

Libertya ERP

Guía de instalación en Linux

Introducción

El presente documento detalla las actividades a realizar a fin de instalar Libertya ERP en sistemas operativos Linux. El mismo es una guía de referencia genérica, la cual no es específica para una distribución o versión en particular, lo que permite mantener su validez a lo largo del tiempo y que en términos generales pueda ser utilizada más allá de la distribución de linux donde se va a realizar la instalación.

Alternativa mediante máquina virtual para VirtualBox

Vale la pena mencionar que junto con cada release de Libertya, se libera también una Máquina Virtual para VirtualBox lista para usar. Por consiguiente, si se desea omitir este proceso de instalación, simplemente se puede optar por dicha opción.

Por ejemplo, al momento de escribir este documento, podrá descargar la versión 19.07 de Libertya ERP lista para usar desde el siguiente enlace:

<http://www.libertya.org/descargas/libertya-pyme-19-07/#maquina-virtual>

Requisitos

Para la utilización de Libertya ERP se requieren básicamente cuatro elementos:

1. Java
2. Postgres
3. Base de datos Libertya ERP
4. Binarios Libertya ERP

En los siguientes apartados veremos cómo realizar la instalación de cada uno de ellos.

Permisos

Se supone que se cuenta con los permisos necesarios a nivel sistema operativo a fin de ejecutar los comandos detallados en el presente documento.

Java

Descarga

Existen varias maneras de instalar Java. En el presente documento se utiliza una opción que no requiere instalación de paquetes dependientes de la distribución de Linux a utilizar.

Al momento de escribir este documento, Libertya soporta las versiones de Java 6 a 8. Esto no quiere decir que con versiones más recientes LY no funcione, sino que todavía no fue verificado y por consiguiente no se puede garantizar su correcto funcionamiento. Por consiguiente, se recomienda la instalación de la versión 8 de Java. Se puede utilizar tanto la distribución de Oracle como alguno de los builds de OpenJDK (AdoptOpenJDK, Zulu, Oracle OpenJDK, etc.).

Sea cual sea la distribución a usar, se debe realizar la descarga correspondiente. Por ejemplo, si se desea descargar AdoptOpenJDK, debemos dirigirnos al site <https://adoptopenjdk.net/> y una vez allí seleccionar la JVM correspondiente según la arquitectura de su sistema operativo. Siguiendo el ejemplo, se descargará la JVM OpenJDK 8 (LTS) para Linux x64, en este caso: *OpenJDK8U-jdk_x64_linux_hotspot_8u252b09.tar.gz*.

Instalación

Una vez descargado el archivo, el mismo debe descomprimirse y extraerse. En este ejemplo suponemos que la descarga se efectuó en **~/Descargas** y que ubicaremos los binarios de la JVM en **/opt**.

```
mv ~/Descargas/OpenJDK8U-jdk_x64_linux_hotspot_8u252b09.tar.gz /opt
cd /opt
gunzip OpenJDK8U-jdk_x64_linux_hotspot_8u252b09.tar.gz
tar -xvf OpenJDK8U-jdk_x64_linux_hotspot_8u252b09.tar
```

Esto dejará la versión de Java lista para ser utilizada en **/opt/jdk8u252-b09**. Ahora falta configurar el entorno para que se utilice esta versión.

Configuración

La manera más sencilla de configurar esto es modificando el archivo **/etc/profile** o el archivo **/root/.bashrc**, incorporando las siguientes líneas al final de dicho archivo:

```
export JAVA_HOME=/opt/jdk8u252-b09
export PATH=$JAVA_HOME/bin:$PATH
```

Luego de realizar este cambio, probar a iniciar sesión nuevamente y verificar desde la terminal si ya está tomando la versión de java correspondiente, por ejemplo mediante el siguiente comando:

```
java -version
```

Lo cual debería arrojar algo como resultado algo similar a lo siguiente (lógicamente variando según la versión instalada):

```
openjdk version "1.8.0_252"  
OpenJDK Runtime Environment (AdoptOpenJDK)(build 1.8.0_252-b09)  
OpenJDK 64-Bit Server VM (AdoptOpenJDK)(build 25.252-b09, mixed mode)
```

Alternativamente, en caso de tener instaladas varias máquinas virtuales de Java, también es posible configurar la JVM por defecto utilizando el comando *update-alternatives*.

