

1 Organización, consulta e tratamento da información

Anterior: [Administración básica e utilización de sistemas ERP-CMR](#)

Seguinte: [Implantación de sistemas ERP/CRM](#)

1.1 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 OpenERP
- 6 Edición de informes con LibreOffice
- 7 Exportar datos en fomato CSV

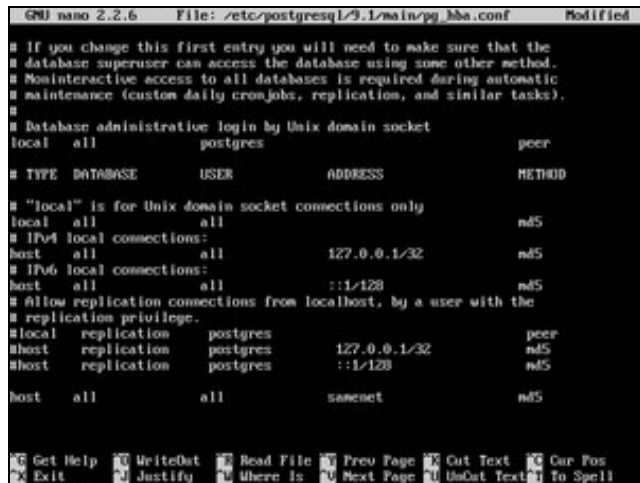
1.2 Configuración inicial

Punto de partida: [Instalación de OpenERP 7.0 en Ubuntu Server](#)

Vamos acceder ao servidor Postgres do Ubuntu Server utilizando a aplicación [PgAdmin](#) instalada en outro equipo. Para permitir este acceso remoto é preciso realizar os pasos descritos no apartado [configuración inicial de OpenERP 6.1](#) cun par de cambios:

- Configuramos Postgresql editando *pg_hba.conf*. Obrigamos a que os usuarios da BD se autenticuen con clave encriptada: Na liña que comeza por "local all all" trocamos a última palabra por "md5". No final do ficheiro inserimos unha nova liña para permitir as conexións desde outro equipo da subrede:

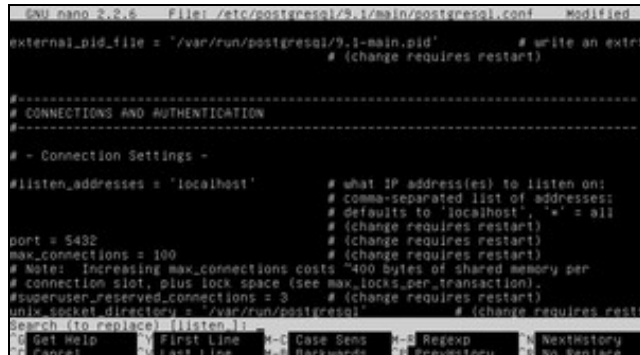
```
host    all    all    samenet    md5
```



```
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

Get Help  Ctrl+O  WriteOut  Ctrl+R  Read File  Ctrl+B  Prev Page  Ctrl+T  Cut Text  Ctrl+E  Cur Pos
Ctrl+X  Exit  Ctrl+J  Justify  Ctrl+W  Where Is  Ctrl+N  Next Page  Ctrl+U  UnCut Text  Ctrl+Q  To Spell
```

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



```
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)
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)

Search (to replace) (listen):
Ctrl+G  Get Help  Ctrl+F  First Line  Ctrl+H  Case Sens  Ctrl+R  Regexp  Ctrl+N  Next History
Ctrl+C  Cancel  Ctrl+B  Last Line  Ctrl+L  Backwards  Ctrl+P  Previous  Ctrl+Q  No Replace
```

```
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 = '*'                                          # 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
```

Depois

- Reiniciamos Postgresql:

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

- Creamos un usuario en Postgres con permiso para crear bases de datos. No noso exemplo chámase *varela*, con clave *abc123*:

```
sudo su postgres
createuser varela --pwprompt
...(Responder as preguntas: Pode crear BDs, o resto non.)
exit
```

