

Organizacion, consulta e tratamento da informacion

Índice

Sumario

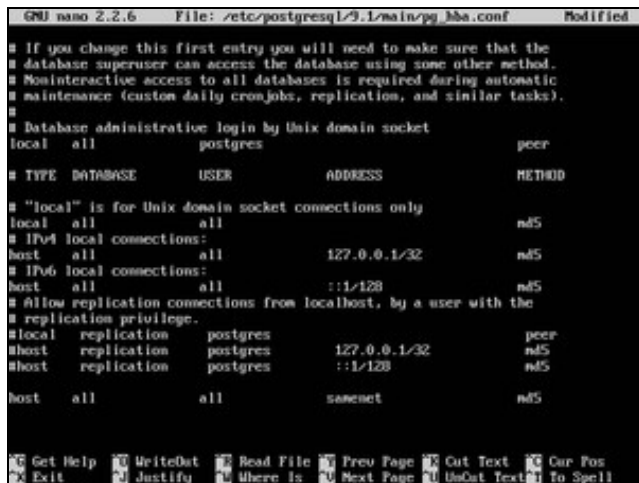
- 1 Configuración inicial
- 2 Acceso á base de datos
- 3 Deseño de consultas
- 4 Creación de vistas na BD
- 5 Creación de vistas en Odoo
- 6 Edición de informes con LibreOffice
- 7 Exportar datos en fomato CSV

Configuración inicial

Vamos acceder ao servidor Postgres do Ubuntu Server utilizando a aplicación **PgAdmin** instalada noutro equipo da subrede. Pódese permitir este acceso remoto de varias maneiras, nós vamos realizar os seguintes pasos:

- Configuramos Postgresql editando *pg_hba.conf*. No final do ficheiro inserimos unha nova liña para permitir as conexións desde outro equipo da subrede, con chave encriptada:

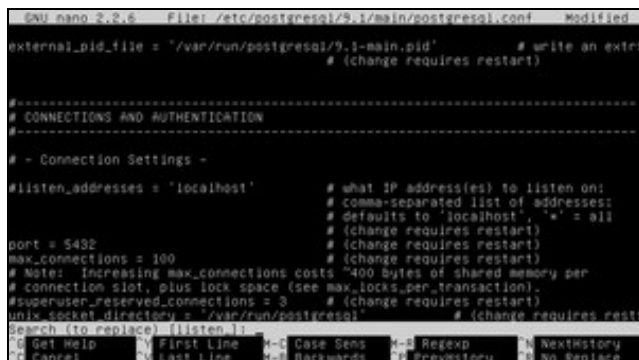
```
host    all    all    samenet    md5
```

A screenshot of a terminal window showing the nano text editor editing the file /etc/postgresql/9.1/main/pg_hba.conf. The file content includes comments about database administrative login and a table-like structure for host authentication. The last line added is 'host all all samenet md5'. The bottom of the screen shows nano editor shortcuts like 'Get Help', 'WriteOut', 'Read File', etc.

```
GNU nano 2.2.6 File: /etc/postgresql/9.1/main/pg_hba.conf Modified
# If you change this first entry you will need to make sure that the
# database superuser can access the database using some other method.
# Noninteractive access to all databases is required during automatic
# maintenance (custom daily cronjobs, replication, and similar tasks).
#
# Database administrative login by Unix domain socket
local   all             postgres                                peer
# TYPE DATABASE          USER        ADDRESS                      METHOD
# "local" is for Unix domain socket connections only
local   all             all                                md5
# IPv4 local connections:
host    all             all            127.0.0.1/32                  md5
# IPv6 local connections:
host    all             all            ::1/128                      md5
# Allow replication connections from localhost, by a user with the
# replication privilege.
#local   replication     postgres                                peer
#host    replication     postgres      127.0.0.1/32                  md5
#host    replication     postgres      ::1/128                      md5
host    all             all            samenet                      md5

^G Get Help  ^O WriteOut  ^R Read File  ^P Prev Page  ^K Cut Text   ^C Cur Pos
^X Exit      ^J Justify   ^W Where Is  ^N Next Page  ^U UnCut Text ^I To Spell
```

