

# ¿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 

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



CoNcept

Lanzado

**TNTConcept versión 0.6** ( 12/07/2007)

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="#">Iván Zaera Avellón</a></b></p> <p><b>Puedes encontrarme en <a href="#">Autentia</a></b>  <b>Somos expertos en Java/J2EE</b>  <b>Contacta en:</b>  <a href="mailto:izaera@autentia.com">izaera@autentia.com</a></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>real business solutions</b></p> <p><a href="#">Catálogo de cursos</a></p>
---	--

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

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

#### [Bugzilla Desktop Client](#)

Discover more features for Bugzilla in the new desktop client!  
[almworks.com/deskzilla](http://almworks.com/deskzilla)

#### [Bug Tracking Tool](#)

Web based bug tracking software  
 Notify, Report, Customize  
[www.FitTrackingSolutions.com](http://www.FitTrackingSolutions.com)

#### [Woodpecker Issue Tracker](#)

flexible, webbased bug, issue and change request tracking  
[www.woodpecker-it.com](http://www.woodpecker-it.com)

#### [Eclipse Plugin](#)

Apollo for Eclipse provides Java 5 &

**Fecha de creación del tutorial: 2007-07-19**

## Integración unidireccional entre Sourceforge y Bugzilla

### Introducción

Este tutorial describe una posible integración entre Sourceforge y Bugzilla. Como sabéis, en Autentia, estamos desarrollando la aplicación de gestión TNTConcept, disponible en Sourceforge. Además, desarrollamos software para nuestros clientes y, como buena empresa de desarrollo informático, en nuestros sistemas internos tenemos instalando un sistema de trazabilidad de cambios: Bugzilla.

En Bugzilla llevamos la gestión de cambios al software de nuestros clientes, pero ahora nos encontramos con el problema de que los errores de TNTConcept los tenemos que llevar por Sourceforge y eso nos causa dificultades, al no estar todo centralizado. Hemos decidido que lo mejor es centralizar todo en Bugzilla, pero no queremos renunciar a Sourceforge como primer nivel de soporte de los usuarios de TNTConcept. Para ello hemos realizado una integración entre Sourceforge y Bugzilla de forma que todo lo que se introduzca en Sourceforge aparezca "mágicamente" en nuestro Bugzilla.

Con esta integración, Sourceforge pasa a ser un canal más de comunicación con nuestros "clientes" (como el correo o el teléfono) y el gestor principal de cambios se unifica en nuestra instalación interna de Bugzilla, permitiéndonos más control sobre nuestros

flujos de trabajo.

Pero bueno, ya está bien de palabrería y vayamos al grano...

## El grano



## Configurar Sourceforge

Lo primero que tenemos que hacer es configurar el proyecto en Sourceforge para que mande correos a una dirección nuestra cada vez que se efectúe una modificación en un tracker (Sourceforge llama tracker a cualquier elemento de información, como "Bug" o "Feature request", asociado al proyecto). Para ello iremos a la ventana de configuración del tracker que queramos integrar con Bugzilla:

SF.net » Projects » TNTConcept » Tracker » Bugs

## TNTConcept

Project ▾ Tracker ▾ Mailing Lists ▾ Forums ▾ Code ▾ Services ▾ Download ▾ Documentation ▾ Tasks ▾ Admin ▾

Submit New Browse Reporting Admin

**Admin functions:** [Add/Browse Artifact Types](#) | [Edit/Update Options in: Bugs](#)

**Add/Update Categories**  
Add categories like, 'mail module', 'gant chart module', 'cvs', etc

**Add/Update Groups**  
Add groups like, 'v1.2', 'unsupported', 'unverified', etc

**Add/Update Canned Responses**  
Create/Change generic response messages for the tracker.

**Add/Update Users & Permissions**  
Add/remove users to/from this tracker.

**Mass Re-assign Tracker Items**  
Re-assign Tracker items from a past/present technician to a present technician.

**Update preferences**  
Set up prefs like expiration times, email addresses, etc.

Ahora escribiremos la dirección elegida y también marcaremos el cuadro para que envíe todos los cambios (si no nos envía sólo correos cuando se da de alta algo):

**Description:**  
Bug Tracking System

**Visibility of this Tracker:** Visible to all site users ▾

**Allow non-logged-in postings**

Preventing anonymous posting will help to prevent abuse (such as spam) of your Tracker, increase the them with your response), as well as helping to strengthen the SourceForge.net user community. We will be able to post once they have logged in to the SourceForge.net site.

