

# PRÁCTICA 5 | TEMA 3

## CONFIGURACIÓN DE COPIA DE SEGURIDAD AUTOMATIZADA EN ODOO

Stanislav Krastev

SISTEMAS DE GESTIÓN 2º DAM

# Instrucciones para crear una copia de seguridad automatizada de odoo en Ubuntu

Primero creamos en la bbdd que tenemos en inventario un par de productos, y creamos desde la web una copia de seguridad y por si acaso falla, la duplicamos y trabajamos con la duplicada...

Accedemos a Odoo con nuestro navegador de confianza



Introducimos los credenciales

**Correo electrónico**

stani@stani.es

**Contraseña**

.....

Iniciar sesión

[¿No tienes una cuenta?](#) [Restablecer contraseña](#)

[Gestionar Bases de datos](#) | [Respaldo por Odoo](#)

Creamos los productos

0,00...  
A Mano

0,00...  
Pronosticado

Movimientos  
productos

0,00...  
Vendido

0  
Reglas de a...

Nombre del producto

Maceta

☒ Puede ser vendido

☒ Puede ser comprado

Información General

Ventas

Compra

Inventario

Tipo de producto

Almacenable

Categoría de producto

All

Referencia interna

Código de barras

Precio de venta

25,00 €

☒ Impuestos del cliente


Coste

18

IVA 21% (Bienes)

Notas internas

Esta maceta es la puta ####



Inventario

Información general

Operaciones


Productos

Informes

Configuración

Productos


Crear



Maceta

Precio: 25,00 €

A mano: 170,00 Unidades



Tijeras de podar

Precio: 18,00 €

A mano: 120,00 Unidades

Para hacer la copia de seguridad de seguridad tenemos que cerrar sesión y gestionar las bases de datos

Iniciar sesión

¿No tienes una cuenta?

Restablecer contraseña

Iniciar sesión como superusuario

Gestionar Bases de datos

Respaldo por Odoo

Pulsamos en **Bakcup**

Página 2 | 11



practica4

Backup

Duplicate

Delete

Create Database

Restore Database

Set Master Password

Nos pide introducir credenciales y además elegir el formato de la base de datos zip es para Windows y pg\_dump para Linux, elegimos pg\_dump

Backup Database

×

Master Password

••••

Database Name

practica4

Backup Format

pg\_dump custom format (without filestore)

zip (includes filestore)

pg\_dump custom format (without filestore)

Backup

Para ver la base de datos en el servidor Ubuntu tenemos que pulsar en **Restore Database** y elegir la copia que acabamos de descargar

Restore Database

Rellenamos los campos y pulsamos en **Continuar**

### Restore Database

Master Password

File

Database Name

This database might have been moved or copied.

In order to avoid conflicts between databases, Odoo needs to know if this database was moved or copied.

If you don't know, answer "This database is a copy".

☒ This database is a copy

☐ This database was moved

Continue

Esperamos que termine de cargar la base de datos y una vez termine nos devuelve a la pantalla de gestionar las bases de datos de Odoo, podemos ver como tenemos una base de datos extra que antes no teníamos.

odoo

practica4	Backup	Duplicate	Delete
practica4_copia	Backup	Duplicate	Delete

Create DatabaseRestore DatabaseSet Master Password

Ahora nos dirigimos al terminal de nuestro servidor Ubuntu;

