

¿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)

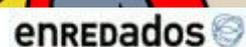
Dirección de Proyectos Informáticos.
 Metodologías ágiles
 Patrones de diseño
 TDD

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

Adictos AL Trabajo



Hosting patrocinado por



[Inicio](#)

[Quienes somos](#)

[Tutoriales](#)

[Formación](#)

[Empleo](#)

[Colabora](#)

[Comunidad](#)

[Libro de Visitas](#)

[Comic](#)

Catálogo de servicios Autentia (PDF 6,2MB)



En formato comic...

Google™

- Web
- www.adictosaltrabaj

Buscar

Últimos tutoriales

2008-10-31
[Planificación de tareas en Java mediante Quartz](#)

2008-10-31
[GrassGames eBook Reader: leer PDF en pantalla cómodamente](#)

2008-10-30
[Autenticación y Autorización mediante JAAS](#)

2008-10-27
[Web Services con Estado](#)

2008-10-24
[Web Services con Axis2. Configuración y ejemplo](#)

2008-10-22
[Migración de JSP a Facelets](#)

2008-10-22
[Rock Band Wii en tu PC](#)

2008-10-18
[Cobertura: Como comprobar cuanto código prueban nuestros test](#)

2008-10-17
[maven-license-plugin:](#)

[cómo gestionar la licencia de nuestros ficheros fuentes](#)

2008-10-10
[Cypal Studio: plugin de GWT para Eclipse](#)

Últimas ofertas de empleo

2008-10-30
[Comercial - Ventas - BARCELONA.](#)

2008-10-30
[T. Información - Analista / Programador - BARCELONA.](#)

2008-10-27
[T. Información - Analista / Programador - CIUDAD REAL.](#)

2008-10-03
[Marketing - Experto en Marketing - MADRID.](#)

2008-10-01
[Atención a cliente - Call Center - MADRID.](#)

Anuncios Google

[Quartz Glass](#)

[Fused Quartz](#)

[Quartz Boat](#)

[Quartz Crucible](#)

NUEVO ¿Quieres saber cuánto ganas en relación al mercado? pincha aquí...

[Ver cursos que ofrece Autentia](#)

[Descargar comics en PDF y alta resolución](#)



[¡NUEVO!] 2008-09-01



2008-07-31



2008-07-08

2008-06-22

Estamos escribiendo un libro sobre la profesión informática y estas viñetas formarán parte de él. Puedes opinar en la sección [comic](#).

Tutorial desarrollado por



Carlos Pérez **García**

Creador de **MobileTest**, un complemento educativo para los profesores y sus alumnos.

Consultor tecnológico en el desarrollo de proyectos informáticos.

Ingeniero Técnico en

Informática *

Puedes encontrarme en [Autentia](#)

Somos expertos en Java/J2EE

Catálogo de servicios de Autentia

[Descargar \(6,2 MB\)](#)

[Descargar en versión comic \(17 MB\)](#)

[AdictosAlTrabajo.com](#) es el Web de difusión de conocimiento de [Autentia](#).



[Catálogo de cursos](#)

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

Fecha de creación del tutorial: 2008-10-31

Planificación de tareas en Java mediante Quartz

- [Introducción](#)
- [Características principales](#)
- [Principales clases e interfaces](#)
- [Ejemplo. Planificación de tareas en modo programático.](#)
- [Ejemplo. Planificación de tareas en modo declarativo.](#)
- [Aplicación Web para gestionar nuestras tareas \(modo declarativo\)](#)
- [Direcciones de interés \(en inglés\)](#)
- [Conclusiones](#)

Introducción

Normalmente los negocios requieren que en determinados momentos se ejecuten tareas de forma automática, por ejemplo: "todos los viernes a las 15:00 enviar informes a clientes".

Crear un sistema robusto y completo que de soporte a la ejecución de esas tareas, además de no ser una tarea fácil, sería como reinventar la rueda pues ya disponemos de frameworks maduros de libre uso como el que vamos a presentar en este tutorial.

Quartz es un framework open source, con licencia Apache 2.0 para la **planificación y gestión de tareas**.

Es usado activamente en conocidos proyectos y organizaciones como JBoss, Cocoon, Apache Jakarta,... [ver más](#)

Características principales

- Válido para aplicaciones tanto J2EE como J2SE.
- Planificación flexible de tareas, por ejemplo: el primer lunes de Enero de cada año a las 17:45.
- Mantenimiento del estado de las tareas incluso en caso de fallos y reinicios de máquinas.
- Posibilidad de participar en transacciones JTA.
- Posibilidad de trabajar en modo Clúster.
- Proporciona un completo API, con muchas clases de utilidad y muchos tipos Listener (JobListener, TriggerListener y SchedulerListener).

