Memoria técnica

# Objetivo

El presente documento tiene como objetivo enlistar las actividades que se llevaron a cabo para instalar el portal de Tickets en **Gentera**.

*Nota. El presente documento está dirigido a personal técnico como son Administradores de Sistemas y Bases de Datos con conocimientos en Linux, Java, Apache Tomcat, MySQL, Oracle DB, protocolos de Internet y LDAP.*

# Especificaciones del ambiente de desarrollo

### Equipo

|  |  |
| --- | --- |
| **Componente** | **Versión** |
| Nombre del servidor | BCBETA704 |
| IP |  |
| Puertos TCP Requeridos | 8080 |
| Memoria RAM | 2.0 GB requeridos  3,2 GB recomendados |
| Disco duro | 512 MB |
| Procesador | 2 Núcleos de 64 Bits |

### Software requerido

|  |  |
| --- | --- |
| **Componente** | **Versión** |
| OS Linux | Compatible con kernel 3 o superior |
| Oracle DB | 10g o superior |
| Oracle Java™ | SDK 8 |
| Apache Tomcat | 8.5 |
| XWiki | 8.4.5 |

### Usuarios

|  |  |  |
| --- | --- | --- |
| **Componente** | **Usuario** | **Contraseña** |
| Linux | tomcat | Ninguna, accedido por root a través del comando **su -** |
| Oracle DB | Ptickets | PTICKETS17# |
| Administrador Tickets | Admin | Ticket101 |

### Directorios

|  |  |
| --- | --- |
| **Componente** | **Carpeta** |
| Java™ SDK | /opt/jdk8 |
| Apache Tomcat | /opt/tomcat |
| Aplicaciones | /opt/tomcat/webapps |
| Logs | /opt/tomcat/logs |

# Instalación

### Preparación del ambiente Linux

Vía Secure Copy (SCP) copie el archivo **java.sh** al directorio /tmp de el servidor Linux

Inicie sesión como root y ejecute la siguiente serie de comandos

# cp /tmp/java.sh /etc/profile.d/

# chmod 755 /opt/profile.d/java.sh

# chown root:root /opt/profile.d/java.sh

# groupadd -g 200 javadev

# useradd -s /bin/bash -m -d /home/tomcat -g javadev -G users -u 201 tomcat

Reinicie el servidor al haber terminado este paso.

### Instalación del Oracle Java SDK 8

Con el usuario root, copie el archivo correspondiente al Java™ SDK 8, en este caso jdk-8u135-linux-x64.tar.gz, al directorio /tmp del servidor por SCP.

Si es necesario abra una sesión de SSH como root y ejecute la siguiente serie de comandos:

# cp /tmp/jdk-8u135-linux-x64.tar.gz /opt/

# cd /opt

# tar zxvf jdk-8u135-linux-x64.tar.gz

# chown -R tomcat:javadev /opt/jdk1.8.0\_135

# chmod –R a+r /opt/jdk1.8.0\_135

# ln -s /opt/jdk1.8.0\_135 /opt/jdk

# ln -s /opt/jdk1.8.0\_135 /opt/jdk8

*Nota. El número 135 referido puede variar si usted instala un JDK más reciente, por ejemplo, si usa la versión 151 del Java™ SDK 8 entonces sustituya todos las instancias de 135 por 151.*

### Instalación de Apache Tomcat 8

Con el usuario root, copie el archivo correspondiente a Apache Tomcat 8, en este caso apache-tomcat-8.5.14.tar.gz, al directorio /tmp del servidor por SCP.

Si es necesario abra una sesión de SSH como root y ejecute la siguiente serie de comandos:

# cp /tmp/apache-tomcat-8.5.14.tar.gz /opt/

# cd /opt

# tar zxvf apache-tomcat-8.5.14.tar.gz

# chown -R tomcat:javadev /opt/apache-tomcat-8.5.14

# chmod –R a+r /opt/apache-tomcat-8.5.14

# ln -s /opt/apache-tomcat-8.5.14 /opt/tomcat

# ln -s /opt/apache-tomcat-8.5.14 /opt/tomcat8

*Nota. El número 8.5.14 referido puede variar si usted instala un Tomcat más reciente, por ejemplo, si usa la versión 8.6.10, entonces sustituya todos las instancias de 8.5.14 por 8.6.10.*

### Instalación de la base de datos

En este paso se requiere que la persona encargada de la instalación de bases de datos lleve a cabo el proceso. En ambientes de desarrollo se recomienda que sea en el mismo servidor (localhost) mientras que en QA y Producción que sea en servidores separados.

En caso de usar servidores separados vea los Anexos para saber como modificar la configuración de fábrica de la base de datos y volver a preparar los archivos de instalación.

