

¿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)



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

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

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

Curso Web J2EE

Curso Avanzado en Desarrollo Web con J2EE
www.eps.mondragon.edu/caj2ee

XML to PDF in Java & J2EE

Output PDF, PCL5, HTML in Java J2EE
 Websphere, Weblogic, Tomcat, Jetty.
www.scriptura-xsl.com

Master Java J2ee Oracle

Prácticas laborales 100% aseguradas Nuevo temario de Struts. Trabaja ya
www.grupoatrium.com

Anuncios Goooooogle

Anunciarse en este sitio

Desarrollo Gráfico ANT

Cuando construimos aplicaciones complejas debemos tratar de automatizar los máximo posible tareas mecánicas como:

- Extracción de fuentes de un repositorio (por ejemplo [CVS](#))
- Compilación de nuestros fuentes
- Creación de los ficheros temporales
- Empaquetamiento de binarios para distintos entornos (publico, privado, intranet, etc.)
- Construcción de componentes complejos (como [EJBs](#))
- Ejecución de pruebas unitarias
- Generación de la documentación
- Despliegue de componentes
- Borrado de ficheros temporales
- Y muchas cosas más

Para esto, podemos utilizar ANT ([ver tutorial](#)) e incluso otras opciones que van a dar mucho que hablar como [MAVEN](#) (tenemos un tutorial en el horno).

Para trabajar con ANT, podemos hacerlo a mano o utilizar herramientas específicas como [Antelope](#) o las características avanzadas de nuestros entornos habituales (NetBeans en nuestro caso)

Antelope

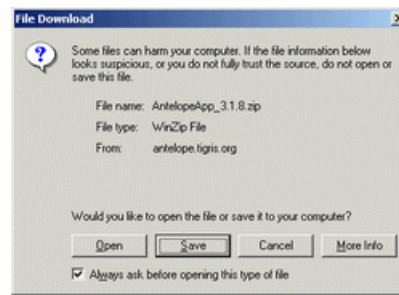
Antelope forma parte de www.Tigris.org que viene a ser como www.sourceforge.org (repositorio de proyectos abiertos) pero un poquito más acotado....



Podemos ir al Web y acceder al link de la guía, que es sorprendentemente buena...

<http://antelope.tigris.org/nonav/docs/manual/index.html>

En el área de Documents & Files podemos descargarnos los binarios



Y, estando seguros de tener instalado ANT, podemos ejecutar el comando para arrancar nuestra herramienta.

`ant -f run.xml`

```
C:\java\AntelopeApp_3.1.8>dir
Volume in drive C has no label.
Volume Serial Number is 7C92-5811

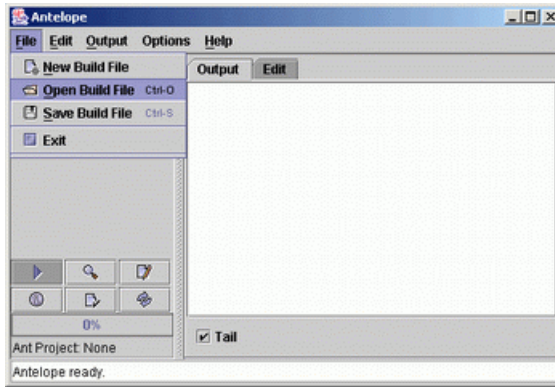
Directory of C:\java\AntelopeApp_3.1.8

06/15/2004 08:37a <DIR> .
06/15/2004 08:37a <DIR> ..
06/06/2004 04:07a 808,495 AntelopeApp_3.1.8.jar
06/06/2004 04:07a 21,663 ChangeLog.txt
06/06/2004 04:07a 350 install.txt
06/06/2004 04:07a 2,258 license.txt
05/07/2004 08:23p 2,363 run.xml
5 File(s) 835,129 bytes
2 Dir(s) 15,535,435,776 bytes free

C:\java\AntelopeApp_3.1.8>echo %ANT_HOME%
C:\java\apache-ant-1.6.1

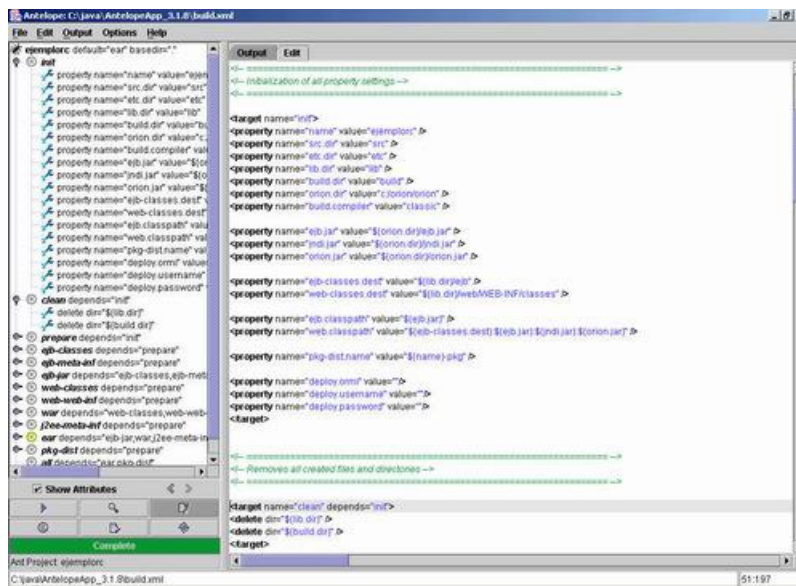
C:\java\AntelopeApp_3.1.8>ant -f run.xml
```

