

¿Qué ofrece Autentia Real Business Solutions S.L?

Somos su empresa de **Soporte a Desarrollo Informático**.
 Ese apoyo que siempre quiso tener...

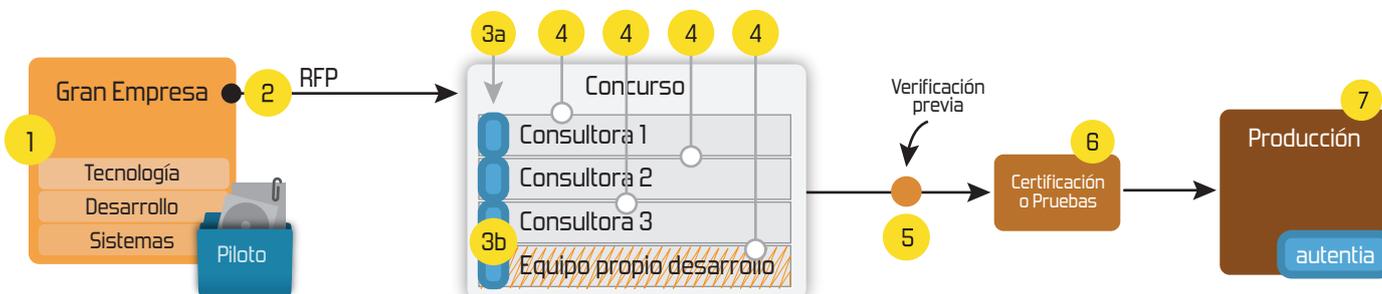
1. Desarrollo de componentes y proyectos a medida



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](#) | [Tutoriales](#) | [Contacte](#)

Tutorial desarrollado por: [Carlos García Pérez](#)

Puedes encontrarme en [Autentia](#)
Somos expertos en **Java/J2EE**
Contacta en info@autentia.com



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

[Firma en nuestro libro de Visitas](#)

[Aplicaciones Monsan](#)

Líneas 803 -8 0 6/7
Profesionalidad Garantía, Servicio
- SMS Premium

[Softeng](#)

Desarrollo soluciones web y
gestión Consultoría informática
Barcelona.

[Instituto Cibernos](#)

Certificaciones y Masters Ccna,
Linux, Java, .Net, Microsoft

[Servicio Técnico](#)

Madrid 91 328 36 00 lavadoras
lavavajillas frigoríficos

Apache Commons Configuration

Introducción

Yo trabajo en [Autentia](#), y en mi trayectoria profesional ligada a la informática he visto como los programadores invertían grandes cantidades de tiempo en reinventar la rueda por desconocimiento de determinadas APIs.

¿ Alguna vez has implementado clases que gestionaran directamente datos de configuración ? Si la respuesta es afirmativa, seguramente hayas reinventado la rueda.

Existen en varias formas estándar de gestionar los datos configuración de nuestros componentes y aplicaciones:

- Mediante la clase `java.util.Properties`
- A través de las clases del paquete `java.util.prefs`.
(API muy potente que permite gestionar datos de configuración de modo herárquico al estilo del "Registro de Windows que recomiendo leerse si el lector la desconoce)

En este pequeño tutorial, voy a nombrar un API que a mi personalmente me parece de una enorme utilidad y potencia para la gestión de datos de configuración. Se trata de [Apache Jakarta Commons Configuration](#).

Apache Jakarta Commons Configuration es un subproyecto de la serie de proyectos Apache Jakarta Commons.

Commons Configuration es un conjunto de proyectos que intentan estandarizar las tareas comunes que casi todas las aplicaciones y componentes realizan normalmente.

Una tarea muy común en cualquier aplicación es la gestión de datos de configuración. Es aquí en donde entra en juego Apache Jakarta Commons Configuration.

