

# ¿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)

 Powered by 	Hosting Patrocinado por <b>enREDados.com</b> 
---	--

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



**CoNcept**

Lanzado

**TNTConcept versión 0.4.1** ( 04/06/2007)

Desde [Autentia](#) ponemos a vuestra disposición el software que hemos construido (100% gratuito y sin restricciones funcionales) para nuestra gestión interna, llamado TNTConcept (auTeNTia).

Construida con las últimas tecnologías de desarrollo Java/J2EE (Spring, JSF, Acegi, Hibernate, Maven, Subversion, etc.) y disponible en licencia GPL, seguro que a muchos profesionales independientes y PYMES os ayudará a organizar mejor vuestra operativa.

**Las cosas grandes empiezan siendo algo pequeño** ..... Saber más en: <http://tntconcept.sourceforge.net/>

<p><b>Tutorial desarrollado por: <a href="#">Francisco Javier Martínez Páez</a></b></p> <p><b>Puedes encontrarme en <a href="#">Autentia</a></b>  <b>Somos expertos en Java/J2EE</b>  <b>Contacta en:</b>  <b><a href="mailto:fjmpaez@autentia.com">fjmpaez@autentia.com</a></b></p>	<p><b>NUEVO CATÁLOGO DE SERVICIOS DE AUTENTIA (PDF 6,2MB)</b></p> <p><a href="http://www.adictosaltrabajo.com">www.adictosaltrabajo.com</a> es el Web de difusión de conocimiento de <a href="http://www.autentia.com">www.autentia.com</a></p>  <p><b>autentia</b> real business solutions</p> <p><a href="#">Catálogo de cursos</a></p>
--	---

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

[Firma en nuestro libro de Visitas](#) <-----> [Asociarme al grupo AdictosAlTrabajo en eConozco](#)

#### SOFTENG

Desarrollo soluciones web y gestión Consultoría informática Barcelona.  
[www.softeng.es](http://www.softeng.es)

#### Consiga certificación SUN

Java, Solaris, Sun Cluster, JCAPS ¡Y si suspende, repesca gratis!  
[es.sun.com/training](http://es.sun.com/training)

#### Manuales profesionales

AutoCAD, Photoshop, Rhino, SolidEdg fotografía, imagen, diseño, color  
[www.latiendadecad.com](http://www.latiendadecad.com)

#### Matrikon Iberica

Clients & Servers OPC Clientes y Servidores OPC  
[www.matrikonOPC.es](http://www.matrikonOPC.es)

Anuncios Google

## Fecha de creación del tutorial: 2007-07-10

### Ordenar Listas en Java

En este tutorial pretendo mostrar una forma sencilla de ordenar listas en java usando dos métodos estáticos de la clase java.util.Collections (si el lector aún no los sabe) e implementando dos interfaces Comparable y Comparator.

Cuando nosotros creamos una clase de algún tipo, podemos especificar el orden natural de los objetos de esa clase cuando se encuentran en una lista, simplemente implementando el interfaz Comparable que declara el método:

```
int compareTo(Object o);
```

Éste método compara éste objeto con el objeto recibido como parámetro. Éste método ha de devolver un valor negativo si éste objeto es menor que el recibido, 0 si son iguales o un valor positivo si es mayor, según el orden natural de los objetos.

Vamos a mostrar un ejemplo. Voy a crear la clase Persona con los siguientes atributos:

- Apellidos
- Nombre
- DNI
- Fecha de Nacimiento.

He decidido que para mí, la forma natural de ordenar los objetos de la clase Persona viene definido:

- Primeramente por el orden alfabético de sus apellidos
- Después por el orden alfabético de su nombre.
- Y finalmente por su DNI.

Definimos la clase Persona de esta manera:

```
public class Persona implements Comparable {
    private String nombre;
    private String apellidos;
    private String dni;
    private Date fechaNacimiento;

    public String toString() {
        return this.apellidos + " " + this.nombre + " " + this.dni + " "
            + DateFormat.getDateInstance(DateFormat.DEFAULT,
```

```

        Locale.getDefault()).format(this.fechaNacimiento);
    }

    public Persona(String nombre, String apellidos, String dni, Date fechaNacimiento) {
        this.nombre = nombre;
        this.apellidos = apellidos;
        this.dni = dni;
        this.fechaNacimiento = fechaNacimiento;
    }

    public String getApellidos() {
        return apellidos;
    }

    public void setApellidos(String apellidos) {
        this.apellidos = apellidos;
    }

    public String getDni() {
        return dni;
    }

    public void setDni(String dni) {
        this.dni = dni;
    }

    public Date getFechaNacimiento() {
        return fechaNacimiento;
    }

    public void setFechaNacimiento(Date fechaNacimiento) {
        this.fechaNacimiento = fechaNacimiento;
    }

    public String getNombre() {
        return nombre;
    }

    public void setNombre(String nombre) {
        this.nombre = nombre;
    }

    public int compareTo(Object o) {
        Persona persona = (Persona)o;

        if(this.apellidos.compareToIgnoreCase(persona.apellidos) == 0) {
            if(this.nombre.compareToIgnoreCase(persona.nombre) == 0) {
                return this.dni.compareTo(persona.dni);
            } else {
                return this.nombre.compareToIgnoreCase(persona.nombre);
            }
        } else {
            return this.apellidos.compareToIgnoreCase(persona.apellidos);
        }
    }
}

