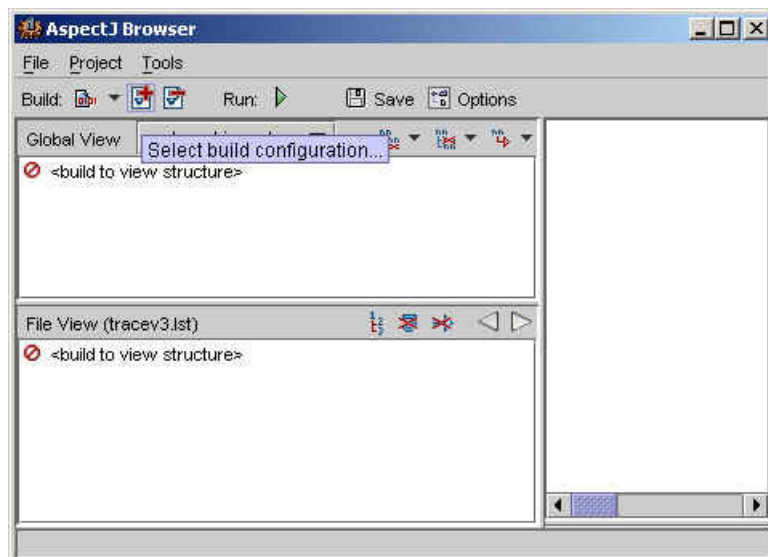
Incluimos en el **classpath** el **jar** : c:\java\aspectj1.1\lib\aspectjrt.jar
y el en **path** el directorio **bin**: c:\java\aspectj1.1\bin



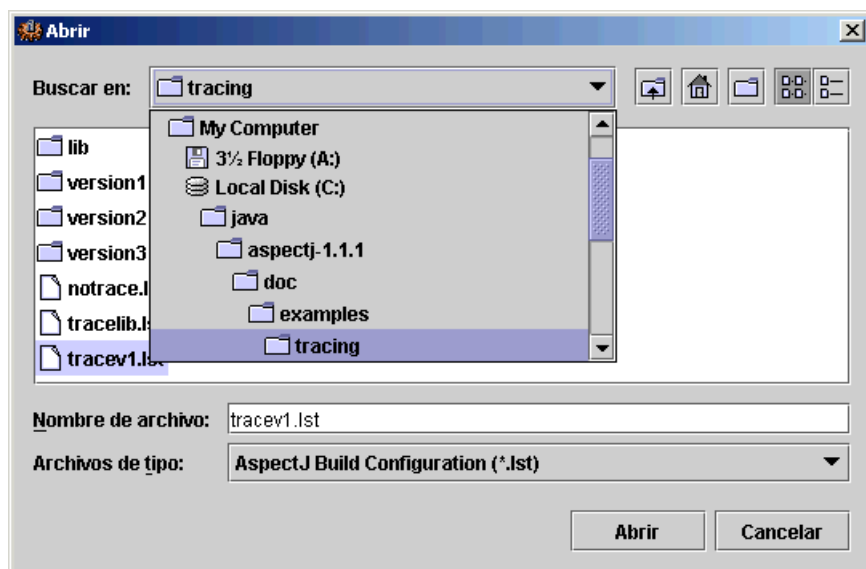
Si vamos al directorio de instalación, podemos encontrar el entorno de desarrollo de AspectJ



Es aspecto inicial es el siguiente. Pulsamos el símbolo + para añadir un nuevo fichero de configuración.