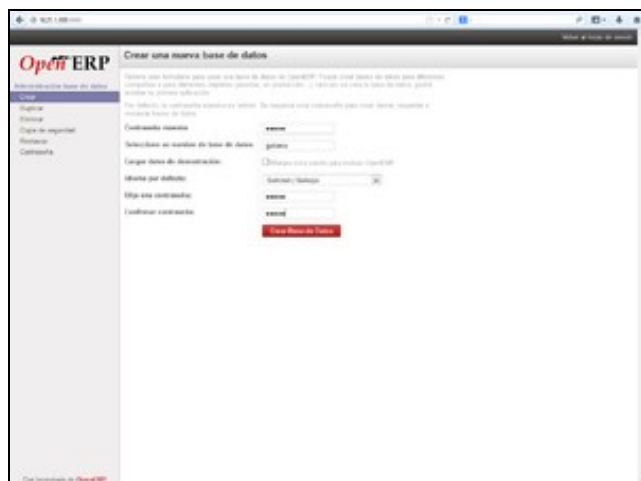
- Configuramos o servidor OpenERP para indicarlle o nome do usuario e clave creados no paso anterior:

```
sudo gedit /etc/openerp/openerp-server.conf
```

- Reiniciamos o servidor OpenERP co seguinte comando:

```
sudo /etc/init.d/openerp restart
```

- Verificar que podemos acceder desde calquera equipo da subrede utilizando un browser, indicando a IP do servidor e o porto 8069



1.3 Acceso á base de datos

Instalamos **pgAdmin** nun equipo da subrede e realizamos os seguintes pasos:

- Accedemos a Odoo e facemos clic en "Xestionar Bases de Datos".

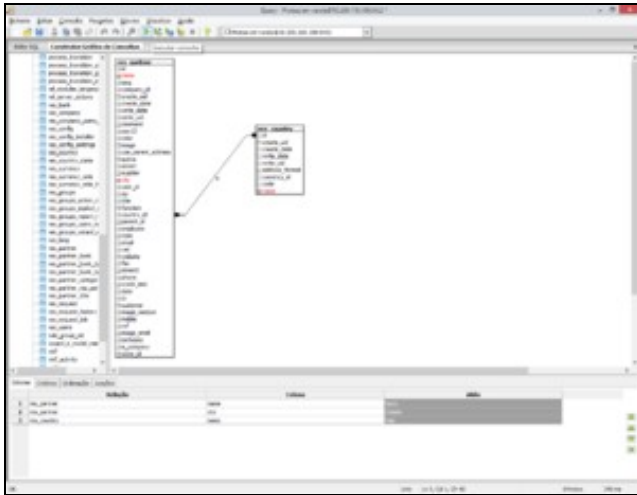


- Creamos unha base de datos de probas. Marcamos a opción "Cargar datos de demostración".

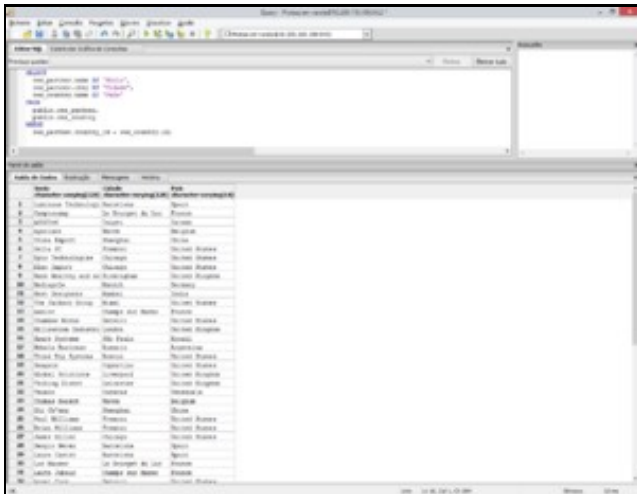


- Iniciamos pgAdmin e conectamos co servidor Postgres

- 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*.