Lo primero es verificar las opciones disponibles mediante el comando:

```
update-alternatives --config java
```

Esto mostrará las opciones actualmente disponibles:

Selección	Ruta	Prioridad	Estado

0	/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java	1081	modo automático
* 1	/opt/jdk1.6.0_45/bin/java	0	modo manual

Incluiremos entonces como seleccion 2 a la JVM descargada:

```
sudo update-alternatives --install /usr/bin/java java /opt/jdk8u252-b09/bin/java 2
```

Esto incorporará una nueva entrada para java, a lo último de la tabla de alternativas.

Selección	Ruta	Prioridad	Estado

0	/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java	1081	modo automático
* 1	/opt/jdk1.6.0_45/bin/java	0	modo manual
2	/opt/jdk8u252-b09/bin/java	3	modo manual

Luego se puede seleccionar la alternativa para la JVM recientemente instalada mediante el siguiente comando e indicando el número 2:

```
update-alternatives --config java
```

De esta forma, la tabla quedará configurada de la siguiente manera (notar el cambio en la posición del asterisco):

Selección	Ruta	Prioridad	Estado

0	/usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java	1081	modo automático
1	/opt/jdk1.6.0_45/bin/java	0	modo manual
* 2	/opt/jdk8u252-b09/bin/java	3	modo manual

Una vez configurada la JVM, se recomienda reiniciar la sesión en terminal y verificar si efectivamente la configuración quedó correctamente realizada. Para ésto, simplemente invocamos al comando java requiriendo su versión:

```
java -version
```

Lo cual debería arrojar algo como resultado algo similar a lo siguiente (variando según la versión):

```
openjdk version "1.8.0_252"  
OpenJDK Runtime Environment (AdoptOpenJDK)(build 1.8.0_252-b09)  
OpenJDK 64-Bit Server VM (AdoptOpenJDK)(build 25.252-b09, mixed mode)
```

PostgreSQL

Descarga

Existen varias maneras de instalar Postgres. En el presente documento se utiliza una opción que no requiere instalación de paquetes dependientes de la distribución de Linux a utilizar. Adicionalmente, esta alternativa será la misma más allá de la versión a instalar, dado que el procedimiento es siempre el mismo. De todas maneras, puede optar por otra opción si es que cuenta con la posibilidad de una alternativa más directa.

Al momento de escribir este documento, Libertya soporta las versiones de PostgreSQL 8.4 a 10.11. También puede utilizarse la versión 12.3 pero se deberá aplicar un parche de binarios (explicado en detalle en la sección Binarios). Esto no quiere decir que con versiones más recientes LY no funcione, sino que todavía no fue verificado y por consiguiente no se puede garantizar su correcto funcionamiento. Por consiguiente, se recomienda la instalación de la versión 10.

Instalación

En este documento realizaremos la instalación de Postgres mediante la compilación de los fuentes, a fin de evitar depender de repositorios y/o versiones según la distribución. Adicionalmente, estos pasos son siempre los mismos más allá de la versión de Postgres a instalar.

En este ejemplo instalaremos la versión 10.10. Para ésto, accederemos al sitio de descargas de Postgres, el cual actualmente se ubica en la URL <https://www.postgresql.org/ftp/source/>, y luego descargaremos la versión en cuestión, por ejemplo: <https://ftp.postgresql.org/pub/source/v10.10/postgresql-10.10.tar.gz>.

Una vez descargada, procederemos a descomprimir y extraer los fuentes. Suponiendo que la descarga se realizó en **~/Descargas**, ejecutaremos:

```
cd ~/Descargas
gunzip postgresql-10.10.tar.gz
tar -xvf postgresql-10.10.tar
```

Una vez realizados estos pasos, en **~/Descargas/postgresql-10.10** se encontrarán los fuentes listos para ser compilados.