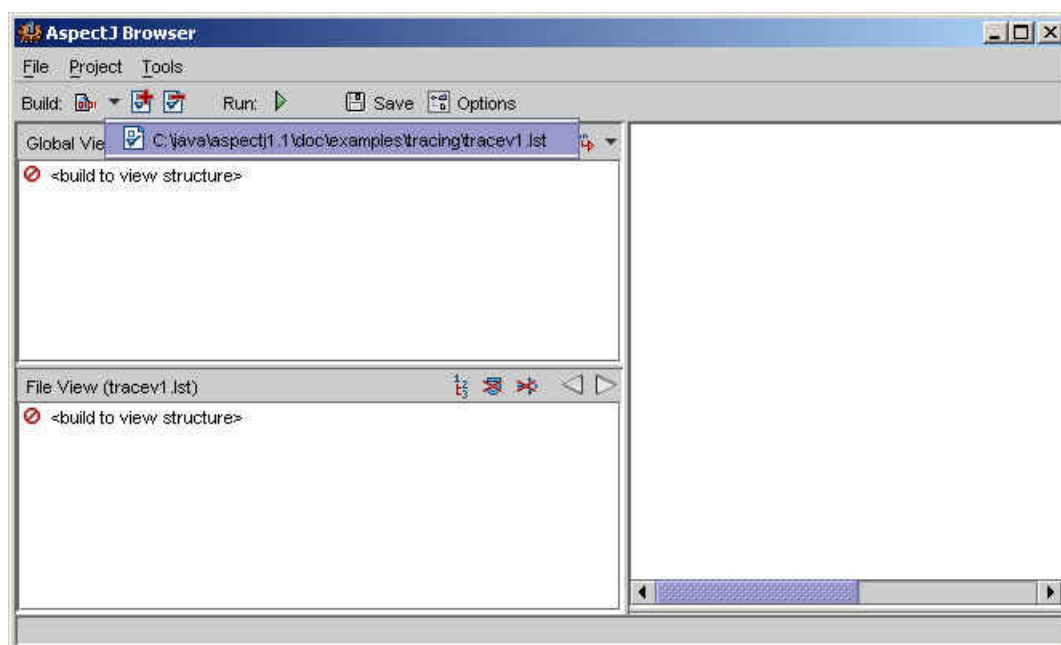


Ahora vamos a cargar uno de los ejemplos para comprobar que el sistema funciona.

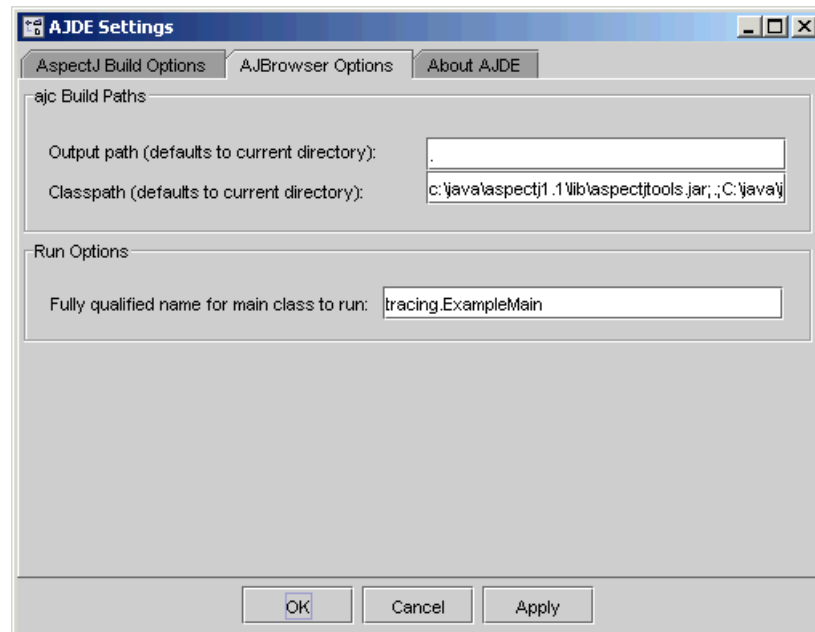


Pulsamos el icono Build (construir)

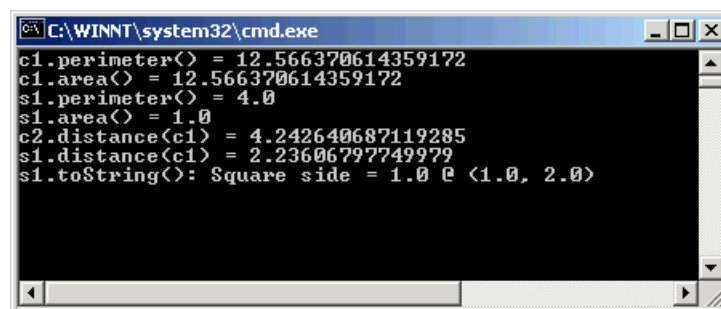
En este caso se compila el sistema....