- Modificamos en *postgresql.conf* a liña "listen_addresses", eliminando # (símbolo de comentario) e substituíndo *localhost* por un asterisco:

A screenshot of a terminal window showing the nano text editor editing the file /etc/postgresql/9.1/main/postgresql.conf. The file content includes settings for external_pid_file, connections, and connection settings. The 'listen_addresses' line has been modified from 'localhost' to '*' (asterisk). The bottom of the screen shows nano editor shortcuts like 'Get Help', 'First Line', 'Case Sens', etc.

```
GNU nano 2.2.6 File: /etc/postgresql/9.1/main/postgresql.conf Modified
external_pid_file = '/var/run/postgresql/9.1-main.pid'           # write an extra
                                                                    # (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

#listen_addresses = 'localhost'          # what IP address(es) to listen on:
#                                          # comma-separated list of addresses:
#                                          # defaults to 'localhost', '*' = all
#                                          # (change requires restart)
#                                          # (change requires restart)
#                                          # (change requires restart)
port = 5432
max_connections = 100
# Note: Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3      # (change requires restart)
unix_socket_directory = '/var/run/postgresql' # (change requires restart)
Search (to replace) (listen):
^G Get Help  ^L First Line  ^- Case Sens  ^- Regexp     ^N NextHistory
^C Cancel    ^B Last Line  ^_ Backwards  ^P PrevHistory ^O No Replace
```

Antes

```
SAO nano 2.2.6- File: /etc/postgresql/9.1/main/postgresql.conf Modified
external_pid_file = '/var/run/postgresql/9.1-main.pid'           # write an extra
                                                                # (change requires restart)

#-----
# CONNECTIONS AND AUTHENTICATION
#-----

# - Connection Settings -

listen_addresses = '*'                                          # what IP address(es) to listen on:
                                                                # comma-separated list of addresses:
                                                                # defaults to 'localhost', '*' = all
                                                                # (change requires restart)
port = 5432                                                      # (change requires restart)
max_connections = 100                                           # (change requires restart)
# Note: Increasing max_connections costs ~400 bytes of shared memory per
# connection slot, plus lock space (see max_locks_per_transaction).
#superuser_reserved_connections = 3                             # (change requires restart)
unix_socket_directory = '/var/run/postgresql'                  # (change requires restart)

Get Help WriteOut Read File Prev Page Cut Text Cur Pos
Exit Justify Where To Next Page UnCut Text To Spell
```

Despois

- Reiniciamos PostgreSQL:

```
/etc/init.d/postgresql restart
```

- No proceso de instalación de Odoos creouse automaticamente un usuario chamado "odoo" sen chave. Agora vamos asignarlle unha chave para permitirle iniciar sesión en pgAdmin. Executamos as accións da captura de pantalla:

```
root@R-0:~# su postgres
postgres@R-0:~# psql
psql (9.1.14)
Type "help" for help.

postgres=# alter role odoo with password 'abc123.';
ALTER ROLE
postgres=# \q
postgres@R-0:~# exit
exit
root@R-0:~#
```

- Configuramos o servidor Odoo para indicarlle a nova chave do usuario "odoo":

```
nano /etc/odoo/openerp-server.conf
```

```
GNU nano 2.2.6 File: /etc/odoo/openerp-server.conf Modified
[options]
; This is the password that allows database operations:
; admin_passwd = admin
db_host = False
db_port = False
db_user = odoo
db_password = 'abc123.'
addons_path = /usr/lib/python2.7/dist-packages/openerp/addons

Get Help  WriteDat  Read File  Prev Page  Cut Text  Cur Pos
Exit      Justify   Where Is  Next Page  UnCut Text To Spell
```

- Reiniciamos o servidor Odoo co seguinte comando:

```
/etc/init.d/odoo restart
```

- Verificamos que continuamos podendo acceder a Odoo desde calquera equipo da subrede utilizando un navegador, indicando a IP do servidor e o porto 8069.