```

Voy a probar

...

```

public static void pintaLista(List lista) {
    for(int i=0;i<lista.size();i++) {
        System.out.println(lista.get(i));
    }
}

public static void main(String[] args) {

    List lista = new ArrayList();

    Calendar cal = Calendar.getInstance();
    cal.set(1976,3,21);
    Persona persona1 =
        new Persona("Francisco Javier","Martínez Páez","11111111A",cal.getTime());
    cal.set(1973,8,12);

    Persona persona2 =
        new Persona("Roberto","Canales Mora","22222222B",cal.getTime());
    cal.set(1975,7,23);

    Persona persona3 =
        new Persona("Alejandro","Pérez García","33333333C",cal.getTime());
    cal.set(1980,6,14);

    Persona persona4 =
        new Persona("Germán","Jiménez Centeno","44444444D",cal.getTime());
    cal.set(1981,5,8);
    Persona persona5 =
        new Persona("Alfonso","Blanco Criado","55555555E",cal.getTime());
    cal.set(1981,9,30);

    Persona persona6 =
        new Persona("Raúl","Expósito Díaz","66666666F",cal.getTime());

    lista.add(persona1);
    lista.add(persona2);
}

```

```

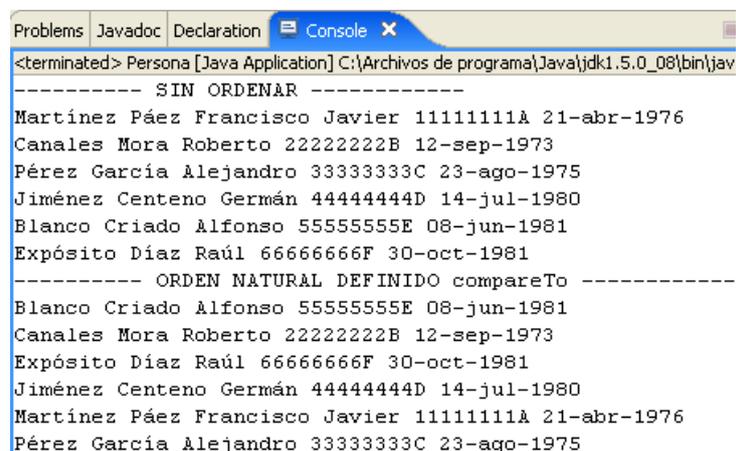
lista.add(persona3);
lista.add(persona4);
lista.add(persona5);
lista.add(persona6);

System.out.println("----- SIN ORDENAR -----");
pintaLista(lista);

System.out.println(
    "----- ORDEN NATURAL DEFINIDO en compareTo -----");
Collections.sort(lista);
pintaLista(lista);
}

```

Ejecutamos:



```

<terminated> Persona [Java Application] C:\Archivos de programa\Java\jdk1.5.0_06\bin\jav
----- SIN ORDENAR -----
Martínez Páez Francisco Javier 11111111A 21-abr-1976
Canales Mora Roberto 22222222B 12-sep-1973
Pérez García Alejandro 33333333C 23-ago-1975
Jiménez Centeno Germán 44444444D 14-jul-1980
Blanco Criado Alfonso 55555555E 08-jun-1981
Expósito Díaz Raúl 66666666F 30-oct-1981
----- ORDEN NATURAL DEFINIDO compareTo -----
Blanco Criado Alfonso 55555555E 08-jun-1981
Canales Mora Roberto 22222222B 12-sep-1973
Expósito Díaz Raúl 66666666F 30-oct-1981
Jiménez Centeno Germán 44444444D 14-jul-1980
Martínez Páez Francisco Javier 11111111A 21-abr-1976
Pérez García Alejandro 33333333C 23-ago-1975

```

Si ahora quisieramos por ejemplo, ordenar por fecha de Nacimiento:

```

public class NacimientoPersonaComparador implements Comparador {

    public int compare(Object o1, Object o2) {
        Persona persona1 = (Persona)o1;
        Persona persona2 = (Persona)o2;
        return persona1.getFechaNacimiento().
            compareTo(persona2.getFechaNacimiento());
    }
}

```

Lo probamos y lo ejecutamos:

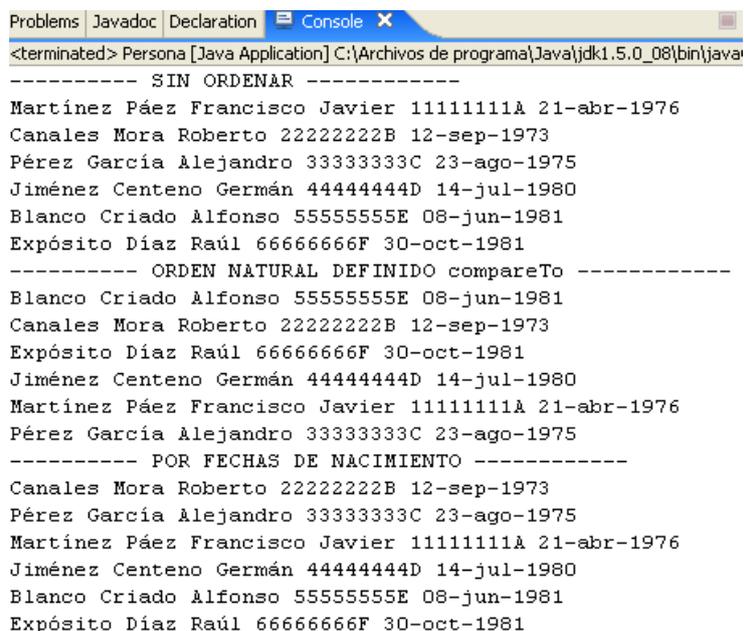
```

public static void main(String[] args) {

    ....

    System.out.println("----- POR FECHAS DE NACIMIENTO -----");
    Collections.sort(lista, new NacimientoPersonaComparador());
    pintaLista(lista);
}

```



```

<terminated> Persona [Java Application] C:\Archivos de programa\Java\jdk1.5.0_06\bin\java
----- SIN ORDENAR -----
Martínez Páez Francisco Javier 11111111A 21-abr-1976
Canales Mora Roberto 22222222B 12-sep-1973
Pérez García Alejandro 33333333C 23-ago-1975
Jiménez Centeno Germán 44444444D 14-jul-1980
Blanco Criado Alfonso 55555555E 08-jun-1981
Expósito Díaz Raúl 66666666F 30-oct-1981
----- ORDEN NATURAL DEFINIDO compareTo -----
Blanco Criado Alfonso 55555555E 08-jun-1981
Canales Mora Roberto 22222222B 12-sep-1973
Expósito Díaz Raúl 66666666F 30-oct-1981
Jiménez Centeno Germán 44444444D 14-jul-1980
Martínez Páez Francisco Javier 11111111A 21-abr-1976
Pérez García Alejandro 33333333C 23-ago-1975
----- POR FECHAS DE NACIMIENTO -----
Canales Mora Roberto 22222222B 12-sep-1973
Pérez García Alejandro 33333333C 23-ago-1975
Martínez Páez Francisco Javier 11111111A 21-abr-1976
Jiménez Centeno Germán 44444444D 14-jul-1980
Blanco Criado Alfonso 55555555E 08-jun-1981
Expósito Díaz Raúl 66666666F 30-oct-1981

```

Ha sido sencillo.



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works 2.5 License](https://creativecommons.org/licenses/by-nc-nd/2.5/).  
[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](mailto:info@autentia.com)

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

**Autentia = Soporte a Desarrollo & Formación**

Creatividad Internet

[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	
e-mail	<input type="text"/>
	<input type="button" value="Enviar"/>

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

### Nombre Corto

[Instrumentación de componentes Java usando JMX](#)

[Mensajes multi-idioma en Java](#)

[Generación .exe con ficheros Java](#)

[Técnicas básicas y poco comentadas en Java](#)

[Decompilar Java](#)

[Optimización de Serialización Java](#)

[Analizar ejecución de programa Java](#)

[Documentar código Java con JavaDoc](#)

[Construir un Servidor Web en Java](#)

[Novedades en Java 1.5](#)

### Descripción

Os mostramos como utilizar las características avanzadas de los servidores de aplicaciones Java/J2EE para realizar la instrumentación de componentes Java usando JMX y poniendo como ejemplo la gestión dinámica de propiedades entre nodos

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

Os mostramos como, utilizando la herramienta gratuita JSmooth, podemos generar un fichero .exe a partir de nuestros programas Java

Os mostramos como realizar algunas cosas simples en Java: Formateo de decimales y enteros, gestión de preferencias y comparación entre objetos de nuevas clases

Os mostramos como recuperar el fuente de vuestro código a partir de los ficheros compilados .class

Os mostramos una sencilla técnica para mejorar el rendimiento de la serialización de objetos en Java, a través de Streams asociados a buffers en memoria.

Os mostramos como investigar el comportamiento de vuestros programas Java, en ejecución, a través del profiling.

Os mostramos como utilizar los comentarios y etiquetas de JavaDoc para documentar programas Java.

En este tutorial os enseñamos los principios de las aplicaciones multi-hilo a través de la creación de un servidor web básico en Java. Podremos ver en un ejemplo real el uso de sockets, threads, excepciones, etc.

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.

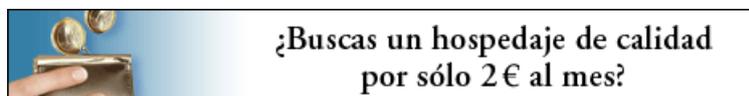
Nota: Los tutoriales mostrados en este Web tienen como objetivo la difusión del conocimiento.

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](mailto:rcanales@adictosaltrabajo.com) para su resolución.

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



[www.AdictosAlTrabajo.com](http://www.AdictosAlTrabajo.com) Optimizado 800X600