Ahora, seleccionamos la clase que tiene un método main (tracing.ExampleMain)



Y podemos ver que aparece en la ventana unos mensajes extraños.....



Esto significa que el sistema está funcionando... ahora vamos a crear nuestro propio ejemplo.

Ejemplo de AspectJ

Vamos a crear un programa sencillo, que tiene una función que nos interesa saber cuanto tarda en ejecutarse

(Nota: Que nadie piense que esta tecnología vale para optimizar o medir velocidad.... podemos utilizarlo para realizar un control de parámetros, flujo de navegación, obtención de pools, paginación, persistencia.... y lo que se nos ocurra...)

```
public class adictosaltrabajo
{
    public static void main(String[] args)
    {
        adictosaltrabajo programa = new adictosaltrabajo();
        programa.procesa();
        System.out.println("Hemos terminado");
    }

    public void procesa()
    {
        long total = 0;

        // ejecutamos un bucle simple
        for(int i=0;i<10000000;i++)
        {
            total = total + 5 / 2 * 7 -4;
        }
    }
}
```

Si compilamos la clase y la ejecutamos obtenemos el siguiente resultado:

```

C:\WINNT\system32\cmd.exe

C:\java\aspectj-1.1.1\doc\examples\adictosaltrabajo>javac a*.java

C:\java\aspectj-1.1.1\doc\examples\adictosaltrabajo>java adictosaltrabajo
Hemos terminado

C:\java\aspectj-1.1.1\doc\examples\adictosaltrabajo>

```

Imaginad que ahora queremos saber el tiempo que tarda en ejecutarse la función **procesa....**

Gracias a AspectJ podemos crear un nuevo aspecto que interactúe con el código existente en base a unas reglas.

Lo que vamos a hacer es "pinchar" la llamada a la función y establecer un contador... todo esto sin tocar la lógica existente...

Editamos clase que define el aspecto:

```

import java.util.*;

aspect miaspecto
{
    private long tiempo = 0;

    pointcut mipuntotrazo(): execution(* adictosaltrabajo.procesa(..));

    before (): mipuntotrazo()
    {
        System.out.println("Entramos en funcion: " + thisJoinPoint);
        tiempo = System.currentTimeMillis();
    }

    after(): mipuntotrazo()
    {
        System.out.println("Salimos de funcion: " + thisJoinPoint +
            " y hemos tardado" + (System.currentTimeMillis() - tiempo ));
    }
}

```

Lo que estamos definiendo es un comportamiento:

Cuando crucemos nuestra clase con este aspecto, en el punto de corte

```
pointcut mipuntotrazo(): execution(* adictosaltrabajo.procesa(..));
```

definimos que cuando se ejecute la función **procesa**, ejecutaremos una función llamada **mipuntotrazo**.