Nos aparece un menú gráfico donde podemos, de un modo bastante intuitivo, gestionar nuestros ficheros ANT (normalmente con el nombre build.xml)



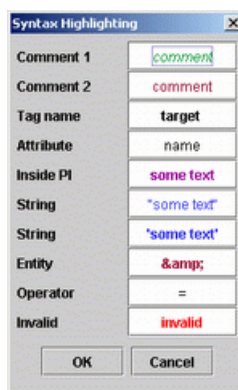
Vamos a abrir uno de los build.xml reales que hemos utilizado anteriormente en este Web ([EJB's y Orion](#).)

Podemos ver, a través del árbol de navegación, las distintas tareas y sus atributos.



Cambiando de lengüeta podemos ejecutar directamente los comandos.

Una de las cosas más simples pero que más nos ayuda es la sintaxis personaliza con colores.



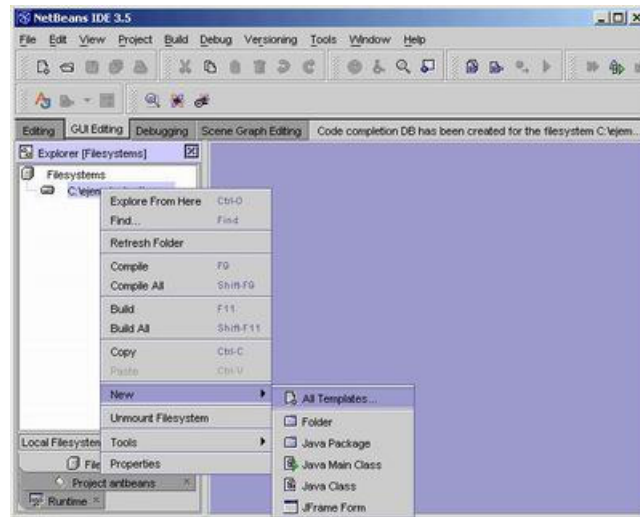
NetBeans

Uno de mis entornos favoritos para el trabajo es NetBeans (para impartición de cursos porque para el trabajo en proyectos me parece igual o incluso mejor eclipse). Dentro de NetBeans, tenemos capacidades avanzadas de creación y ejecución de Scripts ANT y os vamos a mostrar lo sencillo que es

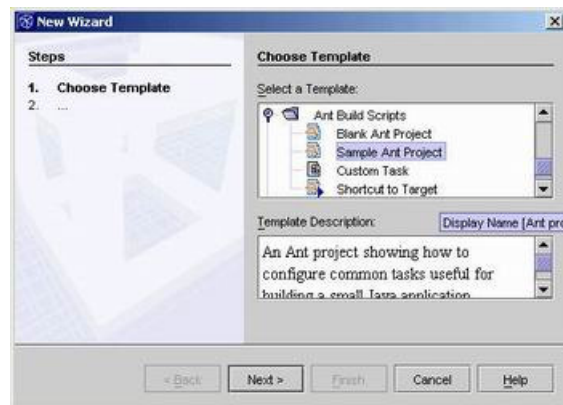
Creemos un nuevo proyecto (y asociamos el filesystem deseado, pinchando con el botón derecho sobre el proyecto creado)