**Display the "Resolution" box**

**Send email on new submission to address:**  
direccion@dominio.cq

**Send email on all changes**

**Automatically re-open closed/deleted items if a new comment is posted by the submitter**

A partir de este momento, Sourceforge enviará correos a la dirección especificada cada vez que alguien introduzca un cambio o cree un tracker.

## Configurar Bugzilla

Lo siguiente que haremos es instalar Bugzilla en un equipo. En nuestro caso hemos instalado Bugzilla 3.0 en una máquina Debian siguiendo el tutorial anterior a este (Como instalar Bugzilla 3.0 en Debian). Después de la instalación debemos configurar Bugzilla para que se puedan dar de alta y modificar bugs por correo. Para ello usaremos el script "email\_in.pl" que viene con Bugzilla.

Primero creamos un usuario en el sistema llamado "bugzilla" con el comando típico de Debian:

```
$ adduser bugzilla
```

Entramos al sistema con dicho usuario y en el directorio HOME creamos un fichero llamado ".forward" para que el MTA (Mail Transfer Agent, o sea, el demonio de correo) invoque nuestro script cada vez que llega un correo a este usuario. El fichero ".forward" debe contener lo siguiente:

```
\bugzilla,"|/home/bugzilla/email_in.sh"
```

Es importante que nuestro MTA sea "exim4" (uno de los varios disponibles en Debian) para que este fichero ".forward" funcione bien.

El fichero "email\_in.sh" contendraoacute;:

```
#!/bin/sh cd /opt/bugzilla ./email_in.pl -vvv
```

Tenemos que asegurarnos de que el script "email\_in.sh" tiene permisos de ejecución porque si no rebotarán todos los correos enviados al usuario "bugzilla". También es importante cambiar "/opt/bugzilla" por el directorio donde hayamos instalado Bugzilla.

Con esta configuración, cada vez que el usuario "bugzilla" reciba un correo se evaluará con el script "email\_in.pl" y se dará de alta o modificará un bug en Bugzilla. Para más información sobre la estructura del correo visitar la página web de Bugzilla que lo explica: [http://www.bugzilla.org/docs/3.0/html/api/email\\_in.html](http://www.bugzilla.org/docs/3.0/html/api/email_in.html). Conviene, a pesar de haber configurado el script, que alguien se haga responsable de la cuenta de correo "bugzilla" para que pueda ver correos que no sean de bugs o posibles "rebotes" y que, de paso, vacie la cuenta de vez en cuando.

## Configurar usuario Sourceforge

Ahora crearemos un usuario llamado "sforge" que recibirá todos los correos de Sourceforge, los transformará mediante un script, y se los reenviará al usuario "bugzilla" para que los de de alta en Bugzilla. Ejecutamos el comando típico de creación de usuarios:

```
$ adduser sforge
```

Entramos al sistema con el usuario nuevo y, en su directorio HOME, creamos un fichero ".forward" que contenga:

```
\sforge, "|/bin/sh /home/sforge/sf2bz.sh"
```