Principales clases e interfaces

A continuación describimos los componentes que lo componen: `Job`, `JobDetail`, `Trigger`, `JobStore`, `Scheduler`.

`org.quartz.Job`

Definir una tarea es tan sencillo como implementar la interface `org.quartz.Job` cuya definición se muestra a continuación:

```
view plain print ?
01. package org.quartz;
02.
03. public interface Job {
04.     public void execute(JobExecutionContext context) throws JobExecuteException;
05. }
```

Como podéis observar, simplemente debemos implementar un método y lanzar una excepción en caso de error para que Quartz reintente su ejecución o no, en función de la configuración que especifiquemos.

`org.quartz.JobDetail`

Es una clase que almacena propiedades de una determinada tarea.

Las tareas se clasifican en grupos de tareas y cada tarea tiene un nombre único dentro del grupo. `JobDetail` define éstas y otras propiedades.

Gracias a esta clasificación podemos pausar, iniciar, detener, etc. tareas o grupos de tareas de manera independiente al resto.

`org.quartz.Trigger`

Es una clase abstracta que define los instantes en que la tarea debe ser ejecutada, por ejemplo todos los lunes a las 16:00 de la tarde.

Existen varias implementaciones de esta clase pero las más usadas son:

`org.quartz.SimpleTrigger`

Permite especificar ejecuciones de tareas teniendo en cuenta los siguientes parámetros: fecha, hora, nº de repeticiones e intervalo entre repeticiones. Por ejemplo, ejecutar la tarea X 3 veces el día 13/12/1976 a las 13:30 con 40 minutos entre cada ejecución.

Para que se haga una idea, la firma de uno de sus constructores es:

```
public SimpleTrigger(String name, String group, Date startTime, Date endTime, int r
```

`org.quartz.CronTrigger`

Es el más utilizado, pues permite especificar mediante expresiones más complejas los instantes en los que deben ejecutarse las tareas.

Por ejemplo, todos los días 14 que caigan en jueves cada 5 minutos desde las 14:00 hasta las 18:00.

Aunque existe mucha funcionalidad de utilidad que permiten hacernos la vida más fácil a la hora de crear el `CronTrigger`, yo suelo usar un constructor que acepta una cadena de texto representando una expresión de tipo Cron de Unix: **segundos, minutos, horas, días del mes, meses, días de la semana, [año]** (observe que todos son obligatorios a excepción de el año).

```
"0 0 23 * * ?"           La tarea será ejecutada todos los días a las 23:00
"0 20 15 * * 1 2007"    La tarea será ejecutada todos los domingos del año 2007 a las 15:20
```

Para más información acerca de `CronTrigger` haga [clic aquí](#).

org.quartz.JobStore

Interface para manejar el almacenamiento y recuperación de la información de planificación (tareas, triggers, etc).

Implementaciones de `org.quartz.JobStore`:

`org.quartz.simpl.RAMJobStore`

Es la implementación de por defecto, almacena la información en memoria RAM por lo que cuando la aplicación finaliza no se guarda el estado.

`org.quartz.impl.jdbcjobstore.JobStoreTX`

Implementación que almacena la información en una base de datos a través de JDBC. Diseñado para entornos no transaccionales.

`org.quartz.impl.jdbcjobstore.JobStoreCMP`

Implementación que almacena la información en una base de datos a través de JDBC. Diseñado para entornos transaccionales.

La elección del JobStore se define a través de la propiedad `org.quartz.jobStore.class` de el archivo de propiedades de Quartz `quartz.properties`, por ejemplo.

```
org.quartz.jobStore.class = org.quartz.simpl.RAMJobStore
```

org.quartz.Scheduler

Su funcionalidad es almacenar y planificar las tareas (Job) en base a los Triggers, recuperar tareas fallidas, realizar reintentos y gestionar el estado del sistema de planificación.

Ejemplo. Planificación de tareas en modo programático.

A continuación y a modo de ejemplo vamos a crear una aplicación que lance una tarea cada 5 segundos.