Algunas de las opciones de pg\_dump

```
stani@stani:~$ pg_dump --help
pg_dump dumps a database as a text file or to other formats.

Usage:
  pg_dump [OPTION]... [DBNAME]

General options:
  -f, --file=FILENAME          output file or directory name
  -F, --format=c|d|t|p        output file format (custom, directory, tar,
                              plain text (default))
  -j, --jobs=NUM               use this many parallel jobs to dump
  -v, --verbose                 verbose mode
  -V, --version                 output version information, then exit
  -Z, --compress=0-9           compression level for compressed formats
  --lock-wait-timeout=TIMEOUT  fail after waiting TIMEOUT for a table lock
  --no-sync                     do not wait for changes to be written safely to disk
  -?, --help                   show this help, then exit

Options controlling the output content:
  -a, --data-only              dump only the data, not the schema
  -b, --blobs                  include large objects in dump
  -B, --no-blobs               exclude large objects in dump
  -c, --clean                  clean (drop) database objects before recreating
  -C, --create                  include commands to create database in dump
  -E, --encoding=ENCODING      dump the data in encoding ENCODING
  -n, --schema=PATTERN         dump the specified schema(s) only
  -N, --exclude-schema=PATTERN do NOT dump the specified schema(s)
  -O, --no-owner                skip restoration of object ownership in
                              plain-text format
  -s, --schema-only             dump only the schema, no data
  -S, --superuser=NAME          superuser user name to use in plain-text format
  -t, --table=PATTERN           dump the specified table(s) only
  -T, --exclude-table=PATTERN   do NOT dump the specified table(s)
  -x, --no-privileges           do not dump privileges (grant/revoke)
  --binary-upgrade              for use by upgrade utilities only
  --column-inserts              dump data as INSERT commands with column names
  --disable-dollar-quoting      disable dollar quoting, use SQL standard quoting
  --disable-triggers            disable triggers during data-only restore
  --enable-row-security          enable row security (dump only content user has
                              access to)
  --exclude-table-data=PATTERN  do NOT dump data for the specified table(s)
  --extra-float-digits=NUM      override default setting for extra_float_digits
  --if-exists                   use IF EXISTS when dropping objects
  --inserts                     dump data as INSERT commands, rather than COPY
```

Recordar para hacer cosas con la bbdd hay que entrar con el usuario odoo

Hay que conseguir crear el fichero copiaSQL en formato copiafecha\_hora.sql

Comando: **su odoo**

Comando: **\$ pg\_dump prueba > copia\$(date +"%y%m%d\_%H%M").sql**

```
odoo@stani: ~/copias_de_seguridad
odoo@stani:~/copias_de_seguridad$ pg_dump practica4_copia > copia$(date +"%y%m%d_%H%M").sql
odoo@stani:~/copias_de_seguridad$ ls
copia221122_0758.sql
odoo@stani:~/copias_de_seguridad$
```

Para restaurar la Copia de Seguridad... Paramos el servicio odoo con systemctl y luego nos aseguramos de que esté parado creamos la bbdd de nuevo, hay que crearla con el mismo nombre que tenía

Paramos el servicio con el comando: **systemctl stop odoo**  
Comprobamos el estado del servicio con el comando: **systemctl status odoo**

```
odoo@stani: ~/copias_de_seguridad
odoo@stani:~/copias_de_seguridad$ systemctl stop odoo
==== AUTHENTICATING FOR org.freedesktop.systemd1.manage-units ====
Authentication is required to stop 'odoo.service'.
Authenticating as: stani
Password:
==== AUTHENTICATION COMPLETE ====
odoo@stani:~/copias_de_seguridad$ systemctl status odoo
● odoo.service - Odoo14
   Loaded: loaded (/etc/systemd/system/odoo.service; enabled; vendor preset: enabled)
   Active: inactive (dead) since Tue 2022-11-22 08:03:19 UTC; 5s ago
     Process: 891 ExecStart=/opt/odoo/odoo-venv/bin/python3 /opt/odoo/odoo/odoo-bin -c /
   Main PID: 891 (code=exited, status=0/SUCCESS)

nov 22 07:54:02 stani odoo[891]: 2022-11-22 07:54:02,273 891 INFO ? odoo: Odoo version
nov 22 07:54:02 stani odoo[891]: 2022-11-22 07:54:02,274 891 INFO ? odoo: Using configu
nov 22 07:54:02 stani odoo[891]: 2022-11-22 07:54:02,274 891 INFO ? odoo: addons paths:
nov 22 07:54:02 stani odoo[891]: 2022-11-22 07:54:02,274 891 INFO ? odoo: database: odo
nov 22 07:54:02 stani odoo[891]: 2022-11-22 07:54:02,686 891 INFO ? odoo.addons.base.mo
nov 22 07:54:03 stani odoo[891]: 2022-11-22 07:54:03,155 891 INFO ? odoo.service.server
nov 22 08:03:19 stani odoo[891]: 2022-11-22 08:03:19,102 891 INFO ? odoo.service.server
nov 22 08:03:19 stani odoo[891]: 2022-11-22 08:03:19,102 891 INFO ? odoo.service.server
Warning: journal has been rotated since unit was started and some journal files were no
lines 1-15/15 (END)
```

```
$ createdb -help
```

```
$ psql nombre_BBDD < copiaDia_fecha.sql
```