Para crear las tablas requeridas para la operación de la aplicación Ejecute el archivo **tickets\_fase\_II.oracle.sql** encontrado en su distribución.

### Instalación de las aplicaciones

El portal de Tickets se encuentra compuesto por las siguientes 4 aplicaciones instalables:

|  |  |
| --- | --- |
| **Componente** | **Archivo** |
| Portal de tickets, versión API de Remedy™ | ticketsApi.war |
| Portal de tickets, versión que usa WebServices | ticketsApi.war |
| Administrador del portal | ticketsAdmin.war |
| Wiki para servicios de referencia a los usuarios | xwiki.war |

Para instalar uno o todos los componentes solo transfiera el o los componentes deseados vía SCP a el servidor en el directorio /opt/tomcat8/webapps. Se recomienda que primero los coloque en /tmp les cambie el dueño con el siguiente comando:

# chown tomcat:javadev \*.war

***Importante****. XWiki es un producto de software libre y su instalación y configuración sale del alcance de este manual. Para saber como manejarlo, por favor refiérase a la siguiente URL:*

**http://www.xwiki.org/xwiki/bin/view/Documentation/UserGuide/GettingStarted/**

### Prueba de la instalación

Para ello deberá usar la siguiente lista de URLs

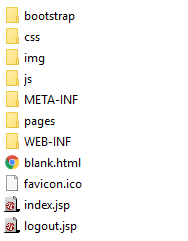
|  |  |
| --- | --- |
| **Componente** | **Archivo** |
| Portal API | http://<Servidor o IP>/ticketsApi |
| Portal WebServices | http://<Servidor o IP>/tickets |
| Administrador del portal | http://<Servidor o IP>/ticketsAdmin |
| XWiki | http://<Servidor o IP>/tickets |

***Importante****. En caso de haber instalado un certificado de seguridad, utilice https en lugar de http.*

## Anexos

### Anexo A. Estructura de la aplicación

**TicketsAPI y Tickets**

Ambas aplicaciones tienen la misma estructura solo difieren en las dependencias internas. Mientras la primera consume lor archivos jar de la API de Remedy™ la segunda consume una librería llamada **helpdeskClient** para hacer la comunicación con Remedy™. En ambas se encuentran los siguientes directorios:

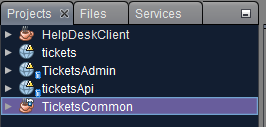
|  |  |
| --- | --- |
| **Carpeta o archivo** | **Propósito** |
| bootstrap | Carpeta de archivos de bootstrap tomados de la imagen gráfica del portal principal de Gentera |
| css | Carpeta con la hoja de estilo de la aplicación |
| img | Carpeta de imágenes usadas en la aplicación |
| js | Carpeta con código Javascript de uso interno. JQuery, JQuery-UI y archivos requeridos por estas librerías |
| META-INF | Carpeta que contiene context.xml requerido por Tomcat para depoyar y configurar la base de datos. |
| pages | Páginas y templates de Apache Struts |
| WEB-INF | Carpeta con la configuración principal de la aplicación (web.xml), librerías, archivos de configuración y clases Java™ |
| blank.html | Archivo HTML en blanco requerido para llenar ‘espacios’ en los templates de Struts. |
| favicon.ico | Icono con el lógo de Gentera a a mostrarse en los navegadores de Internet. |
| index.jsp | Archivo temporal de inicio, todas las aplicaciones se redireccionan a main.jsp en el directorio **/pages** |
| logout.jsp | Página JSP que termina la sesión del usuario. |

**TicketsAdmin**

En el caso de esta aplicación la estructura es la misma que la de las dos anteriores pero los nombres de las clases y muchas de sus páginas son diferentes debido a que tiene funcionalidades diferentes.

Cabe recalcar que para poder hacer el manejo de los incidentes críticos (o incidentes padre) se recurre solamente a la API de Remedy™ por lo que en este caso no tenemos 2 versiones de la misma aplicación.

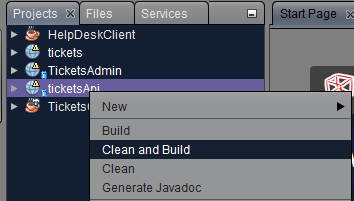
### Anexo B. Recompilación



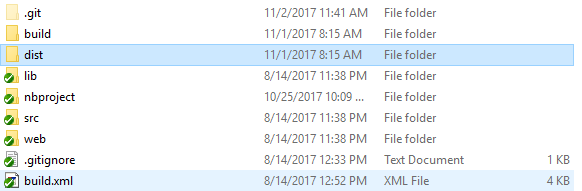
El proyecto del portal de Tickets se compone de varios proyectos de Netbeans versión 8, por lo que para poder recompilar la aplicación fácilmente, además del software listado previamente, se requerirá que descargue e instale Netbeans. Una vez hecho esto, deberá abrir los siguientes archivos del repositorio Git. En el órden que se presenta en la tabla siguiente:

|  |  |
| --- | --- |
| **Proyecto** | **Descripción** |
| TicketsCommons | Clases de Java y archivos comunes a todos los proyectos. |
| HelpDeskClient | Clientes de los Webservices proporcionados por **Gentera** para crear tickets y hacer búsquedas. |
| TicketsAdmin | Administrador del portal de Tickets. **Requiere** de TicketsCommons. |
| TicktetsApi | Versión del portal de Tickets que usa las librerías del API de Remedy™. **Requiere** de TicketsCommons. |
| Tickets | Versión del portal de Tickets que se comunica con Remedy™ a través de WebServices. **Requiere** de TicketsCommons y HelpDeskClient. |

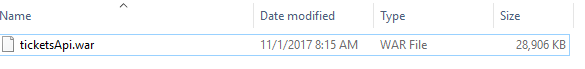
Para recompilar alguno de los proyectos lo que deberá hacer es hacer clic derecho en el proyecto que haya modificado y seleccionar la opción **Clean and Build**



Nótese que para cada proyecto primero deberá ejecutar la opción Clean and Build en cada una de sus dependencias para asegurar que se tienen todos los cambios integrados en la versión a generar. Esto provocará que Netbeans cree un nuevo archivo war para distribución en una carpeta llamada **dist** dentro del proyecto que ha recompilado. Siguiendo nuestro ejemplo para **TicketsApi**, esta quedaría así:



Entrando a esa carpeta **dist** podrá ver el archivo correspondiente al proyecto que acaba de recompilar, para efectos de este ejemplo: **ticketsApi**.



***Importante****. La serie de tecnologías que componen a Java, herramientas y servidores fueron desarrollados pensando en ambientes Unix™ por lo que si modifica las mayúsculas y minúsculas de alguno de los archivos de cualquiera de los proyectos dejará a este proyecto inusabe. Ejemplo: si cambia index.jsp por Index.jsp la aplicación dejará de funcionar. Mismo caso si cambia de ticketsApi.jar a TicketsApi.jar.*

Opcionalmente, si tiene los conocimientos adecuados de Java™, puede instalar Apache Ant 1.8 o superior y recompilar con esta herramienta los proyectos ya que Netbeans genera archivos **build.xml** compatibles con el software mencionado. Los detalles exactos escapan del alcance de esta memoria técnica.

### Configuraciones

La aplicación cuenta con varios archivos de configuraciones y una tabla en la base de datos específica para datos comunes, a continuación se listan estos en relación con el código fuente.

**src/resources/labelsResources.properties**

Archivo con los textos usados en toda la aplicación: En JSPs, mensajes de error, mensajes a la bitácora, etc.

**src/resources/log4j.xml**

Archivo de configuración de los logs de las aplicaciones. Controla el destino de las entradas de bitácora y la frecuencia con que esta es rotada.

**src/resources/struts.xml**

Archivo de control de flujo de las páginas JSP. NO LO MODIFIQUE a menos que conozca bien Struts 2.

**web/WEB-INF/tiles.xml**

Archivo de configuración de las plantillas de Struts 2. NO LO MODIFIQUE a menos que conozca bien Struts 2.

**web/WEB-INF/web.xml**

Archivo estándar de configuración de aplicaciones web Java™ usado por lo servidores de aplicaciones.

**web/META-INF/ context.xml**

Archivo de contexto web de Apache Tomcat. En el se encuentran los siguientes valores escenciales:

1. Ruta relativa al servidor de la aplicación, controla el path del servidor web donde se verá la aplicación:

<Context antiJARLocking="true" autodeploy="true" path="/ticketsApi" reloadable="true">

1. Pool de conexiones de la base de datos. Si requiere que la base de datos resida en otro servidor o funcione con otro RDMS modifique la siguiente entrada:

<Resource auth="Container"

type="javax.sql.DataSource"

driverClassName="oracle.jdbc.driver.OracleDriver"

logAbandoned="true"

maxIdle="4"

maxTotal="32"

maxWaitMillis="10000"

name="jdbc/ticketsDS"33

removeAbandoned="true"

removeAbandonedTimeout="60"

url="jdbc:oracle:thin:@localhost:1521:XE"

username="ptickets"

password="PTickets"

/>

La aplicación ha sido probada e incluye los controladores de datos para Oracle 10g o superior, MySQL 5 o superior y SQL Server 2012 o superior.