Aunque la tarea podría realizar cualquier operación compleja, en este caso simplemente mostrará un mensaje por pantalla.

```
com.autentia.tutoriales.quartz.StatusMonitorJob.java
```

Implementación de la tarea:

view plain print ?

```
01. package com.autentia.tutoriales.quartz;
02.
03. /**
04.  * Tarea de ejemplo.
05.  * Guarda en un archivo la memoria total y libre de la máquina virtual
06.  * @author Carlos García. Autentia Real Business Solutions
07.  * @see http://www.mobiletest.es
08.  */
09. public class StatusMonitorJob implements org.quartz.Job {
10.     public static final String TARGET_FILE_PROP = "outputFile";
11.
12.     /*
13.      * La tarea es ejecutada
14.      * @see org.quartz.Job#execute(org.quartz.JobExecutionContext)
15.      */
16.     public void execute(org.quartz.JobExecutionContext context) throws org.quartz.JobExecutionException {
17.         org.apache.log4j.Logger logger = org.apache.log4j.Logger.getLogger(StatusMonitorJob.class);
18.
19.         if (logger.isDebugEnabled()){
20.             logger.debug("Executing");
21.         }
22.
23.         StringBuffer statusLine = new StringBuffer(128);
24.         java.io.PrintWriter writer = null;
25.
26.         // Obtenemos la propiedad que especifica el archivo donde se desea guardar el estado
27.         // valor por defecto en caso de que no se haya especificado
28.         org.quartz.JobDataMap properties = context.getJobDetail().getJobDataMap();
29.         String outputFile = properties.getString(StatusMonitorJob.TARGET_FILE_PROP);
30.         if ((outputFile == null) || ("".equals(outputFile.trim()))){
31.             outputFile = "status.txt";
32.         }
33.
34.         try {
35.             // Escribimos en el archivo una línea de estado con formato: 1
36.             writer = new java.io.PrintWriter(outputFile, true);
37.
38.             statusLine.append(System.currentTimeMillis());
39.             statusLine.append('\t');
40.             statusLine.append(Runtime.getRuntime().totalMemory());
41.             statusLine.append('\t');
42.             statusLine.append(Runtime.getRuntime().freeMemory());
43.             statusLine.append(System.getProperty("line.separator"));
44.
45.             if (logger.isDebugEnabled()){
46.                 logger.debug("Storing VM state to file: " + outputFile);
47.             }
48.
49.             writer.write(statusLine.toString());
50.         } catch (java.io.IOException e) {
51.             logger.error("Error ", e);
52.         } finally {
53.             try {
54.                 writer.close();
55.             } catch (Exception ex){}
56.         }
57.     }
58.
59. }
60.
```

com.autentia.tutoriales.quartz.StatusMonitor.java

Aplicación de escritorio que realiza la planificación de la tarea:

view plain print ?

```
01. package com.autentia.tutoriales.quartz;
02.
03. /**
04.  * Planificación de tareas en Java mediante Quartz
05.  * @author Carlos García. Autentia Real Business Solutions
06.  * @see http://www.mobiletest.es
07.  */
08. public class StatusMonitor {
09.     private org.quartz.Scheduler scheduler;
10.
11.     /**
12.      * Instancia, configura e inicia la tarea para ser ejecutada cada 5 s
13.      * @throws org.quartz.SchedulerException
14.      */
15.     public void start() throws org.quartz.SchedulerException {
16.         try {
17.             // Definimos la tarea (nombre de la tarea, nombre del grupo de
18.             org.quartz.JobDetail jobDetail = new org.quartz.JobDetail("Status
19.
20.             // Configuramos los parámetros de la tarea, en este caso le de
21.             jobDetail.getJobDataMap().put(StatusMonitorJob.TARGET_FILE_PROP, "vr
22.
23.             // Configuramos el trigger que avisará al planificador de cuar
24.             org.quartz.CronTrigger trigger = new org.quartz.CronTrigger("Statu
25.
26.             // Obtenemos el planificador
27.             scheduler = org.quartz.impl.StdSchedulerFactory.getDefaultScheduler(
28.
29.             // La tarea definida en JobDetail será ejecutada en los instar
30.             scheduler.scheduleJob(jobDetail, trigger);
31.
32.             // Iniciamos las tareas planificadas en el scheduler
33.             scheduler.start();
34.         } catch (java.text.ParseException e) {
35.             // No se dará
36.         }
37.     }
38.
39.     /**
40.      * Detiene el proceso de planificación
41.      */
42.     public void stop(){
43.         try {
44.             scheduler.shutdown();
45.         } catch (Exception ex) {
46.             // Nada
47.         }
48.     }
49.
50.     /**
51.      * Punto de entrada a la aplicación
52.      */
53.     public static void main(String[] args) {
54.         StatusMonitor monitor = new StatusMonitor();
55.
56.         try {
57.             monitor.start();
58.
59.             System.out.println("Pausa... pulse una tecla para finalizar la aplic
60.             System.in.read();
61.
62.             monitor.stop();
63.         } catch (Exception ex) {
64.             System.err.println(ex);
65.         }
66.     }
67.
68. }
69.
```

Ejemplo. Planificación de tareas en modo declarativo.