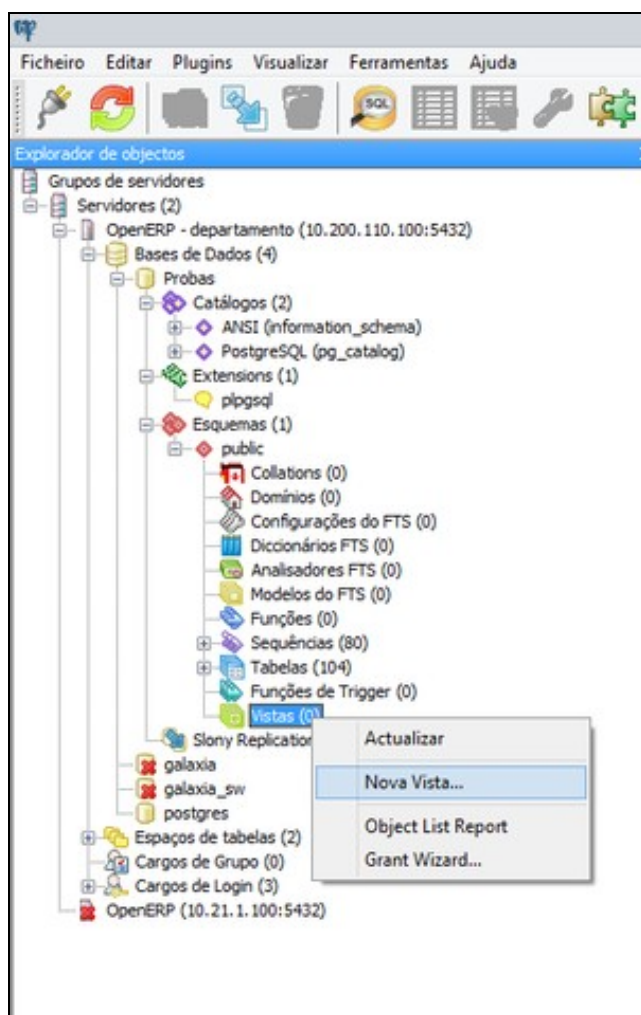
- Podemos ver o código SQL xerado e o resultado da consulta.



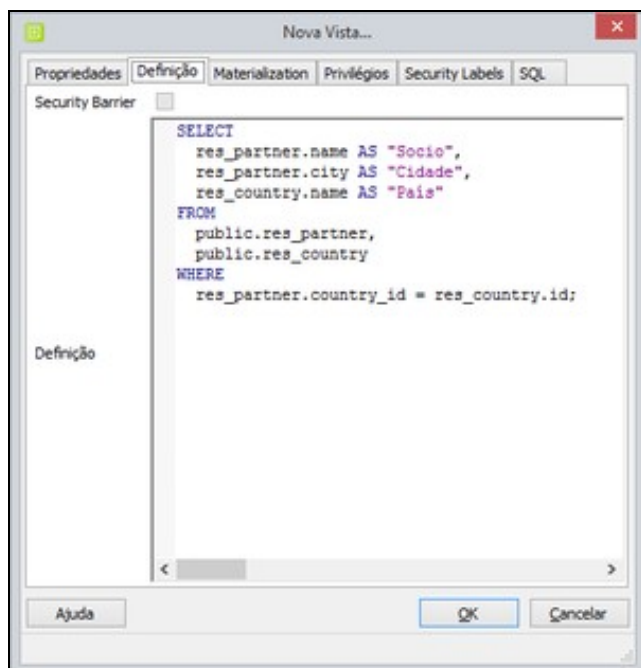
1.5 Creación de vistas na BD

Vamos crear unha vista co código SQL da consulta anterior.

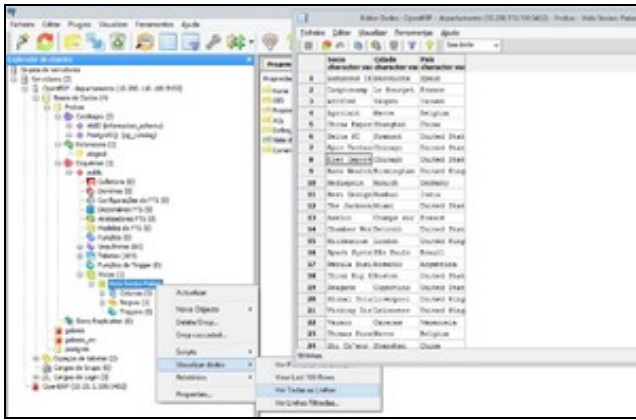
- Na pantalla inicial procuramos o apartado *Vistas* da base de datos *Probas*. Creamos unha nova vista chamada *Vista Socios-Países*.