Acceso á base de datos

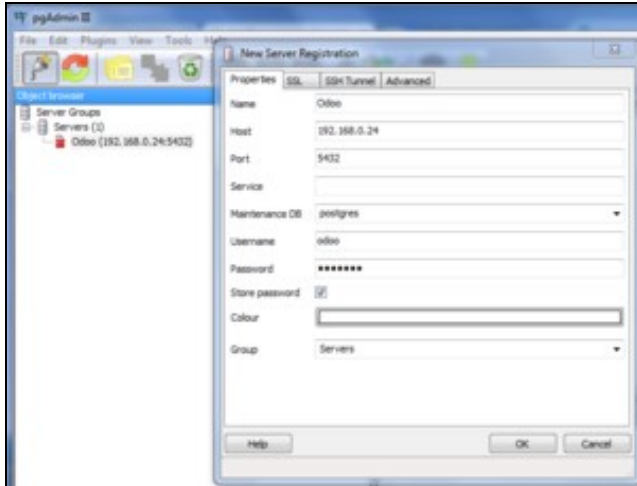
Instalamos **pgAdmin** nun equipo da subrede. Precisamos unha base de datos con moita información para realizar consultas de proba e a BD de Galaxia Software de momento non nos dá moito xogo. Vamos crear unha nova base de datos:

- Accedemos ao xestor de BDs de Odoo no enderezo *IP:8069/web/database/manager*.
- Creamos unha base de datos de probas. Marcamos a opción "Cargar datos de demostración".

The screenshot shows a web browser window with the address bar displaying `192.168.0.24:8069/web/database/manager?action=database_manager`. The page has a header with the "Galaxia SW" logo and a sidebar menu with options: "Database Management", "Create", "Duplicate", "Drop", "Backup", "Restore", and "Password". The main content area is titled "Create a New Database" and contains the following fields and options:

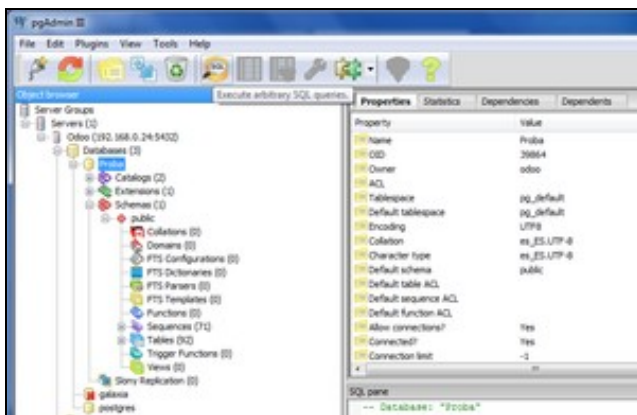
- Master password:** A text input field with four asterisks.
- Select a database name:** A text input field with the value "Prueba".
- Load demonstration data:** A checkbox labeled "Check this box to evaluate Odoo" which is checked.
- Default language:** A dropdown menu showing "Spanish / Español".
- Choose a password:** A text input field with six asterisks.
- Confirm password:** A text input field with six asterisks.
- Create Database:** A red button at the bottom of the form.