Antes de ejecutar esta nuevo función

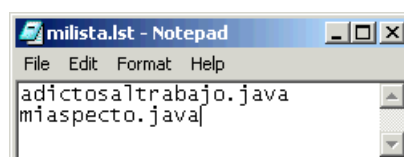
```
before (): mipuntotrazo()
```

definimos un contador

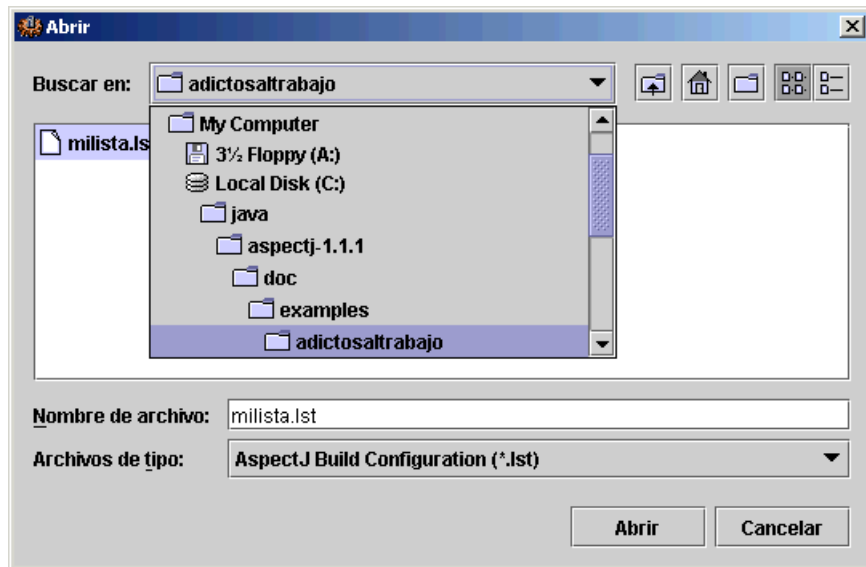
Y despues, calculamos el tiempo que ha tardado en ejecutarse

```
after(): mipuntotrazo()
```

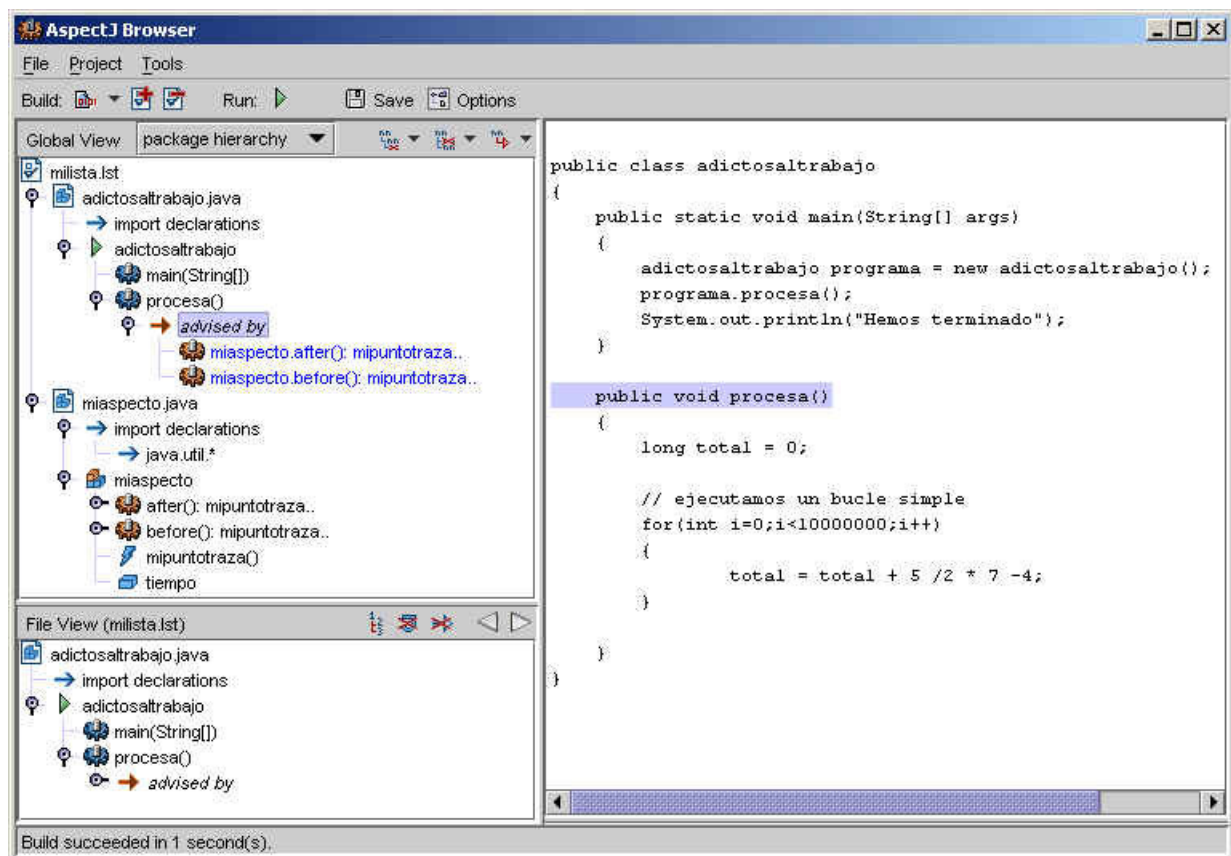
Ahora, copiamos los dos ficheros a un directorio y creamos un tercer fichero llamado milista.lst