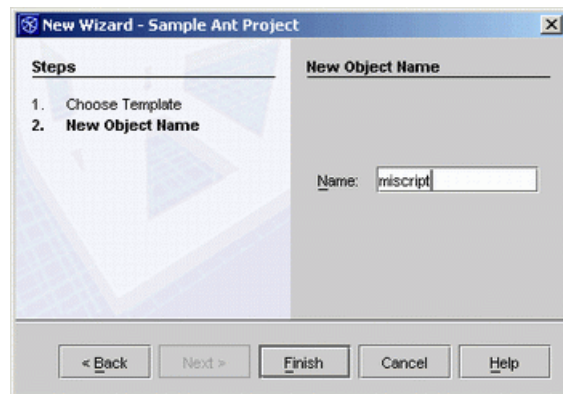
Elegimos el nuevo tipo de elemento a insertar



Elegimos las relacionadas con Ant (si no las veis, ejecutar el Tools->Update para que se conecte al Web de NetBeans)



Elegimos el nombre del Script



Y se nos genera automáticamente algo como esto.

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- Written to assume that classpath is rooted in the current directory. -->
<!-- So this should be OK if you make this script in the root of a filesystem. -->
<!-- If not, you may prefer to adjust the basedir, or move some directories around. -->
<!-- The idea is that both Ant and NetBeans have to know what the package root is -->
<!-- for the classes in your application. -->
<project basedir="." default="all" name="myapp">

<!-- Don't worry if you don't know the Ant syntax completely or need help on some tasks! -->
<!-- The standard Ant documentation can be downloaded from AutoUpdate and -->
<!-- and then you can access the help in menu Help | Help Sets | Ant 1.5.1 Manual. -->

<target name="init">
<!-- You can set up any variables you want used throughout the script here. -->
<property name="hello" value="world"/>
<!-- To use e.g. Jikes, uncomment this line. -->
<!-- (Or make the same change in Tools | Options | Ant Settings | Properties.) -->
<!-- <property name="build.compiler" value="jikes"/> -->
```

```

<!-- You might like to set up some overridable paths, etc.: -->
<!-- <property name="mylib" value="../lib/mylib.jar"/> -->
</target>

<target depends="init" name="compile">
<!-- Both srcdir and destdir should be package roots. -->
<!-- They could be different of course; in that case NetBeans can also be set -->
<!-- up to compile to a different filesystem in the same way; see Compiler Types: -->
<javac debug="true" deprecation="true" destdir="." srcdir=".">
<!-- To add something to the classpath: -->
<!-- <classpath><pathelement location="{mylib}"/></classpath> -->
<!-- To exclude some files: -->
<!-- <exclude name="com/foo/SomeFile.java"/><exclude name="com/foo/somepackage"/> -->
</javac>
</target>

<target depends="init,compile" name="jar">
<!-- To make a standalone app: -->
<!-- 1. Create a myapp.mf manifest somewhere. -->
<!-- 2. Put in it two lines: -->
<!-- Manifest-Version: 1.0 -->
<!-- Main-Class: com.foo.Main -->
<!-- 3. Pass to <jar>: manifest="myapp.mf" -->
<jar basedir="." compress="true" jarfile="myapp.jar">
<exclude name="**/*.java"/>
<exclude name="**/*.form"/>
<exclude name="myapp.mf"/>
<exclude name="myapp.jar"/>
<exclude name="apidoc"/>
</jar>
</target>

<target depends="init,jar" description="Build everything." name="all">
<echo message="Application built. Hello ${hello}!"/>
</target>

<target depends="init,all" description="Try running it." name="test">
<java classname="com.foo.Main" failonerror="true" fork="true">
<classpath>
<pathelement location="."/>
</classpath>
<!-- Pass some args, perhaps: -->
<arg value="-myfile"/>
<!-- Will be given as an absolute path: -->
<arg file="myfile.txt"/>
</java>
</target>