Dependiendo la instalación de Linux, es probable que se requieran librerías adicionales para realizar estos pasos (ante esta situación, el *configure* se detendrá a fin de que se puedan instalar las librerías en cuestión, para luego reintentar nuevamente la acción):

- **gcc**
- **make**
- **libreadline-dev** (Debian) / **readline-devel** (Red Hat)
- **zlib1g-dev** (Debian) / **zlib-devel** (Red Hat)

En caso de no contar con estas librerías, simplemente deberán instalarse mediante el gestor correspondiente según la distribución de Linux (hasta que la ejecución del *configure* no arroje errores), por ejemplo:

```
apt-get install gcc
```

o por ejemplo:

```
yum install gcc
```

Los pasos para la compilación se encuentran detallados en el archivo **~/Descargas/postgresql-10.10/INSTALL**. Por ejemplo:

```
./configure
make
su
make install
adduser postgres
mkdir /usr/local/pgsql/data
chown postgres /usr/local/pgsql/data
su - postgres
/usr/local/pgsql/bin/initdb -D /usr/local/pgsql/data
/usr/local/pgsql/bin/postgres -D /usr/local/pgsql/data >logfile 2>&1 &
```

Se deberá ejecutar paso a paso cada uno de los comandos a fin de realizar la compilación e instalación de postgres.

Una vez finalizada la instalación, la instancia de postgres quedará ubicada en **/usr/local/pgsql**.

Configuración

La configuración que se detallará aquí es la que permite el acceso menos restrictivo al servicio. Por consiguiente, una vez verificado el correcto funcionamiento de Libertya ERP y el acceso a la base de datos, se sugiere ajustar la configuración de seguridad según considere necesario.

La configuración de PostgreSQL consta de dos simples pasos, los cuales se detallan a continuación:

Permisos de acceso

En el archivo **/usr/local/pgsql/data/pg_hba.conf** deberá incorporarse una nueva línea luego de la línea **IPv4 local connections**, con el siguiente contenido:

```
# IPv4 local connections:
host    all             all             0/0             trust
```

Direcciones de escucha

En el archivo **/usr/local/pgsql/data/postgresql.conf** deberá modificarse la línea **listen_address**, con el siguiente contenido:

```
listen_addresses = '*'           # what IP address(es) to listen on;
```

El dueño de estos archivos debe ser el usuario **postgres**, con lo cual se deberá modificar el owner de los mismos en caso de haber sido modificado el contenido de dichos archivos por otro usuario:

```
chown postgres /usr/local/pgsql/data/pg_hba.conf
chown postgres /usr/local/pgsql/data/postgresql.conf
```

Variables de entorno

Incorporar en **/etc/profile** o **/root/.bashrc** las siguientes entradas al final del archivo a fin de especificar la ubicación de postgres y sus bases de datos:

```
export PGDATA=/usr/local/pgsql/data
export PATH=/usr/local/pgsql/bin:$PATH
```

Iniciando postgres y verificando conexión

En caso de querer iniciar PostgreSQL luego de - por ejemplo - reiniciar el sistema, es posible hacerlo mediante:

```
su - postgres
/usr/local/pgsql/bin/postgres -D /usr/local/pgsql/data >logfile 2>&1 &
exit
```

Si desea verificar la conexión a la base de datos, puede probar a ejecutar:

```
/usr/local/pgsql/bin/psql -h localhost -U postgres -c "select 1"
```

Esto deberá retornar un resultado como el siguiente, lo cual indica que el servicio de postgres se encuentra corriendo y escuchando:

```
?column?
-----
         1
(1 row)
```

Base de datos Libertya ERP

Descarga

Una vez instalado PostgreSQL, ya es posible descargar y restaurar la base de datos de Libertya. Más allá de la versión a utilizar, los pasos son siempre los mismos. En este ejemplo utilizaremos la versión 19.07AR para postgres 9.5 o superior, la cual podemos descargar desde:

https://sourceforge.net/projects/libertya/files/libertya/release/1907/multi-platform/dump_libertya_1907ar_postgres9.5.sql.zip

Notar que dependiendo la versión de PostgreSQL que estemos utilizando, existen distintas descargas. Por ejemplo, en caso de estar utilizando PostgreSQL 9.4 o inferior, debemos descargar:

https://sourceforge.net/projects/libertya/files/libertya/release/1907/multi-platform/dump_libertya_1907ar.sql.zip/

De manera similar, si deseamos descargar otras versiones de Libertya, deberemos buscar la opción correspondiente. Por ejemplo, si deseamos descargar Libertya PyME 19.07AR para Postgres 9.5 o superior, debemos descargar:

https://sourceforge.net/projects/libertya/files/libertya/release/1907/PyME/dump_libertya_1907ar_pyme_postgres9.5.sql.zip

Nota: La estructura de archivos a descargar respetará siempre el mismo formato más allá del release de Libertya a utilizar (18.06, 19.07, etc.).