Por defecto, en la configuración de Quartz puede especificar tareas externamente a la aplicación en un

Este método le ofrece una gran flexibilidad, ya que podrá añadir o eliminar tareas sin realizar la gestión de planificación dentro de la aplicación... simplemente deberá copiar las clases en el CLASSPATH de la máquina virtual Java.

quartz_jobs.xml

```

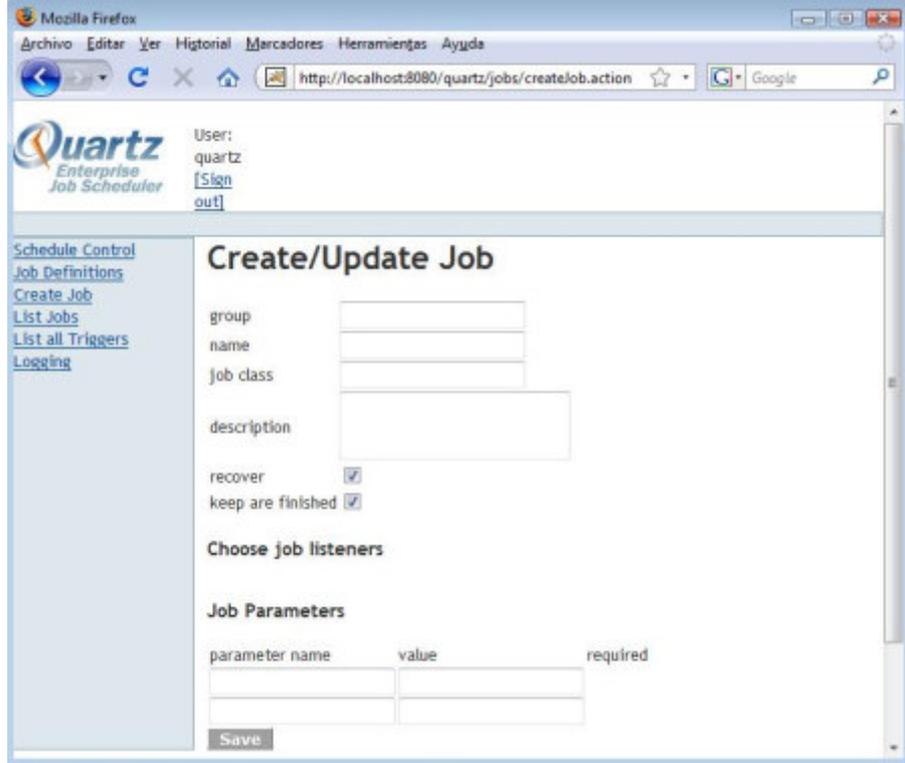
view plain print ?
01. <?xml version="1.0" encoding="utf-8"?>
02. <quartz>
03.   <job>
04.     <job-detail>
05.       <name>QuartzTarea1</name>
06.       <group>DEFAULT</group>
07.       <description>Esta es una tarea de ejemplo del uso de Quartz</descrip
08.       <job-class>com.autentia.tutoriales.quartz.MyTask</job-class>
09.       <volatility>>false</volatility>
10.       <durability>>false</durability>
11.       <recover>>false</recover>
12.       <job-data-map allows-transient-data="true">
13.         <entry>
14.           <key>propiedad1</key>
15.           <value>valor_propiedad1</value>
16.         </entry>
17.       </job-data-map>
18.     </job-detail>
19.     <trigger>
20.       <simple>
21.         <name>QuartzTrigger1</name>
22.         <group>DEFAULT</group>
23.         <job-name>QuartzTrigger1</job-name>
24.         <job-group>DEFAULT</job-group>
25.         <start-time>2005-09-17 0:10:00 PM</start-time>
26.         <repeat-count>-1</repeat-count>
27.         <repeat-interval>1000</repeat-interval>
28.       </simple>
29.     </trigger>
30.   </job>
31. </quartz>
32.