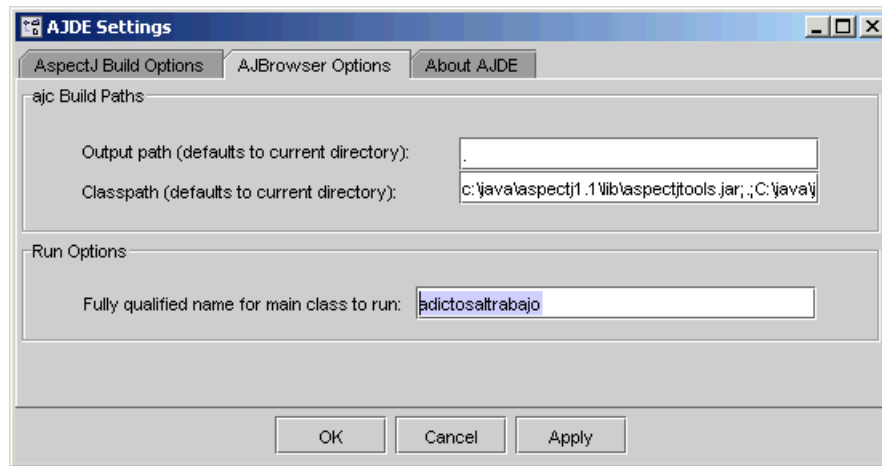
Arrancamos el browser de AspectJ y le decimos que utilice este fichero .lst



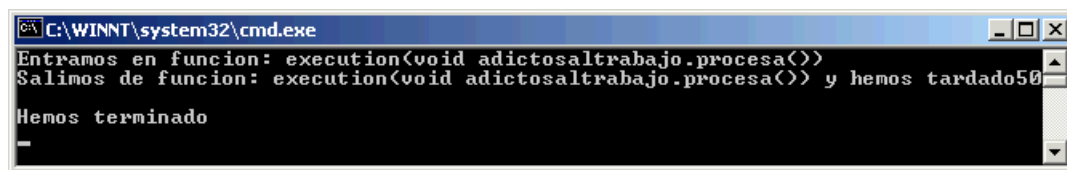
Y pulsamos el botón build y vemos que el sistema ha identificado este punto de cruce



Configuramos en las opciones que queremos ejecutar nuestra clase java



Y ejecutamos.... la sorpresa es que se ejecuta nuestra clase ... pero el sistema nos dice lo que ha tardado en ejecutarse



Podemos decompilar el código de nuestra clase **adictosaltrabajo.class** y ver como le ha afectado el uso de AspectJ

```
public class adictosaltrabajo
{
    public adictosaltrabajo()
    {
    }

    public static void main(String args[])
    {
        adictosaltrabajo programa = new adictosaltrabajo();
        programa.procesa();
        System.out.println("Hemos terminado");
    }

    public void procesa()
    {
        Object aobj[];
        org.aspectj.lang.JoinPoint joinpoint = Factory.makeJP(ajc$tjp_0, this, this, aobj = new Object[0]);
        try
        {
            miaspecto.aspectOf().ajc$before$miaspecto$96(joinpoint);
            long total = 0L;
            for(int i = 0; i < 0x989680; i++)
                total = (total + 14L) - 4L;

        }
        catch(Throwable throwable)
        {
            miaspecto.aspectOf().ajc$after$miaspecto$125(joinpoint);
            throw throwable;
        }
        miaspecto.aspectOf().ajc$after$miaspecto$125(joinpoint);
    }

    public static final org.aspectj.lang.JoinPoint.StaticPart ajc$tjp_0;

    static
    {
        Factory factory = new Factory("adictosaltrabajo.java", Class.forName("adictosaltrabajo"));
        ajc$tjp_0 = factory.makeSJP("method-execution", factory.makeMethodSig("1-procesa-adictosaltrabajo----void-", 13));
    }
}
```

