# 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...

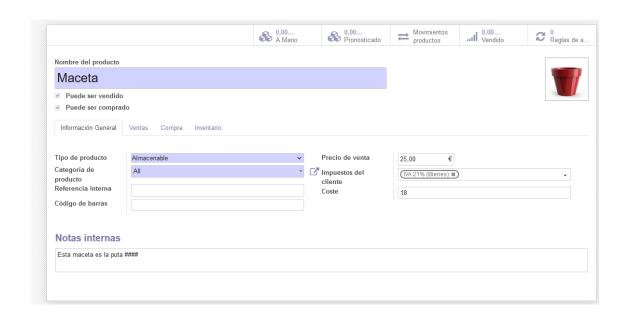
Accedemos a Odoo con nuestro navegador de confianza

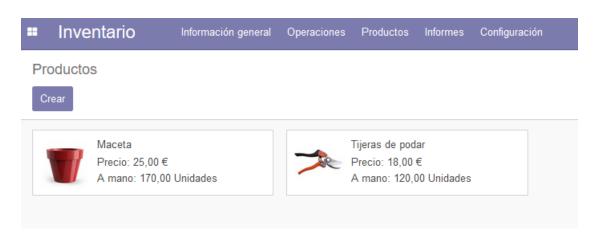


### Introducimos los credenciales



Creamos los productos





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

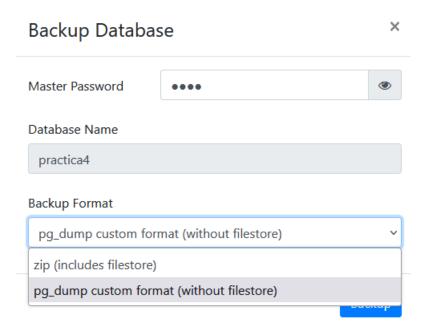


Pulsamos en Bakcup

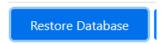




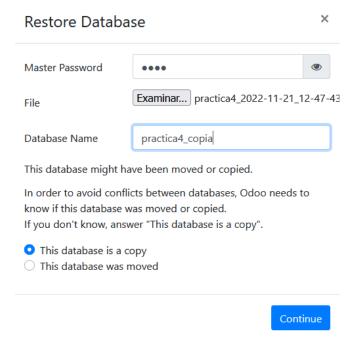
Nos pide introducir credenciales y además elegir el formato de la base de datos zip es para Windows y pg\_dump para Linux, elegimos pg\_dump



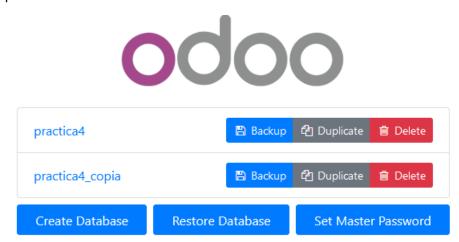
Para ver la base de datos en el servidor Ubuntu tenemos que pulsar en **Restore Database** y elegir la copia que acabamos de descargar



## Rellenamos los campos y pulsamos en Continuar



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.



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
                               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
                               do not wait for changes to be written safely to d
 --no-sync
isk
 -?, --help
                               show this help, then exit
Options controlling the output content:
 -a, --data-only
-b, --blobs
                               dump only the data, not the schema
                               include large objects in dump
                               exclude large objects in dump
 -B, --no-blobs
                               clean (drop) database objects before recreating
 -C, --create
                               include commands to create database in dump
                            dump the data in encoding ENCODING
 -E, --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
                               dump only the schema, no data
 -s, --schema-only
                            superuser user name to use in plain-text format dump the specified table(s) only
 -S, --superuser=NAME
 -t, --table=PATTERN
 -T, --exclude-table=PATTERN do NOT dump the specified table(s)
 -x, --no-privileges
                             do not dump privileges (grant/revoke)
                               for use by upgrade utilities only
 --binary-upgrade
 --column-inserts
                               dump data as INSERT commands with column names
                               disable dollar quoting, use SQL standard quoting disable triggers during data-only restore
 --disable-dollar-quoting
 --disable-triggers
 --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
                                         as INSERT co
```

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

```
doo@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

```
doo@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:
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</pre>
```

(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 ejecuciónes 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

```
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.

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.

### Procedemos:

```
odoo@stani: ~/copias_de_seguridad

odoo@stani: ~/copias_de_seguridad$ crontab -1

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 ejeucatará) y guardamos cambios

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

```
doo@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 -1
# 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 /tm

# 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 Is -Iht

```
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