Podemos obtener todo lo necesario desde la siguiente dirección: <http://jakarta.apache.org/commons/configuration/>

Básicamente Apache Commons Configuration nos permite:

1. Gestión de los datos de configuración independientemente del lugar en el que esten almacenados. (Ficheros de propiedades, documentos XML, JNDI, Base de datos, etc.).
2. Un potente motor de consultas para realizar búsquedas del valor que tienen determinadas propiedades de configuración.
3. Recarga automática de los datos de configuración en caso de que sean modificados en su lugar físico de origen (Fichero de texto plano, fichero xml, etc)
4. Posibilidad de almacenar los datos de configuración automáticamente en su lugar físico (fichero de texto plano, fichero xml, etc).

Algunos ejemplos

Antes de nada, deseo resaltar que este tutorial es sólo una introducción al API, y que debe documentarse en la Web oficial de API para ver todas las posibilidades que nos proporciona este API.

Ejemplo 1

Supongamos que tenemos el siguiente fichero de propiedades, en el que se definen datos de configuración relacionada con interfaz gráfico de la aplicación:

app.windows.properties:

```
mainmdi.title=Título de mi aplicación de prueba
mainmdi.width=800
mainmdi.height=600
mainmdi.state=maximized

userform.title=Gestión de usuarios
userform.width=320
userform.height=200
userform.state=normal

# También sería válido: language=Castellano,English,French
language=Castellano
language=English
language=French
```

Pues a través de Commons Configuration, podríamos acceder a las propiedades de la siguiente forma:

```
try {
    // Leemos los datos de configuración. El fichero debe estar en el mismo directorio que la aplicac:
    PropertiesConfiguration config = new PropertiesConfiguration("app.windows.properties");

    // La siguiente sentencia imprimirá: 800
    System.out.println(config.getInt("main.mdi.width"));

    // La siguiente sentencia imprimirá: Mi aplicación de prueba
    System.out.println(config.getString("main.mdi.title"));

    // La siguiente sentencia imprimirá: Castellano
    System.out.println(config.getStringArray("language")[0]);

    // La siguiente sentencia configura el objeto para que salve las propiedades
    // automáticamente en su origen (En este caso un fichero) cuando se actualizen
    // o cuando se agregen nuevas propiedades
    config.setAutoSave(true);

    // Añadimos un par de propiedades
    config.addProperty("version", "1.0");
    config.addProperty("author", "Carlos García Pérez");

    // No hace falta gracias a que está establecida la propiedad autoSave
    // config.save();
} catch (ConfigurationException ex){
    System.out.println(ex);
}
```

Ejemplo 2

Ahora vamos a ver las posibilidades de este API para tratar con propiedades Jerárquicas en formato xml.

app.windows.properties.xml

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<gui>
  <mainmdi>
    <title>Mi aplicación de prueba</title>

    <dimension>
      <width>800</width>
      <height>600</height>
    </dimension>
    <state>maximized</state>
  </mainmdi>

  <userform>
    <title>Gestión de Usuarios</title>
    <dimension>
      <width>800</width>
      <height>600</height>
```

```

        </dimension>
        <state>normal</state>
    </userform>

    <printer.form>
        <dimension>
            <width>320</width>
            <height>200</height>
        </dimension>
    </printer.form>

    <languages default="Castellano">
        <language>Castellano</language>
        <language>English</language>
        <language>French</language>
    </languages>
</gui>

```

Ahora podemos acceder a los datos de la siguiente forma. El ejemplo es autodescriptivo y está comentado.

```

try {
    // Leemos los datos de configuración. El fichero debe estar en el mismo directorio que la aplicac:
    XMLConfiguration config = new XMLConfiguration("app.windows.properties.xml");

    // La siguiente sentencia imprimirá: 800
    System.out.println(config.getInt("mainmdi.dimension.width"));

    // La siguiente sentencia imprimirá: Mi aplicación de prueba
    System.out.println(config.getString("mainmdi.title"));

    // La siguiente sentencia imprimirá: Castellano
    System.out.println(config.getString("languages[@default]"));

    // La siguiente sentencia imprimirá: 3
    System.out.println(((java.util.List) config.getProperty("languages.language")).size());

    // La siguiente sentencia imprimirá: French
    System.out.println(config.getString("languages.language(2)"));

    // La siguiente sentencia imprimirá: 320
    // Observe que debido a que el elemento contiene un punto en su nombre
    // debemos acceder a el escapando el punto mediante ..
    System.out.println(config.getInt("printer..form.dimension.width"));