<target depends="init" description="Javadoc for my API." name="javadoc">
<mkdir dir="apidoc"/>
<javadoc destdir="apidoc" packagenames="com.foo.mylib.*,com.foo.myapi.*">
<sourcepath>
<pathelement location="."/>
</sourcepath>
</javadoc>
</target>

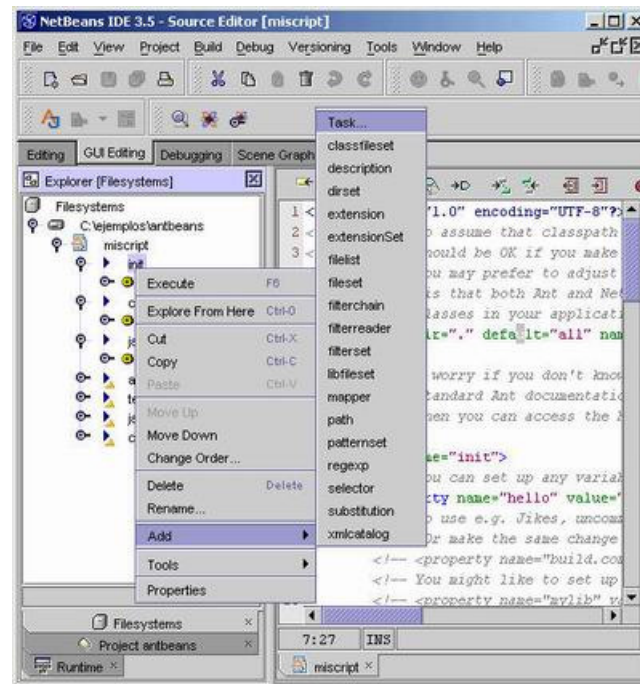
<target depends="init" description="Clean all build products." name="clean">
<delete>
<fileset dir=".">
<include name="**/*.class"/>
</fileset>
</delete>
<delete file="myapp.jar"/>
<delete dir="apidoc"/>
</target>