Esto hará que se ejecute el script "sf2bz.sh" cada vez que se recibe un correo. Dicho script debe contener:

```
#!/bin/sh

log()
{
    LOGFILE=$1
    MSG=$2

    echo `date` - $MSG >> $LOGFILE
}

extract_id()
{
    HEADER=$1
    FILE=$2

    LINE=`grep "$HEADER" $FILE`
    I=`expr index "$LINE" '\('`
    I=`expr $I + 1`
    J=`expr index "$LINE" '\)`
    J=`expr $J - $I`
    expr substr "$LINE" $I $J
}

extract_header()
{
    HEADER=$1
    FILE=$2

    RET=`grep "$HEADER" $FILE | awk 'BEGIN {FS=":"} {print $2}`
    echo $RET
}

extract_first()
{
    LINE=$1
    FILE=$2

    grep -m 1 "$LINE" $FILE
}
```

```

extract_section()
{
    FILE=$1
    SECTION=$2
    OFFSET=$3

    I=`grep -n "$SECTION" $FILE | awk 'BEGIN {FS=":"} {print $1}`'
    I=`expr $I + $OFFSET`
    N=`wc -l < $FILE`
    I=`expr $N - $I`
    J=`tail -n $I $FILE | grep -n -- "-----" | awk 'BEGIN {FS=":"} {print $1}`'
    J=`expr $J - 1`
    tail $FILE -n $I | head -n $J
}

generate_new()
{
    FILE=$1
    SEVERITY=$2
    SUMMARY=$3
    TICKET_ID=$4
    TICKET_URL=$5
    SUBMITTER=$6
    TICKET_DESC=$7

    cat << EOF > $FILE
@product=TestProduct
@version=unspecified
@op_sys=Linux
@component=TestComponent
@rep_platform=PC
@short_desc=$SUMMARY
@assigned_to=admin
@bug_severity=$SEVERITY
@cf_sforge_id=$TICKET_ID
@bug_file_loc = $TICKET_URL

==== SOURCEFORGE.NET INFO ====
$TICKET_URL
Ticket ID: $TICKET_ID
Submitter: $SUBMITTER
==== SOURCEFORGE.NET INFO ====

$TICKET_DESC
EOF
}

generate_comment()
{
    FILE=$1
    BUG_ID=$2
    SUBMITTER=$3
    COMMENT=$4

    cat << EOF >$FILE
@bug_id=$BUG_ID

==== Submitted by: $SUBMITTER ====

$COMMENT
EOF
}

# Define variables
WORKDIR=/home/sforge/sf2bz
MSGFILE=$WORKDIR/sfmsg
LOGFILE=$WORKDIR/log

# Create workdir if it does not exist
log $LOGFILE "=====
if [ ! -d $WORKDIR ]
then
    mkdir $WORKDIR