Creación de usuario libertya (a nivel postgres)

Libertya utiliza un usuario y schema (de postgres) llamados libertya. Por única vez será necesario entonces realizar la creación correspondiente. En caso de haber realizado instalaciones previas de Libertya en la instancia en cuestión, este paso no deberá ser necesario.

```
psql -h localhost -U postgres -c "CREATE ROLE libertya LOGIN PASSWORD 'libertya' SUPERUSER CREATEDB CREATEROLE VALID UNTIL 'infinity' IN ROLE postgres"
```

Unicamente para versiones de postgres anteriores a la 9.0: es necesario crear el lenguaje plpgsql en la base de datos template1 para que cada vez que se cree una nueva base de datos la misma ya contenga el lenguaje requerido por las funciones que contiene la base de datos de libertya.

```
createlang -U postgres -d template1 plpgsql
```

Creación Base de Datos

Una vez descargado el archivo con el dump de la base de datos, deberemos descomprimirlo, para esto ejecutaremos el siguiente comando suponiendo que la descarga se efectuó en ~/Descargas:


```
cd ~/Descargas
unzip dump_libertya_1907ar_postgres9.5.sql.zip
```

Posteriormente debemos crear la base de datos y volcar allí el contenido del archivo descargado. El nombre de la base de datos puede ser cualquiera, en este caso utilizaremos el nombre **libertya_prod**:

```
cd ~/Descargas

psql -h localhost -U postgres -c "CREATE DATABASE libertya_prod WITH ENCODING='UTF8'
OWNER=libertya TEMPLATE=template1;"

psql -U libertya -d libertya_prod -f dump_libertya_1907ar_postgres9.5.sql
```

El último comando efectuará el restore del dump descargado en la base de datos **libertya_prod**, lo cual requerirá de cierto tiempo para ser completado.

Binarios Libertya ERP

Descarga

Una vez instalado Java, ya es posible descargar e instalar los binarios de Libertya ERP. Más allá de la versión a utilizar, los pasos son siempre los mismos. En este ejemplo utilizaremos la versión de binarios 19.07, la cual sirve tanto para Libertya 19.07 como para Libertya 19.07AR.

Importante: recordar siempre utilizar la misma versión de release (18.06, 19.07, etc.) a nivel base de datos y a nivel de binarios. No es correcto mezclar binarios de un release con la base de datos de otro release.

Los binarios de Libertya ERP 19.07 puede descargarse desde el siguiente enlace:

https://sourceforge.net/projects/libertya/files/libertya/release/1907/multi-platform/ServidorOXP_V19.07.zip

***Nota:** Los binarios son únicos tanto para la versión 19.07 como para la versión 19.07AR. Para la versión PyME de Libertya existen binarios específicos, así como base de datos específica.*

Una vez descargado los binarios debemos descomprimirlos. Se recomienda descomprimirlos en el directorio raíz:

```
cd ~/Descargas
mv ServidorOXP_V19.07.zip /
cd /
unzip ServidorOXP_V19.07.zip
```

Esto generará el directorio **/ServidorOXP** con los binarios de Libertya ERP. Luego debemos darle permisos de ejecución a los archivos .sh allí contenidos:

```
cd /ServidorOXP
chmod ugo+x *.sh -R
```

Configuración

Lo primero a realizar es configurar la variable de entorno **OXP_HOME** en el archivo **/etc/profile** o **/root/.bashrc**, incorporando la siguiente entrada al final de dicho archivo:

```
export OXP_HOME=/ServidorOXP
```

Luego de descomprimir los binarios será necesario configurar la instancia. Se recomienda realizar estos pasos con el usuario **root** para evitar problemas de permisos (más adelante veremos alternativas para mejorar la seguridad)

Nota: Si se utiliza postgres 12 para la instalación, en este momento se deberá colocar el parche **org.libertya.core.fix_MSequence_r3032.jar** dentro del directorio **/ServidorOXP/lib/plugins**. y luego pasar a la configuración siguiente.