</project>

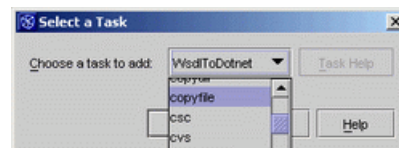
```

Realizar las modificaciones más básicas, necesarias habitualmente, es tan sencillo como quitar algunas de las etiquetas de comentario.

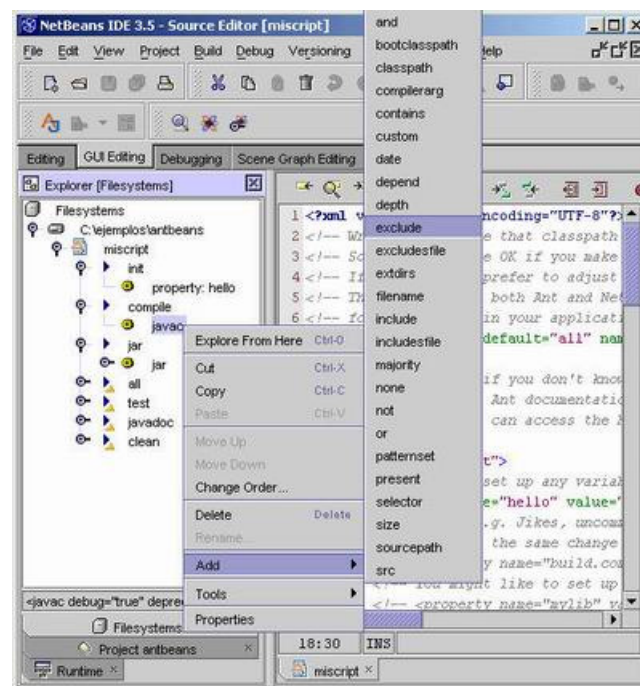
Podemos también, gráficamente haciendo uso del botón derecho sobre cada uno de los targets, seleccionar gráficamente las tareas...



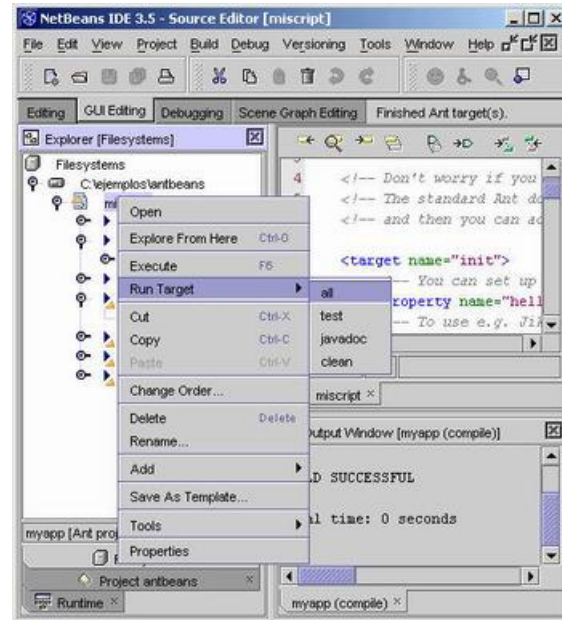
Podemos elegir otras (ver opción marcada) y nos aparecerá un desplegable completo



Si ahora pinchamos sobre una tarea específica, podemos acceder gráficamente a todos sus modificadores



Con el botón derecho podemos ejecutar el target deseado

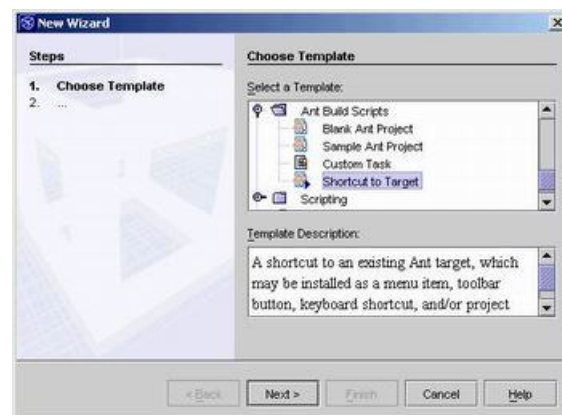


Como curiosidad, podéis comprobar que solo se ven los target que disponen de descripción:

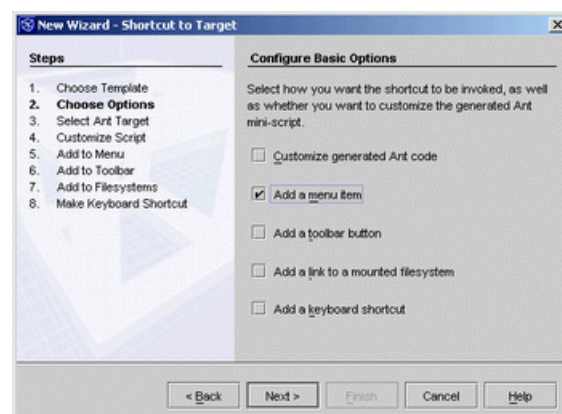
```
<target depends="init" description="Clean all build products." name="clean">
```

Una cosa muy interesante que podemos hacer es añadir uno de los targset como línea de menú (o botón u otro elemento del entorno)

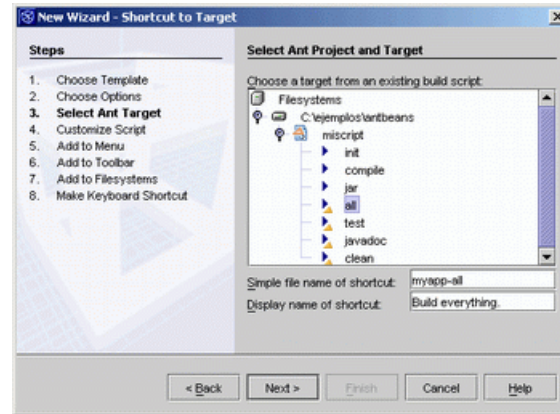
Añadimos un nuevo elemento al proyecto y seleccionamos un Shortcut



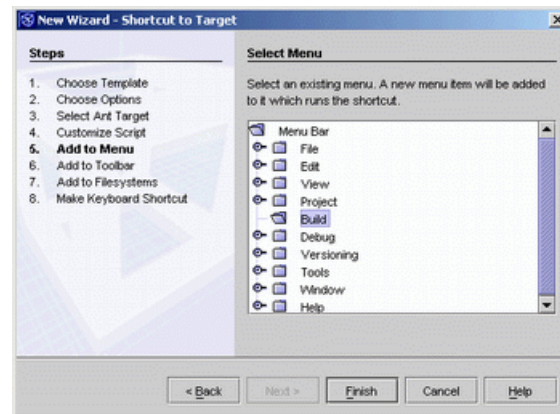
Elegimos un elemento de menú



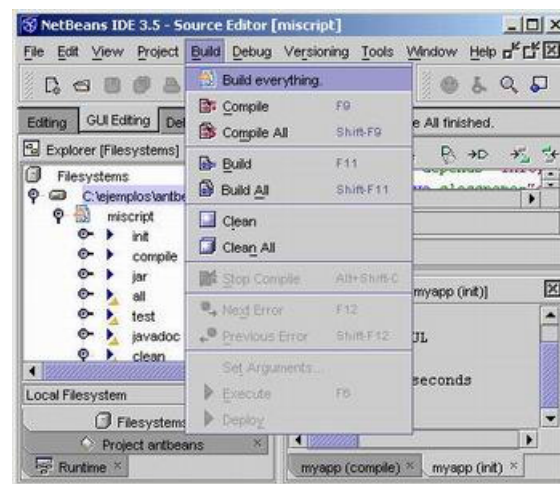
Seleccionamos el Target



Elegimos el menú



Y ya lo tenemos, un menú nuevo con nuestro comando.



Conclusiones

Herramientas hay muchas y además gratuitas. En función de nuestro contexto pueden ser más adecuadas unas que otras.

Si disponemos de poca memoria es probable que herramientas tipo Antelope nos saque de un apuro de un modo rápido. Si no tenemos estas limitaciones, podemos utilizar NetBeans como entorno integrado de desarrollo (u otras herramientas como eclipse).... que es mucho más potente de lo que parece.

Todo jefe de proyecto debería tratar de liberar a uno de sus recursos del trabajo del día a día (de un modo esporádico) para que aprendiera nuevas técnicas de desarrollo y así aumentar la productividad del equipo completo. Es fácil que si no hacemos esto, nuestra productividad sea injustificadamente baja por la falta de realimentación de conocimientos. La nuevas tecnologías requieren formación continua