- Iniciamos pgAdmin e conectamos co servidor Postgres

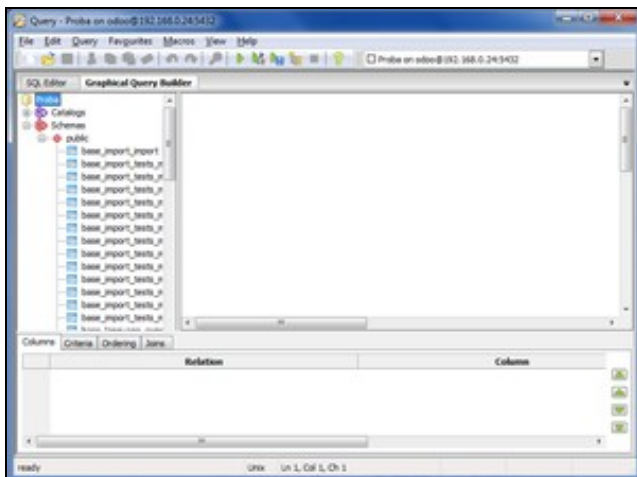


Deseño de consultas

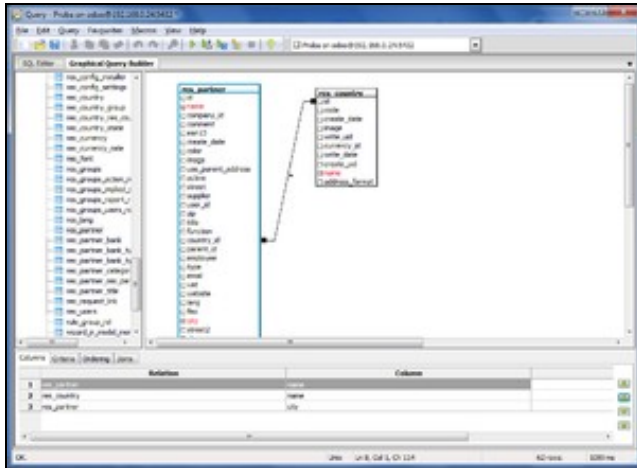
- Escollemos a BD coa que queremos traballar. Neste exemplo, "Probas". Prememos en *Executar consultas SQL arbitrarias*.



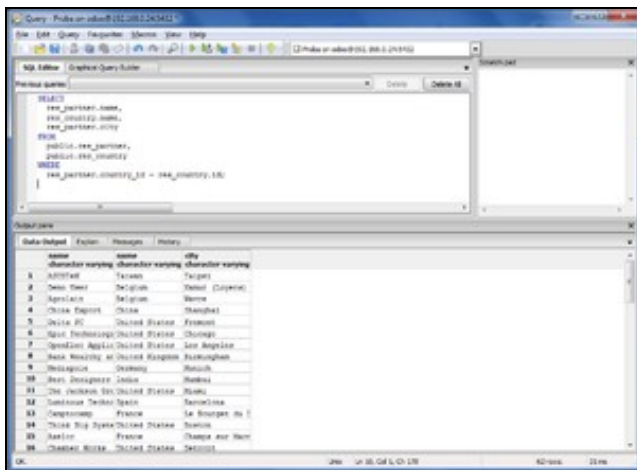
- No separador *Construtor gráfico de consultas*, escollemos *esquemas*, *public* e procuramos as táboas coas que queremos traballar.



- Por exemplo, queremos saber o nome dos socios e de que cidade e país son. Primeiro facemos clic nas táboas *res_partner* e *res_country*. Despois arrastramos a clave foránea *country_id* para relacionala co campo *id* da táboa *res_country*. Escollemos para visualizar os campos *name* e *city* e escribimos os alias correspondentes. Finalmente prememos en *Executar SQL*.



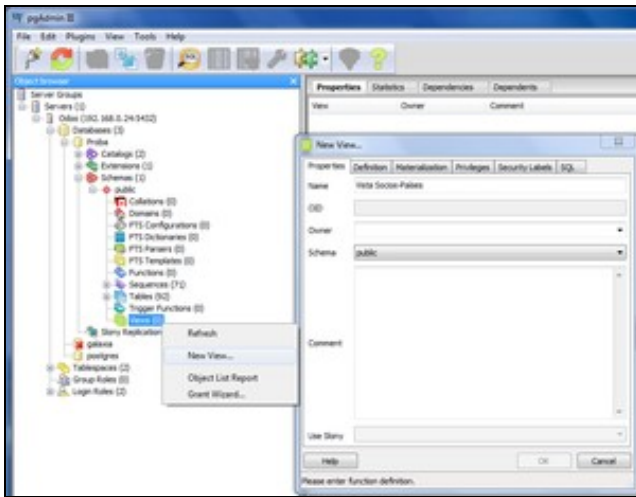
- Podemos ver o código SQL xerado e o resultado da consulta.