El parche puede obtenerse desde sourceforge:

https://sourceforge.net/projects/libertya/files/libertya/dev/support/22.0/postgres12/org.libertya.core.fix_MSequence_r3032.jar/download

Opción 1: Configurando mediante GUI

```
cd /ServidorOXP
./Configurar.sh
```

Esto abrirá una ventana con las opciones de configuración, deberá completarse la información acordemente, indicando la ubicación de java, el nombre de la base de datos, la contraseña de los usuarios postgres y libertya, los puertos para las distintas actividades, etc.

En lo que refiere a los puertos, se recomienda utilizar los siguientes valores:

- Puerto JNP: 1099
- Puerto Web: 8080
- Puerto SSL: 8443

Siguiendo el ejemplo visto hasta el momento, la ventana de configuración debería quedar configurada de esta manera (lógicamente con variaciones dado que - por ejemplo - el equipo probablemente tenga un nombre diferente a pc-usuario):

Configuración del servidor Libertya

Archivo Ayuda

Directorio base de Java JDK	/opt/jdk8u252-b09	Tipo de Java	sun
Directorio base de Libertya	/ServidorOXP	Contraseña de keyStore
Servidor de aplicaciones	pc-usuario	Tipo de servidor	jboss
Directorio de despliegue	/ServidorOXP/jboss/server/	Puerto JNP	1099
Puerto Web	8080	Puerto SSL	8443
Servidor base de datos	pc-usuario	Tipo de base de datos	PostgreSQL
Nombre de la base de datos	libertya_prod	Contraseña de superusuario
Puerto de la base de datos	5432	Contraseña de la base de datos
Usuario de la base de datos	libertya	Correo del administrador	info@pc-usuario
Servidor de correo	pc-usuario	Contraseña de correo
Usuario de correo	info		

Comprobar ? Guardar

Una vez realizada la configuración, es posible hacer click en el botón **Comprobar**. Esto verificará que los valores indicados sean los correctos, o presentará un error según corresponda. Ciertos errores pueden ignorarse, como por ejemplo si el servidor de correo presenta error. Sin embargo, si por ejemplo hay un error de puerto usado o de imposibilidad de conexión a la base de datos, se deberá verificar el problema y resolver el inconveniente.

Por último deberá hacer click en Guardar a fin de completar el proceso. Esta actividad puede demorar varios minutos, y al finalizar se cerrará la ventana de configuración. La instancia ya estará lista para ser usada.

Opción 2: Configurando desde terminal

Si se está realizando la instalación en una instancia sin interfaz gráfica, también es posible realizar la configuración desde terminal. Para esto es necesario copiar el template de configuración y luego editarlo:

```
cd /ServidorOXP
cp LibertyaEnvTemplate.properties LibertyaEnv.properties
nano LibertyaEnv.properties
```

*Se deberá configurar adecuadamente el archivo **LibertyaEnv.properties** de manera similar a lo realizado mediante la configuración gráfica. Una vez realizado esto, se deberá ejecutar el **ConfigurarAuto.sh**, el cual realiza la misma actividad que al clicar el botón **Guardar** desde la ventana de configuración.*

Al configurar **LibertyaEnv.properties** cambiar los puertos por defecto de a los utilizados en el ejemplo anterior: Puerto JNP: 1099, Puerto Web: 8080, Puerto SSL: 8443 en caso contrario pueden presentarse problemas al iniciar el servidor automáticamente (daemon libertyad).

```
cd /ServidorOXP
./ConfigurarAuto.sh
```

Iniciando el servidor de aplicaciones

El servidor de aplicaciones se encarga de varias tareas como brindar el portal web de Libertya ERP, generar las entradas contables automáticamente, realizar tareas programadas en segundo plano, etc. Para cualquiera de estas actividades es necesario iniciar el servidor de aplicaciones. Para esto debe ejecutar el siguiente comando, el cual ejecutaremos mediante el usuario **root** (luego veremos alternativas para mejorar la seguridad):

```
cd /ServidorOXP/utils
./IniciarServidor.sh
```

Esto iniciará el proceso de booteo del contenedor web JBoss. Si no se presentan inconvenientes, se visualizará como última salida en la terminal un mensaje similar al siguiente:

```
14:32:04,823 INFO [Server] JBoss (MX MicroKernel) [4.0.2 (build: CVSTag=JBoss_4_0_2
date=200505022023)] Started in 18s:277ms
```

Esto significa que el servidor ha iniciado correctamente. Si se desea finalizar la ejecución del mismo, simplemente podemos presionar **CTRL+C** desde el teclado.