[Sobre el Autor ...](#)

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

Creatividad Internet

[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	
	<input type="button" value="Enviar"/>

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

Nombre Corto	Descripción
Temperatura del PC y Acoplamiento Electromagnético	Os mostramos algunos programas para ver en pantalla la temperatura de vuestra CPU y velocidad de ventiladores así como consejos de como proteger a vuestro PC de campos electromagnéticos.
Administrar UNIX mediante Webmin	En este artículo os mostramos como administrar una máquina UNIX, mediante un interfaz web, con Webmin
AspectJ, Programación con Aspectos	Os mostramos como configurar AspectJ (extensión Java para la programación basada en aspectos) y un pequeño ejemplo para medir la velocidad de una función sin alterar su código.
Decompilar Java	Os mostramos como recuperar el fuente de vuestro código a partir de los ficheros compilados .class
Introducción a log4j	En un desarrollo Java es vital normalizar los logs para posteriormente poder depurar el funcionamiento de nuestra aplicación. Os mostramos como usar Log4J.
Test con JUnit	Cuando se hacen desarrollo profesionales, no basta con hacer los programas, hay que asegurarse de que van a funcionar. Una de las técnicas más seguras es crear aplicaciones que incluyan el código para autoprobarse. Os mostramos como usar JUnit
Novedades en Java 1.5	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.
Generación de Informes pdf con DataVision	Os mostramos como simplificar la creación de informes un múltiples formatos, incluyendo pdf, con una fantástica herramienta visual (DataVision). Os mostramos como integrarla con Servlets
Introducción a ANT	En el mundo Java, la compilación, verificación e instalación de aplicaciones se ha normalizado con este potente paquete llamado ANT.
Repositorio CVS en Windows	Os mostramos como montar un servidor para el control de versiones CVS en Windows así como acceder a él a través de WinCVS

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](#)