```

```
fi

# Dump received mail to temporary file
log $LOGFILE "Processing new message"
cat /dev/null > $MSGFILE
while read LINE
do
    echo $LINE >> $MSGFILE
done

# See if message should be processed
grep "From.*noreply@sourceforge.net" $MSGFILE
if [ $? != 0 ]
then
    log $LOGFILE "Message does not come from noreply@sourceforge.net: ignored"
    exit 0
fi

# Get field values
SF_TICKET_URL=`extract_first "https://sourceforge.net/tracker" $MSGFILE`
log $LOGFILE "SF_TICKET_URL=$SF_TICKET_URL"

SF_TICKET_REASON=`extract_header "X-SourceForge-Tracker-itemupdate-reason" $MSGFILE`
log $LOGFILE "SF_TICKET_REASON=$SF_TICKET_REASON"

SF_TICKET_CATEGORY=`extract_header "X-SourceForge-Tracker-trackerid" $MSGFILE`
log $LOGFILE "SF_TICKET_CATEGORY=$SF_TICKET_CATEGORY"

case "$SF_TICKET_CATEGORY" in
    933043)
        SF_TICKET_SEVERITY=enhancement
        ;;
    933040)
        SF_TICKET_SEVERITY=normal
        ;;
    *)
        log $LOGFILE "Unknown ticket category: will file as enhancement"
        SF_TICKET_SEVERITY=enhancement
        ;;
esac
log $LOGFILE "SF_TICKET_SEVERITY=$SF_TICKET_SEVERITY"

SF_TICKET_ID=`extract_header "X-SourceForge-Tracker-itemid" $MSGFILE`
log $LOGFILE "SF_TICKET_ID=$SF_TICKET_ID"

SF_TICKET_SUMMARY=`extract_header "Summary" $MSGFILE`
log $LOGFILE "SF_TICKET_SUMMARY=$SF_TICKET_SUMMARY"

SF_TICKET_SUBMIT=`extract_header "Submitted By" $MSGFILE`
log $LOGFILE "SF_TICKET_SUBMIT=$SF_TICKET_SUBMIT"

SF_TICKET_ASSIGN=`extract_header "Assigned to" $MSGFILE`
log $LOGFILE "SF_TICKET_ASSIGN=$SF_TICKET_ASSIGN"

SF_TICKET_DESC=`extract_section $MSGFILE "Initial Comment" 0`
log $LOGFILE "SF_TICKET_DESC=$SF_TICKET_DESC"

# See if ticket has new comment attached
if [ "$SF_TICKET_REASON" == "Comment added" ]
then
    # Get ticket new comment
    SF_TICKET_COMMENT=`extract_section $MSGFILE ">Comment By" 7`
    log $LOGFILE "SF_TICKET_COMMENT=$SF_TICKET_COMMENT"

    # Get ticket new comment submitter
    SF_TICKET_COMMENT_SUBMIT=`extract_header ">Comment By" $MSGFILE`
fi

# Check what to do with ticket
SEND_MAIL=0
case "$SF_TICKET_REASON" in

    "Tracker Item Submitted")
```

```

generate_new $WORKDIR/bzmsg "$SF_TICKET_SEVERITY" "$SF_TICKET_SUMMARY"
"$SF_TICKET_ID" "$SF_TICKET_URL" "$SF_TICKET_SUBMIT" "$SF_TICKET_DESC"
log $LOGFILE "Generated new ticket mail for Bugzilla"
SEND_MAIL=1
;;

"Comment added")
wget -o /dev/null -O $WORKDIR/search "http://server/bugzilla/buglist.cgi?
query_format=advanced&
short_desc_type=allwordssubstr&
shor&long_desc_type=substring&long_desc=&
bug_file_loc_type=allwordssubstr&
bug_file_loc=&deadlinefrom=&deadlineto=&
bug_status=&emailassigned_to1=1&
emailtype1=substring&email1=&emailassigned_to2=1&
emailreporter2=1&emailcc2=1&
emailtype2=substring&email2=&bugidtype=include&
bug_id=&votes=&chfieldfrom=&
chfieldto=Now&chfieldvalue=&cmdtype=doit&
order=Reuse+same+sort+as+last+time&
field0-0-0=cf_sforge_id&
type0-0-0>equals&value0-0-0=$SF_TICKET_ID"

BZ_ID=`grep "show_bug.cgi.id=" $WORKDIR/search`
log $LOGFILE "grep returned - $BZ_ID-"
BZ_ID=`echo $BZ_ID | cut -d = -f 3`
BZ_ID=`echo $BZ_ID | cut -d \ " -f 1`
log $LOGFILE "BZ_ID=$BZ_ID"

generate_comment $WORKDIR/bzmsg $BZ_ID "$SF_TICKET_COMMENT_SUBMIT"
"$SF_TICKET_COMMENT"
log $LOGFILE "Generated comment mail for Bugzilla"
SEND_MAIL=1
;;

*)
log $LOGFILE "Unknown ticket reason: $SF_TICKET_REASON"
;;

esac

# Send mail to Bugzilla
if [ $SEND_MAIL -eq 1 ]
then
mail -i bugzilla -s "Automatic bug submission from Sourceforge" < $WORKDIR/bzmsg
log $LOGFILE "Sent mail to Bugzilla"
fi

```

Por supuesto, es importante cambiar los textos en color verde por los valores adaptados a nuestro sistema. El script básicamente lo que hace es buscar en el mensaje de correo cabeceras que envía Sourceforge y también obtiene ciertos contenidos del cuerpo del mensaje. Después, con los valores extraídos compone un mensaje en el formato de Bugzilla y se lo envía.