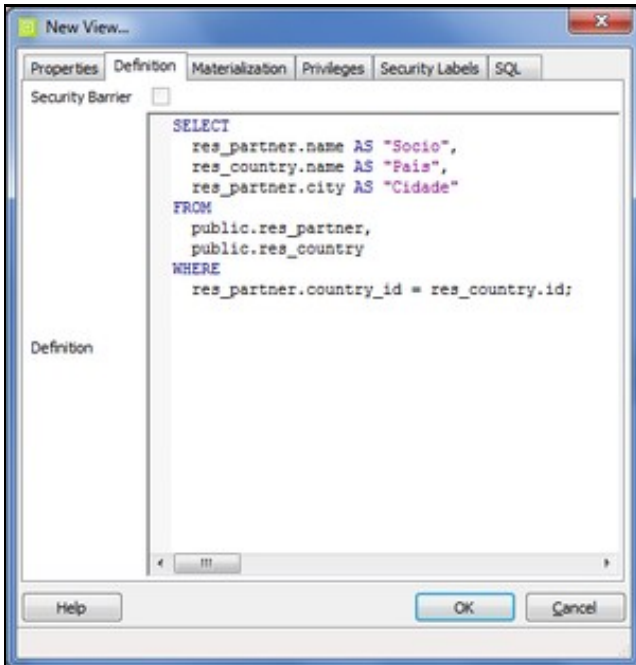
Creación de vistas na BD

Vamos crear unha vista co código SQL da consulta anterior.

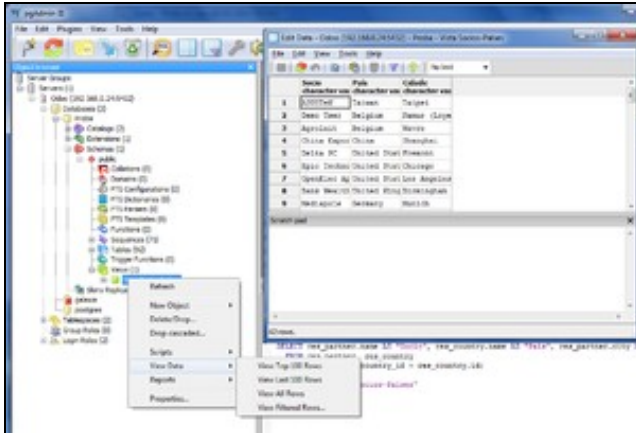
- Na pantalla inicial procuramos o apartado *Vistas* da base de datos *Probas*. Creamos unha nova vista chamada *Vista Socios-Países*.



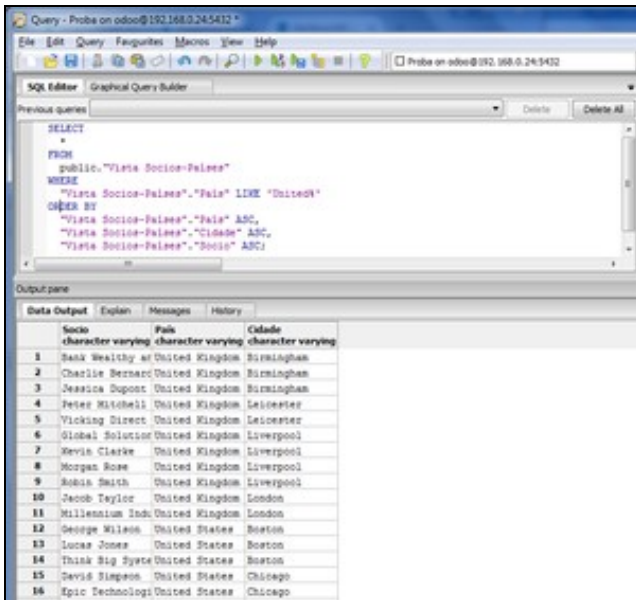
- No separador *Definição*, pegamos o código SQL da consulta.