Accediendo a Libertya ERP

Ver el documento **Iniciando Libertya**.

Configuraciones avanzadas

Importante: estos pasos pueden variar dependiendo la distribución del sistema operativo.

Inicio de PostgreSQL en arranque del sistema

Para que PostgreSQL pueda iniciar automáticamente al iniciar el sistema operativo, se pueden seguir los siguientes pasos.

Dentro de los fuentes de postgres se incluye un script de base, el cual debe ser copiado al init.d del sistema operativo. Suponiendo que los fuentes de postgres fueron descargados en **~/Descargas**, y siguiendo con el ejemplo de versión 10.10 entonces deberemos realizar:

```
cp ~/Descargas/postgresql-10.10/contrib/start-scripts/linux /etc/init.d/postgresql
chmod ugo+x /etc/init.d/postgresql
```

Luego debemos configurar para que el script se ejecute al inicio. Esto dependerá de la distribución de Linux, por ejemplo Red Hat o derivados (también de la versión, ver **NOTA** a continuación).

```
chkconfig --add postgresql
```

o bien para Debian o derivados:

```
update-rc.d postgresql defaults
```

NOTA: En versiones más recientes, es posible utilizar el comando **systemctl**. Para ésto se deben incorporar las siguientes líneas al inicio del script ubicado en **/etc/init.d/postgresql**:

```
#!/bin/sh

### BEGIN INIT INFO
# Provides: postgresql
# Required-Start: $local_fs $network
# Required-Stop:
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
### END INIT INFO
```

Posteriormente ejecutar:

```
systemctl enable postgresql
```

En caso de que el servicio no inicie correctamente al bootear el equipo, se deberá verificar el origen del problema. En ocasiones los inconvenientes de inicio automático se originan en la falta de permisos para el acceso y ejecución de ciertos archivos.

Ejecutando el siguiente comando podremos determinar la causa del problema:

```
journalctl -u postgresql.service
```

Esto devolverá por ejemplo información de error como la siguiente:

```
localhost.localdomain postgresql[783]: /etc/rc.d/init.d/postgresql: línea 101: /bin/su: Permiso denegado
localhost.localdomain systemd[1]: postgresql.service: Control process exited, code=exited, status=126/n/a
localhost.localdomain systemd[1]: postgresql.service: Failed with result 'exit-code'.
localhost.localdomain systemd[1]: Failed to start SYSV: PostgreSQL RDBMS.
```

En función del error especificado se deberá actuar acordemente a fin de resolver el problema en cuestión. Dado que el script postgresql requiere el uso del comando *su*, las acciones correctivas pueden abarcar la necesidad de realizar modificaciones sobre el grupo wheel de */etc/group*, sobre los archivos */etc/pam.d/su* y/o */etc/pam.d/sudo*, entre otros.

Para verificar el inicio o detención manualmente, podemos ejecutar:

```
service postgresql start
service postgresql stop
```

Si se optó por utilizar la versión con *systemctl* (*systemd*) el inicio y detencion manual se realiza con los siguientes comandos:

```
systemctl start postgresql
systemctl stop postgresql
```

Y puede verificarse el estado de la siguiente forma:

```
systemctl status postgresql
```

Deshabilitar SELinux

SELinux puede traer problemas a la hora de configurar el daemon para inicio automático de PostgreSQL, para deshabilitarlo modificar el archivo ***/etc/selinux/config*** ubicar la linea

```
SELINUX=enforcing
```

y cambiarla por

```
SELINUX=disabled
```

guardar los cambios y cerrar el archivo. Ahora en el próximo reboot SELinux estará deshabilitado

Inicio de Libertya ERP en arranque del sistema

Para que Libertya ERP pueda iniciar automáticamente al iniciar el sistema operativo, se pueden seguir los siguientes pasos.