Ni que decir tiene que este script funcionará mientras Sourceforge no cambie el formato de sus correos. Si esto sucediese, habría que volver a hacer la ingeniería inversa y retocar el script. Para efectuar la ingeniería inversa basta con ver el código fuente de algún correo que Sourceforge nos haya enviado.

También hay que prestar especial interés a la línea en la que se invoca a "wget". Esta línea consulta a Bugzilla cuando Sourceforge nos envía un correo sobre un bug ya existente. Esta consulta es necesaria para mapear el id del bug de Sourceforge al de Bugzilla. Lo que hace el script es buscar todos los bugs de Bugzilla cuyo campo "cf\_sforge\_id" contiene el id del ticket en Sourceforge. El campo "cf\_sforge\_id" es un campo de usuario de Bugzilla que nosotros hemos creado ad hoc para almacenar el id del ticket en Sourceforge. Se rellena cuando el script da de alta un bug por vez primera. El retorno de la consulta es una página HTML con el listado de bugs encontrados (debería haber sólo uno si todo está bien). Después se busca en este HTML, con "grep", el id del ticket devuelto y se extrae con "cut". Obviamente este código sólo funcionará mientras la estructura del HTML de Bugzilla no cambie. Si cambia hay que volver a hacer la ingeniería inversa lanzando una búsqueda desde Bugzilla y viendo el código fuente de la página devuelta. Se supone que tiene que haber formas más civilizadas de hacer esto llamando a los scripts Perl de Bugzilla pero aún no las hemos investigado.

Para crear el campo de usuario "cf\_sforge\_id" debemos acceder en Bugzilla a la página de administración "Custom fields" y seleccionar "Add a new custom field":

## Configurar fetchmail

El siguiente paso es configurar "fetchmail". Este paso sólo será necesario cuando los mensajes enviados a la dirección de correo que hemos dado de alta en Sourceforge no se reciban directamente en la máquina donde hemos creado el usuario "sforge". Por ejemplo, imaginemos que hemos creado una dirección en GMail llamada "errores@gmail.com" donde estamos enviando los correos de Sourceforge. Es evidente que los correos que lleguen a esa dirección se quedan en la bandeja de entrada de GMail y no se reenvían a nuestro equipo. Por eso necesitamos "fetchmail" para que se baje los correos por POP del buzón de GMail y haga como que llegan al usuario Bugzilla.

Primero instalamos fetchmail con el usuario root:

```
# apt-get install fetchmail
```

A continuación lo configuramos para que arranque como servicio. Para ello editamos el fichero "/etc/default/fetchmail" y cambiamos el valor de la variable "START\_DAEMON" de "no" a "yes". El servicio "fetchmail" arrancará automáticamente con el próximo reinicio.

Creamos ahora un fichero llamado "/etc/fetchmailrc" con root como dueño y permisos "600". En este fichero escribiremos los parámetros de configuración de la cuenta de correo cuyos mensajes queremos descargar con "fetchmail". Por ejemplo:

```
set postmaster "root"
set nobouncemail
set no spambounce
set properties ""
set daemon 1
poll pop.gmail.com with proto POP3
  user 'errores@gmail.com' there with password 'ERRORES123'
  is 'bugzilla' here options no rewrite mimedecode ssl
```

Hecho esto lanzamos con root el servicio "fetchmail" con el comando (también lo podemos hacer reiniciando la máquina):

```
# /etc/init.d/fetchmail start
```

A partir de este momento "fetchmail" debería bajarse los correos de "errores@gmail.com" y reenviárselos al usuario "sforge".