- Agora podemos visualizar os datos da vista e traballar con ela como se se tratase dunha táboa.



- Por exemplo, vamos realizar unha segunda consulta cos datos da vista, que amose o resultado ordenado por país, despois por cidade e por nome. Podemos filtrar para que só vexamos os datos dos socios dos países que comezan por "United".



Creación de vistas en Odoo

A seguinte acción relacionada co acceso aos datos é a creación dunha vista en Odoo. É importante non confundirmos as vistas da BD (vistas no apartado anterior) coas vistas de Odoo.

En Odoo, para a empresa *Galaxia Software*, queremos crear un novo menú cunha ligazón para visualizar os datos dos socios (clientes, provedores, usuarios e a propia empresa).

Escollemos a base de datos certa e facemos login coa conta de administración.

En primeiro lugar creamos unha vista en Odoe que conecte coa táboa *res.partner* seleccionando os campos que nos interese amosar (pódense visualizar previamente con pgAdmin).

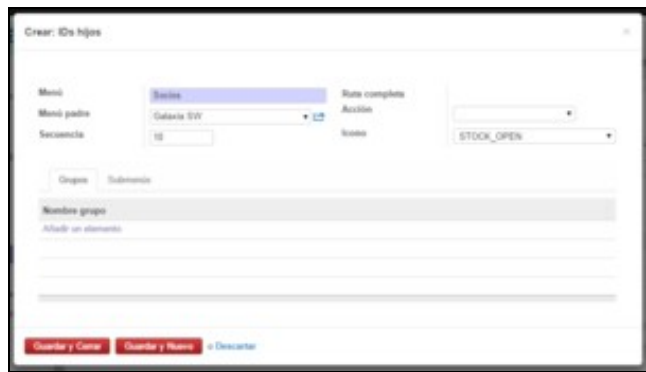
Facemos clic en *Configuración, Técnico, Interfaz de Usuario, Vistas, Crear..* Escribimos o nome do obxecto *res.partner* e o nome da vista *res.partner.vistaArbore*. Mediante código XML especificamos os campos que queremos visualizar.

Aproveitamos para duplicar a vista, editar a copia, e transformala nunha vista de formulario.

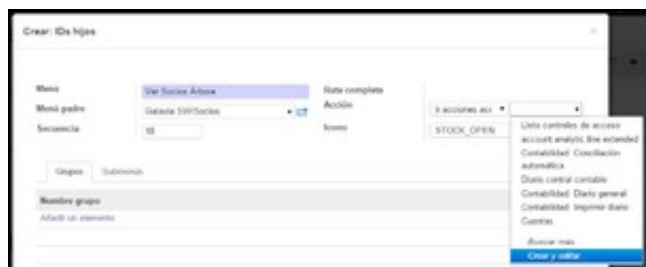
O seguinte paso é crear os elementos do menú para enlazar esta vista. En *Configuración, Técnico, Interfaz de Usuario, Elementos Menú, Crear*, creamos o menú *Galaxia Sw*.



Gardamos, editamos, e no separador *submenú* facemos clic en *añadir un elemento*. Agora vamos crear o submenú *Socios*.



Repetimos a operación con *Socios*: gardamos, editamos e clic en *añadir un elemento*, para crear unha opción do menú: *Ver Socios Árbore*. En *Acción* escollemos *ir.acciones.acc-ventana* e *Crear y editar*.



Agora vamos crear a acción *AbrirVistaSociosArbore* que se executa ao premer na opción do menú. Esta acción é a que amosa a vista previamente creada.