    // La siguiente sentencia configura el objeto para que salve las propiedades
    // automáticamente en su origen (En este caso un fichero) cuando se actualizen
    // o cuando se agregen nuevas propiedades
    config.setAutoSave(true);

    // Cambiamos el juego de caracteres, pues usaremos propiedades
    // que precisan de tildes.
    config.setEncoding("ISO-8859-1");

    // Añadimos un par de propiedades
    // Añadimos la propiedad 1.0 que desciende de la raiz
    config.addProperty("version", "1.0");

    // Añadimos la propiedad Carlos García Pérez que desciende de la raiz
    config.addProperty("author", "Carlos García Pérez");

    // No hace falta gracias a que está establecida la propiedad autoSave establecida
    // config.save();
} catch (ConfigurationException ex){
    System.out.println(ex);
}

```

Otras Características interesantes

Este API también nos da la posibilidad de que se recargen automáticamente los datos de configuración sobre el objeto Configuration en caso de que estos hayan sido modificados.

Para habilitar esta característica deberá indicarlo mediante la siguiente línea de código:

```
config.setReloadingStrategy(new FileChangedReloadingStrategy());
```

Conclusiones

A mi juicio este API es más completo, potente y flexible que el resto de APIS o métodos para gestionar datos de configuració

Aunque java.util.prefs proporciona una funcionalidad y potencia similar a Commons Configurations, los datos de configuració en java.util.prefs sólo pueden ser modificados desde aplicaciones y no manualmente, pues entre otras cosas no se sabe dond

guarda realmente este API los datos de configuración.

Cuando utilizamos objetos Properties para gestionar las propiedades, debemos realizar conversiones de tipos de datos, con e API nos ahorramos ese trabajo.

Por comentar algo negativo, este API depende de otros subproyectos de Jakarta, por lo que debemos incluir sus JARs asociad si queremos usarlo. Por ejemplo, para usar las funciones básicas, el API depende de:

- Apache Jakarta Commons Lang
- Apache Jakarta Commons Collections
- Apache Jakarta Commons Loggin

Puede encontrar más información acerca de las dependencias en la siguiente dirección
<http://jakarta.apache.org/commons/configuration/dependencies.html>



[Puedes opinar sobre este tutorial aquí](#)

Recuerda

que el personal de [Autentia](#) te regala la mayoría del conocimiento aquí compartido ([Ver todos los tutoriales](#))

¿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?

info@autentia.com

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

Autentia = Soporte a Desarrollo & Formación



[Autentia S.L.](#) Somos expertos en:

J2EE, Struts, JSF, C++, OOP, UML, UP, Patrones de diseño ..
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	
<i>e-mail</i>	<input type="text"/>
	<input type="button" value="Enviar"/>

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

Nombre Corto

[Problemas al planificar un proyecto](#)

Descripción

En este tutorial/artículo os presentamos una plantilla modelo (básica) para un proyecto software (orientado a aplicaciones Web/Java OOP) y os comentamos por qué es tan difícil cumplir con un plan de proyecto informático

[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.

[XML básico](#)

Si quieres ver de un modo visual como crear un documento XML, este es tu tutorial. Este es el primero de un conjunto de tutoriales que iremos publicando sobre esta fascinante y amplia tecnología

[Seguridad en Tomcat](#)

Os mostramos como proteger de un modo básico el acceso a recursos dentro de vuestro servidor de componentes Tomcat

[Trabajando con Axis](#)

Utilizando Apache Axis, os mostramos otro interesante tutorial que ilustra su utilización para implementar web services

[JSF en Java Studio Creator 2](#)

En este tutorial os mostramos como realizar una aplicación JSF utilizando la herramienta Java Studio Creator en su segunda versión

[JUnit 4. Pruebas de Software Java](#)

Tutorial que describe como utilizar la herramienta JUnit 4 para realizar pruebas de integridad y errores sobre Java.

[Instalación de Subversion \(SVN\) en Windows XP](#)

En este tutorial os mostramos cómo instalar y utilizar la herramienta SVN en vuestro entorno XP

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](#)