```

Aplicación Web para gestionar nuestras tareas (modo declarativo)

En el siguiente enlace <http://prdownloads.sourceforge.net/quartz/quartz-web-app.zip> puede obtener una útil aplicación Web para gestionar las tareas a través de una completa aplicación Web. Su uso queda fuera del alcance de este tutorial...

Los datos de acceso de por defecto son quartz tanto para el nombre de usuario como para la contraseña.



Direcciones de interés (en inglés)

- [Página principal del proyecto Quartz](#)
- [Tutorial de Quartz](#)
- [¿Qué es Quartz?](#)
- [Job Scheduling in Java](#)
- [Using Quartz Scheduler in a Java EE Web Application](#)
- [The Quartz web application](#)

Conclusiones

Quartz es un Framework maduro y muy extendido, que nos proporciona ampliar la funcionalidad de nuestros proyectos que basan en tareas de una forma fácil y flexible.

Sin lugar a dudas, Quartz es una opción a tener en cuenta en estos casos... Para más información diríjase a la documentación del proyecto.

Un saludo.

Carlos García. Creador de [MobileTest](#), un complemento educativo para los profesores y sus alumnos.

- Puedes opinar sobre este tutorial [haciendo clic aquí](#).
- Puedes firmar en nuestro libro de visitas [haciendo clic aquí](#).
- Puedes asociarte al grupo [AdictosAlTrabajo en XING](#) [haciendo clic aquí](#).

- [Añadir a favoritos Technorati](#). 



Esta obra está licenciada bajo [licencia Creative Commons de Reconocimiento-No comercial-Sin obras derivadas 2.5](#)

Recuerda

[Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#)). Somos expertos en: J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ... y muchas otras cosas.

¿Nos vas a tener en cuenta cuando necesites

consultoría o formación en tu empresa?, ¿Vas a ser tan generoso con nosotros como lo tratamos de ser con vosotros?

Somos pocos, somos buenos, estamos motivados y nos gusta lo que hacemos ...

Autentia = Soporte a Desarrollo & Formación.

info@autentia.com

Gestión de contenidos

Servicio de notificaciones:

Si deseas que te enviemos un correo electrónico cuando introduzcamos nuevos tutoriales.

Formulario de subscripción a novedades:

E-mail

Aceptar

Tutoriales recomendados

Nombre	Resumen	Fecha	Visitas	pdf
OpenCms	Este tutorial nos va a mostrar una serie de documentos enviados por OpenCmsHispano sobre OpenCms, como instalarlo, como crear un portal y diferentes funcionalidades fundamentales	2007-04-02	2642	pdf
Gestionando nuestro tiempo, en GNU/Linux Debian	En este tutorial vamos a ver el funcionamiento de una herramienta llamada Karm que nos permitirá dar de alta varias tareas y subtareas, y medir el tiempo que dedicamos a cada una.	2007-08-10	1506	pdf
Cómo Crear una Tarea Ant	Este tutorial tiene como objetivo explicar cómo crear una tarea nueva Ant y utilizarla en un script.	2007-07-23	2495	pdf
Sincronización de carpetas con SyncToy de Microsoft	En este tutorial se va a explicar el funcionamiento de una herramienta de Microsoft, SyncToy, que nos permitirá mantener sincronizadas dos carpetas diferentes de diferentes máquinas.	2006-11-21	3264	pdf
Recomendaciones y trucos para mejorar la presencia de páginas web en buscadores	Con este tutorial, se intenta presentar algunos consejos, trucos y recomendaciones para mejorar el posicionamiento y presencia en los buscadores web de Internet (Google, Yahoo, MSN, Altavista, AlltheWeb, etc.) de nuestras páginas web, basándose en distintas	2007-01-02	3668	pdf
Backup Manager: Herramienta de Backup en sistema linux	En este tutorial se explica el funcionamiento de Backup Manager, una herramienta muy completa y sencilla para la realización de copias de seguridad de sistema de ficheros locales y remotos, bases de datos, etc...	2007-03-12	4352	pdf
Apache con Tomcat	En este tutorial vamos a poder ver y comprobar como Apache Web Server y Tomcat pueden convivir perfectamente uno con el otro para así poder aumentar el rendimiento total de nuestra aplicación	2007-04-04	6500	pdf
JSP's y Modelo-Vista-Controlador	En este tutorial os enseñamos como crear un JSP, su relación con los servlets y como crear un ejemplo MVC en Tomcat	2003-06-16	59756	pdf
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.	2003-06-13	33439	pdf
Planificar tareas en JBoss	En este tutorial os enseñaremos a planificar tareas periódicas con JBoss	2006-06-07	6316	pdf

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.