El primer paso es crear el usuario **libertya** (a nivel de sistema operativo) dado que de esta manera será dicho usuario quien inicie el servidor de aplicaciones, lo cual es una medida de seguridad a fin de evitar que el superusuario **root** sea quien realice esta tarea.

```
adduser libertya
```

Por consiguiente, a partir de ahora el dueño de la carpeta **/ServidorOXP** y todos sus archivos deberán ser del usuario **libertya**:

```
cd /  
chown libertya:libertya /ServidorOXP -R
```

Dado que el inicio del servicio se iniciará internamente con el usuario llamado libertya, se creará el PID del proceso con ese usuario. Como en **/var/run** no tendremos permisos para escritura al iniciar el proceso, se debe crear la carpeta **libertya** dentro de ese directorio, y asignarle el usuario **libertya** como dueño:

```
cd /var/run  
mkdir libertya  
chown libertya libertya -R
```

De manera similar a PostgreSQL, Libertya ERP distribuye el script necesario para incorporarlo a la configuración de inicio. Este script se encuentra en **/ServidorOXP/utils/unix** y existen variantes según la distribución del sistema operativo:

- libertyad_centos.sh
- libertyad_fedora.sh
- libertyad_ubuntu-debian.sh

Para el ejemplo, utilizaremos la opción **libertyad_ubuntu-debian.sh**. Copiamos el script y damos permisos de ejecución al mismo:

```
cp /ServidorOXP/utils/unix/libertyad_ubuntu-debian.sh /etc/init.d/libertyad  
chmod ugo+x /etc/init.d/libertyad
```

Luego debemos configurar a fin de que el script se ejecute al inicio, similar a lo realizado con PostgreSQL. Esto dependerá de la distribución de Linux, por ejemplo Red Hat o derivados:

```
chkconfig --add libertyad
```

o bien para Debian o derivados:


```
update-rc.d libertyad defaults
```

NOTA: En versiones más recientes, es posible utilizar el comando **systemctl**. Para ésto se deben incorporar las siguientes líneas al inicio del script ubicado en /etc/init.d/libertyad:

```
#!/bin/bash

### BEGIN INIT INFO
# Provides: libertyad
# Required-Start: $ALL
# Required-Stop:
# Default-Start: 2 3 4 5
# Default-Stop: 0 1 6
### END INIT INFO
```

Posteriormente ejecutar:

```
sudo systemctl enable libertyad
```

Para verificar el inicio o detención manualmente, podemos hacer lo siguiente (vale mencionar que específicamente para distros Red-Hat, CentOS o derivados, se requiere previamente realizar los pasos detallados a continuación):

```
service libertyad start
service libertyad stop
```

Si se optó por utilizar la versión con systemctl (systemd) el inicio y detención manual se realiza con los siguientes comandos:

```
systemctl start libertyad
systemctl stop libertyad
```

Y puede verificarse el estado de la siguiente forma:

```
systemctl status libertyad
```

Importante: el script **libertyad** utiliza el comando **start-stop-daemon**, el cual es propio de distribuciones Debian. Este comando puede no estar disponible bajo Red Hat y derivados, con lo cual es necesario instalarlo manualmente.

Para ésto, será necesario descargar el archivo con los fuentes de la funcionalidad start-stop-daemon, para luego compilarlos:

```
cd /usr/local/src
wget http://developer.axis.com/download/distribution/apps-sys-utils-start-stop-daemon-IR1_9_18-2.tar.gz
```

Desempaquetar el archivo comprimido

```
tar zxvf apps-sys-utils-start-stop-daemon-IR1_9_18-2.tar.gz  
cd apps/sys-utils/start-stop-daemon-IR1_9_18-2/
```

Compilar la aplicación

```
gcc start-stop-daemon.c -o start-stop-daemon
```

Copiar el binario de start-stop-daemon en /usr/sbin

```
cp start-stop-daemon /usr/sbin
```

En caso de que el servicio no inicie correctamente, se deberá verificar el origen del problema ejecutando el siguiente comando a fin de poder resolver el inconveniente:

```
journalctl -u libertyad.service
```

En función del error especificado se deberá actuar acordemente a fin de resolver el problema en cuestión, de manera similar a lo realizado en la configuración de inicio automático de postgres.