(Recordar antes para el servicio de odoo)

Una vez ejecutado el comando psql ... se empezará a restaurar la copia de la base de datos (se verán múltiples ejecuciones en la pantalla de sentencias sql, una vez haya terminado terminará y tendremos control del terminal de nuevo)

**Automatizar la BBDD Creamos un fichero shell .sh donde escribimos el comando para realizar la copia, solo tendrá dentro ese comando.**

Creamos un archivo con el comando: **nano copia\_automatizada.sh**

Dentro de ese archivo escribimos la siguiente línea

```
odoo@stani: ~/copias_de_seguridad
GNU nano 4.8
pg_dump practica4 > /home/odoo/copias_de_seguridad/copia$(date +"%y%m%d_%H%M").sql
```

Guardamos los cambios.

```
odoo@stani: ~/copias_de_seguridad
odoo@stani:~/copias_de_seguridad$ ls
copia221122_0758.sql  copia_automatizada.sh
odoo@stani:~/copias_de_seguridad$
```

**Hay que darle permisos de ejecución +x**

```
odoo@stani:~/copias_de_seguridad$ ls -la
total 13876
drwxrwxr-x 2 odoo odoo 4096 nov 22 08:14 .
drwxr-xr-x 5 odoo odoo 4096 nov 21 13:02 ..
-rw-rw-r-- 1 odoo odoo 14194985 nov 22 07:58 copia221122_0758.sql
-rw-rw-r-- 1 odoo odoo 88 nov 22 08:14 copia_automatizada.sh
odoo@stani:~/copias_de_seguridad$
```

No tiene permisos de ejecución, procedemos a dárselos.



```

odoo@stani:~/copias_de_seguridad$ chmod +x copia_automatizada.sh
odoo@stani:~/copias_de_seguridad$ ls -la
total 13876
drwxrwxr-x 2 odoo odoo      4096 nov 22 08:14 .
drwxr-xr-x 5 odoo odoo      4096 nov 21 13:02 ..
-rw-rw-r-- 1 odoo odoo 14194985 nov 22 07:58 copia221122_0758.sql
-rwxrwxr-x 1 odoo odoo       88 nov 22 08:14 copia_automatizada.sh
odoo@stani:~/copias_de_seguridad$

```

Ya tiene permisos para ejecutarse.

Estudiamos el comando `$ crontab -help` Planificamos tarea en el crontab, solo tenemos que decirle el día y la hora que queramos que se ejecute, mirando la ayuda veríamos como sería el comando los que nos interesan son `-e` para editar y `-l` para ver los servicios que tenemos activados.

```

odoo@stani: ~/copias_de_seguridad
odoo@stani:~/copias_de_seguridad$ crontab --help
crontab: invalid option -- '-'
crontab: usage error: unrecognized option
usage: crontab [-u user] file
       crontab [ -u user ] [ -i ] { -e | -l | -r }
              (default operation is replace, per 1003.2)
       -e      (edit user's crontab)
       -l      (list user's crontab)
       -r      (delete user's crontab)
       -i      (prompt before deleting user's crontab)
odoo@stani:~/copias_de_seguridad$