**src/resources/tickets.properties**

Este es el archivo principal de configuración y controla la operación de la aplicación en uso. Cada una de las 3 contiene un archivo con las mismas entradas por lo cual se puede modificar como opera una sin afectar a las otras aunque no todas usan los mismos detalles.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Valor** | **Descripción** | **Default** | **Opcional** | **Aplica para** |
| datasource.name | Nombre JNDI de la fuente de datos | jdbc/ticketsDS | No | Todos |
| datasource.factory | Contexto de creación de conexiones | java:comp/env | Si | Todos |
| mail.host | Servidor de correo electrónico | localhost | No | 1,2 |
| mail.port | Puerto del servidor de correo | 25 | No | 1,2 |
| mail.user | Usuario de servidor de correo |  | Si | 1,2 |
| mail.secret | Contraseña del usuario servidor de correo |  | Si | 1,2 |
| mail.from | Remitente del correo | noreply | No | 1,2 |
| mail.subject | Título del correo electrónico | Notificación | No | 1,2 |
| mail.body | Texto del mensaje de correo |  | No | 1,2 |
| mail.secure | Si se debe usar una conexión segura al servidor de correo | 0 | No | 1,2 |
| mail.fakeTo | Destinatario simulado en envío de correo |  | Si | 1,2 |
| ldap.host | Servidor de autenticación de usuarios |  | No | 1,2 |
| ldap.port | Puerto del servidor de autenticación de usuarios |  | No | 1,2 |
| ldap.user | Usuario de servidor de autenticación de usuarios |  | No | 1,2 |
| ldap.secret | Contraseña del usuario servidor de autenticación de usuarios |  | No | 1,2 |
| ldap.branch | Rama acceso al servidor de autenticación de usuarios |  | No | 1,2 |
| ldap.group | Grupo de inicio de conexión al LDAP |  | Si | 1,2 |
| ldap.domain | Dominio de la empresa |  | Si | 1,2 |
| ws.timeout | Tiempo máximo de espera de peticiones a los webservices (ms) | 25000 | Si | 1,2 |
| ws.inc.url | 3 |  | No | 1,2 |
| ws.inc.pid | Clave única de acceso al webservice |  | No | 1,2 |
| ws.inc.user | Usuario de conexión al webservice |  | No | 1,2 |
| ws.inc.pwd | Contraseña del usuario de conexión al webservice |  | No | 1,2 |
| ws.wo.url | URL del webservice de requisiciones |  | No | 1,2 |
| ws.wo.pid | Clave única de acceso al webservice |  | No | 1,2 |
| ws.wo.user | Usuario de conexión al webservice |  | No | 1,2 |
| ws.wo.pwd | Contraseña del usuario de conexión al webservice |  | No | 1,2 |
| app.chat.enabled | Si se debe de generar o no el applet de chat | 1 | No | 1,2 |
| app.mock.remedy | Si se debe de simular o no el funcionamiento de remedy | 0 | No | 1,2 |
| app.soap.debug | Si se debe de capturar los sobres SOAP para debug | 0 | No | 2 |
| app.dev.mode | Aplicación en modo de desarrollo (mocks) | 0 | No | Todos |
| ws.people.url | URL del webservice de búsqueda de personas |  | No | 2 |
| ws.wo.user | Usuario de conexión al webservice |  | No | 2 |
| ws.wo.pwd | Contraseña del usuario de conexión al webservice |  | No | 2 |
| api.host | IP o Hostname del servidor de Remedy |  | No | 1,3 |
| api.port | Puerto de conexión del servidor de Remedy |  | No | 1,3 |
| api.user | Usuario de conexión a servidor de Remedy |  | No | 1,3 |
| api.secret | Contraseña del usuario de conexión a servidor de Remedy |  | No | 1,3 |

Valores de Aplica para:

Todos. Todas las aplicaciones

1. TicketsApi

2. Tickets

3. TicektsAdmin

### Anexo D. Contenido del repositorio en GitHub

|  |  |
| --- | --- |
| **Carpeta** | **Propósito** |
| Documentacion | Documentción del proyecto, incluyendo esta guía. |
| helpdeskClient | Código fuente del cliente de los webservice |
| Mocks | Simuladores de los webservices, proyectos en Eclipse |
| remedyApi | Documentación y jar del API de Remedy |
| Otros | Proyectos de SOAP, SQL, archivos con datos de prueba, etc. |
| TestWS | Proyecto de prueba a los simuladores de WS En Eclipse |
| tickets | Portal versión que usa Webservices |
| ticketsAdmin | Administrador del portal |
| ticketsApi | Portal versión que usa el API de Remedy™ |
| ticketsCommon | Código fuente común a todos los proyectos |