## Conclusión

Ya tenemos montado el chiringo. Vamos a recapitular ahora que pasa cuando alguien da de alta un bug en Sourceforge:

1. Un usuario da de alta un bug en Sourceforge.
2. Sourceforge envía un correo de notificación a "errores@gmail.com".
3. Fetchmail recupera el correo de "errores@gmail.com" usando POP y se lo envía al usuario local "sforge".
4. Se lee el fichero ".forward" en el directorio HOME de "sforge" y se invoca el script "sf2bz.sh" pasándole el correo descargado de "errores@gmail.com".
5. El script "sf2bz.sh" analiza el correo y genera otro nuevo con el formato de Bugzilla.
6. El script "sf2bz.sh" envía el correo generado al usuario local "bugzilla".
7. Se lee el fichero ".forward" en el directorio HOME de "bugzilla" y se invoca el script "email\_in.pl" pasándole el correo.
8. El script "email\_in.pl" da de alta el bug creado en Sourceforge en Bugzilla.

Y ahora que pasa cuando alguien modifica un bug en Sourceforge y que, por tanto, ya se ha creado en Bugzilla:

1. Un usuario modifica un bug en Sourceforge.
2. Sourceforge envía un correo de notificación a "errores@gmail.com".
3. Fetchmail recupera el correo de "errores@gmail.com" usando POP y se lo envía al usuario local "sforge".
4. Se lee el fichero ".forward" en el directorio HOME de "sforge" y se invoca el script "sf2bz.sh" pasándole el correo descargado de "errores@gmail.com".
5. El script "sf2bz.sh" analiza el correo.

6. El script "sf2bz.sh" hace una búsqueda en Bugzilla para encontrar el bug de Bugzilla que se corresponde con el de Sourceforge.
7. El script "sf2bz.sh" genera un nuevo correo con el formato de Bugzilla que incluye el id del bug buscado.
8. El script "sf2bz.sh" envía el correo generado al usuario local "bugzilla".
9. Se lee el fichero ".forward" en el directorio HOME de "bugzilla" y se invoca el script "email\_in.pl" pasandole el correo.
10. El script "email\_in.pl" modifica el bug correspondiente al de Sourceforge en Bugzilla.

## Sobre el autor

Iván Zaera Avellón, Ingeniero en Informática por la Universidad Autónoma de Madrid - [izaera@autentia.com](mailto:izaera@autentia.com)  
 Consultor tecnológico de Autentia (Formación, Consultoría, Desarrollo de sistemas transaccionales)

Autentia Real Business Solutions S.L. - Soporte a Desarrollo - <http://www.autentia.com>



This work is licensed under a [Creative Commons Attribution-NonCommercial-No Derivative Works 2.5 License](http://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**

[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**[Detección de errores Java con FindBugs](#)[Publicación de una nueva release en SourceForge.Net](#)[Problemas al planificar un proyecto](#)[Herramientas gratuitas de gestión de proyectos](#)[MS Access y la de gestión de Proyectos](#)[Gestión de Errores con Bugzilla](#)[Gestión de proyectos con project](#)[Instalación de Bugzilla 3.0 en Debian](#)**Descripción**

Os mostramos como instalar y utilizar FindBugs, una excelente herramienta para analizar, de un modo estático, posibles problemas en vuestro código Java

En este tutorial vamos a compartir como se hace una nueva release (entrega) dentro del sistema de gestión de proyectos de software libre [www.sourceforge.net](http://www.sourceforge.net), aprovechando el lanzamiento de la segunda Beta de TNTConcept

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

Os mostramos dos opciones de código abierto, construidas en Java, para ayudaros en la organización de vuestro trabajo diario (Memoranda) y proyectos (GanttProject).

Os mostramos como herramientas sencillas y potentes, como Microsoft Access, pueden ayudarnos a capturar y organizar información vital para los responsables de proyectos.

Alejandro Pérez nos enseña como instalar (en Debian) y utilizar Bugzilla, una herramienta gratuita de gestión de errores.

En este tutorial os enseñaremos crear un plan, realizar el seguimiento del proyecto, como cerrar el proyecto y comunicar los resultados

Instalación de Bugzilla 3.0 en Debian

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](#)