Podemos repetir os pasos para crear a opción de menú *Ver Socios Formulario* ligada á acción *AbrirVistaSociosFormulario*.

Actualizamos o *browser* para que apareza o novo menú e xa podemos utilizar as dúas opcións, tanto para navegar polos datos como para crear novos socios. Tamén se pode cambiar o modo de visualización no botóns da dereita.

Vista de árbore:

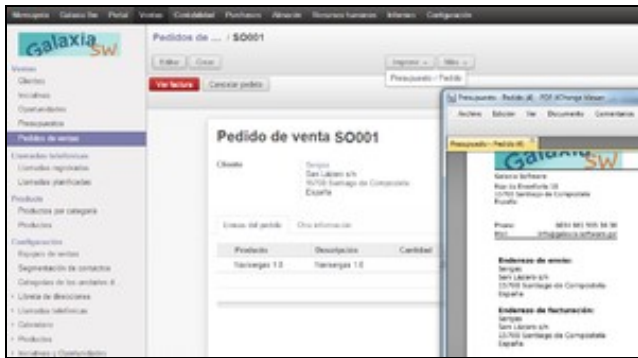
A vista de formulario ten un erro, non amosa as etiquetas dos campos:

Captura da versión 7.0 coa vista de formulario correcta:

Edición de informes con LibreOffice

É posible modificar os informes predefinidos de Odoo utilizando o Writer. Para iso, instalamos o módulo *Diseñador de informes OpenOffice*. Tamén **hai que descargar** o *plugin* en formato ZIP para Writer e seguir as instrucións de configuración: [openerp_report_designer.zip](#).

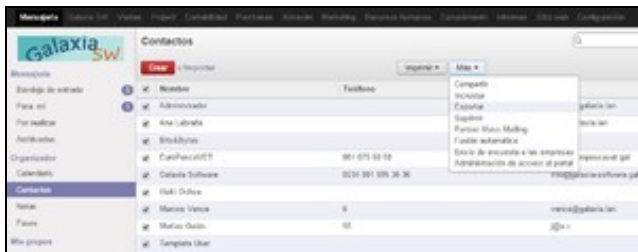
- Finalmente enviamos o informe ao servidor executando *Send to server*.
- En OpenERP, no menú *Ventas, Pedidos de ventas*, escollemos un pedido, e facemos clic en *Imprimir*.



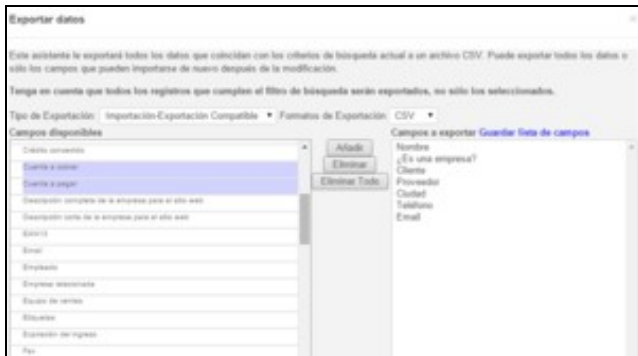
Exportar datos en fomato CSV

Os datos de calquera vista pódense exportar en formato CSV, de forma que serán facilmente procesables por outras aplicacións.

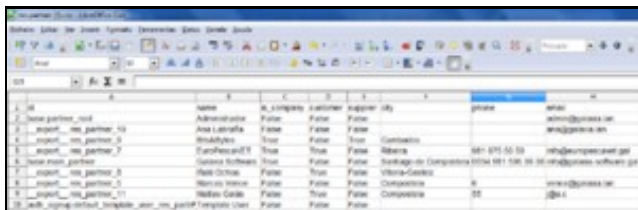
Neste exemplo seleccionamos todos os socios e facemos clic en *Más, Exportar*.



Despois seleccionamos os campos do noso interese.



O ficheiro CSV pode abrirse cun editor de texto plano, cunha aplicación de folia de cálculo, etc.



--Vence