```

Procedemos:

```

odoo@stani: ~/copias_de_seguridad
odoo@stani:~/copias_de_seguridad$ crontab -l
no crontab for odoo
odoo@stani:~/copias_de_seguridad$

```

Ejecutamos: **crontab -e**

```
Select an editor. To change later, run 'select-editor'.
 1. /bin/nano          <---- easiest
 2. /usr/bin/vim.basic
 3. /usr/bin/vim.tiny
 4. /bin/ed

Choose 1-4 [1]:
```

Elegimos opción (nano por ejemplo)

*Nota: a partir de ahora, cada vez que ejecutemos **crontab -e** se ejecutará por defecto con el editor nano*

Escribimos lo siguiente: 0 2 \* \* \* (la ruta del script que se ejecutará) y guardamos cambios

0 2 \* \* \* significa una copia diaria a las 2 AM (madrugada)

```
odoo@stani: ~/copias_de_seguridad
GNU nano 4.8
# COPIA DE SEGURIDAD DIARIA A LAS 2 DE LA MADRUGADA DE LA BASE DE DATOS practica4
0 2 * * * /home/odoo/copias_de_seguridad/copia_automatizada.sh

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
```

```

crontab: installing new crontab
odoo@stani:~/copias_de_seguridad$ crontab -l
# COPIA DE SEGURIDAD DIARIA A LAS 2 DE LA MADRUGADA DE LA BASE DE DATOS practica4
0 2 * * * /home/odoo/copias_de_seguridad/copia_automatizada.sh

# Edit this file to introduce tasks to be run by cron.
#
# Each task to run has to be defined through a single line
# indicating with different fields when the task will be run
# and what command to run for the task
#
# To define the time you can provide concrete values for
# minute (m), hour (h), day of month (dom), month (mon),
# and day of week (dow) or use '*' in these fields (for 'any').
#
# Notice that tasks will be started based on the cron's system
# daemon's notion of time and timezones.
#
# Output of the crontab jobs (including errors) is sent through
# email to the user the crontab file belongs to (unless redirected).
#
# For example, you can run a backup of all your user accounts
# at 5 a.m every week with:
# 0 5 * * 1 tar -zcf /var/backups/home.tgz /home/
#
# For more information see the manual pages of crontab(5) and cron(8)
#
# m h dom mon dow   command
odoo@stani:~/copias_de_seguridad$

```

En la memoria tienes que mostrar como la planificación se ha cumplido con \$ ls -lht que se ha recuperado el fichero con el día y la hora que hemos establecido en el programador de tareas.

Para ver si funciona modificamos el crontab y ponemos que se ejecute lo mas pronto posible (2 minutos posterior a la hora actual)

```

odoo@stani: ~/copias_de_seguridad
GNU nano 4.8 /tr
# COPIA DE SEGURIDAD DIARIA A LAS 2 DE LA MADRUGADA DE LA BASE DE DATOS practica4
47 8 * * * /home/odoo/copias_de_seguridad/copia_automatizada.sh

odoo@stani:~/copias_de_seguridad$ date
mar 22 nov 2022 08:46:17 UTC

```

Comprobamos con el comando **ls -lht**

```
odoo@stani:~/copias_de_seguridad$ ls -lht
total 28M
-rw-rw-r-- 1 odoo odoo 14M nov 22 08:47 copia221122_0847.sql
-rwxrwxr-x 1 odoo odoo  83 nov 22 08:33 copia_automatizada.sh
-rw-rw-r-- 1 odoo odoo 14M nov 22 07:58 copia221122_0758.sql
odoo@stani:~/copias_de_seguridad$
```

Efectivamente se ha creado la copia automatizada el día 22 / 11 / 2022 a las 8:47 como especificamos en el crontab