Es decir, se ha generado un código binario combinado (respetando nuestro fuente original), añadiendo los nuevos aspectos

La verdad que es sorprendente y profundizaremos con usos más avanzados porque creo que esta tecnología tiene mucho futuro.....

[Sobre el Autor..](#)

Si desea contratar formación, consultoría o desarrollo de piezas a medida puede contactar con

Formación en nuevas tecnologías

[Autentia S.L.](#) Somos expertos en:
J2EE, C++, OOP, UML, Vignette, Creatividad ..
 y muchas otras cosas

Nuevo servicio de notificaciones

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales, inserta tu dirección de correo en el siguiente formulario.

Subscribirse a Novedades	
e-mail	
	Enviar

Otros Tutoriales Recomendados ([También ver todos](#))

Nombre Corto

[Medida del Rendimiento en aplicaciones J2EE](#)

[Reingeniería JDO con Druid](#)

[JDO con OJB](#)

[Cachear porciones de JSPs](#)

[Novedades en Java 1.5](#)

[Soporte de Asserts en Java 1.4.x](#)

[Mensajes multi-idioma en Java](#)

[CMP Entity Beans y MySql](#)

[Herramientas Gratuitas UML](#)

[Escritura log con Fichero UDP y JMS](#)

Descripción

Os mostramos como medir el rendimiento de vuestras aplicaciones Java J2E

Os mostramos como crear vuestras clases y descriptores JDO, de tablas existentes, con la herramienta gratuita Druid.

Os mostramos como configurar el entorno OJB de apache para construir la primera aplicación JDO

En este tutorial os enseñamos como incrementar increíblemente el rendimiento de vuestro Web basado en tecnología JSP con el FrameWork de cache OSCACHE

Ya está disponible la versión Beta del J2SDK 1.5. Os mostramos algunas de las nuevas características introducidas en el lenguaje Java: Clases genéricas, enumeraciones, bucles simplificados, etc.

Os mostramos como utilizar los asserts en Java (disponibles a partir de la versión 1.4)

Os mostramos como aprovechar las características multilenguaje de Java, usando las clases: Locate, ResourceBundle, MessageFormat, etc. Fundamental para un correcto diseño ...

Os mostramos como crear un Entity Bean con persistencia controlada por el servidor, configurado para usar MySql

Os mostramos como obtener algunas herramientas gratuitas UML, ArgoUML y Poseidon.

Os mostramos ejemplos para cuantificar el coste de escritura de Logs por pantalla, fichero, UDP y JMS (describiendo como configurar el entorno)

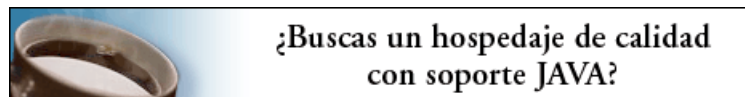
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

Los contenidos y comentarios de los tutoriales son responsabilidad de sus respectivos autores.

En algún caso se puede hacer referencia a marcas o nombres cuya propiedad y derechos es de sus respectivos dueños. Si algún afectado desea que incorporemos alguna reseña específica, no tiene más que solicitarlo.

Si alguien encuentra algún problema con la información publicada en este Web, rogamos que informe al administrador rcanales@adictosaltrabajo.com para su resolución.

[Patrocinados por enredados.com Hosting en Castellano con soporte Java/J2EE](#)



www.AdictosAlTrabajo.com Optimizado 800X600