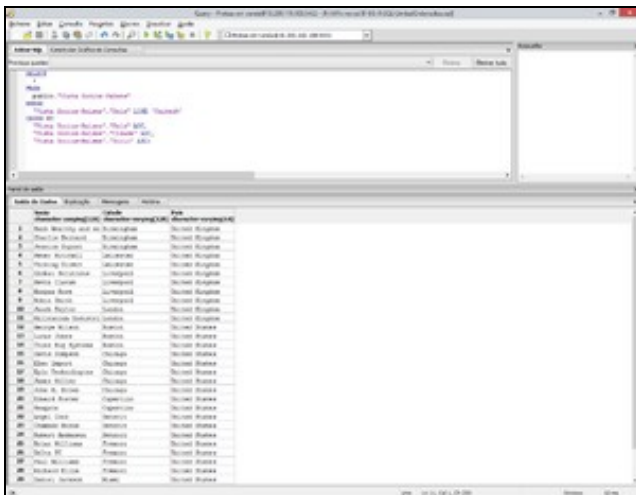
- No separador *Definição*, pegamos o código SQL da consulta.



- 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".



1.6 Creación de vistas en OpenERP

Agora que xa sabemos utilizar pgAdmin nunha BD de probas, vamos a realizar unha práctica completa, seguindo o fío da páxina [Administración básica e utilización de sistemas ERP-CMR](#).

En OpenERP, 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).

En primeiro lugar creamos unha vista en OpenERP que conecte coa táboa *res.partner* seleccionando os campos que nos interese amosar.

Coa conta de administrador/a, 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*.

Mais unha vez: gardamos, editamos, 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*.

Crear: IDs hijos

Nombre: Ver Socios Arbore

Rol completo:
 Accion:
 Boton: STOCK OPEN

Grupos:
 Submenús:
 Agregar:
 Nombre grupo:
 Guardar y Crear:
 Guardar y Nuevo:
 Descartar

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.

Crear: m2o

Nombre de acción: AbrirVistaSociosArbore

Objeto: Ver socios

Tipo de acción: AbrirVistaSocios

Ventana destino: Ventana actual

Vistas:
 Tipo de vista: Actual
 Modo de vista: Ver socios
 Rol vista: Ver socios: Ver socios
 Rol vista: Ver socios: Ver socios

Filtros:
 Valor de dominio: 0
 Valor de contenido: 0
 Límite: 0
 Auto refrescar: ☒
 Auto refrescar:
 Filtro:
 Guardar:
 Descartar

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

Crear: IDs hijos

Nombre: Ver Socios Formulario

Rol completo:
 Accion:
 Boton: STOCK OPEN

Grupos:
 Submenús:
 Agregar:
 Nombre grupo:
 Guardar y Crear:
 Guardar y Nuevo:
 Descartar

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.

Depositar Vista#117: AbrirVistaSocios

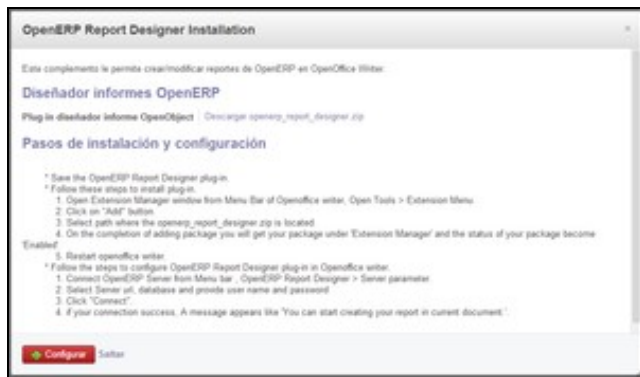
Kilobytes:
 Ciudad:
 Salomancia:
 Nombre:
 Proveedor:
 Ciudad:
 Cliente:
 Salomancia:
 Editar:
 Crear:
 Imprimir:
 Más:
 0 / 11

Nombre	Ciudad	Proveedor	Vista Formulario
Administradora			
Ara Tamarit Devesa			
Atomomiso			
emittedid=482295238aa414410d4896734135a2			
Galaxia Software	Santiago de Compostela		
Kilite	Salamanca		True
Manuel Wanda Perera			
MUCR	Viana do Castelo	True	
MUCR2	Oviedo	True	
Sergas	Santiago de Compostela		True
Template User			

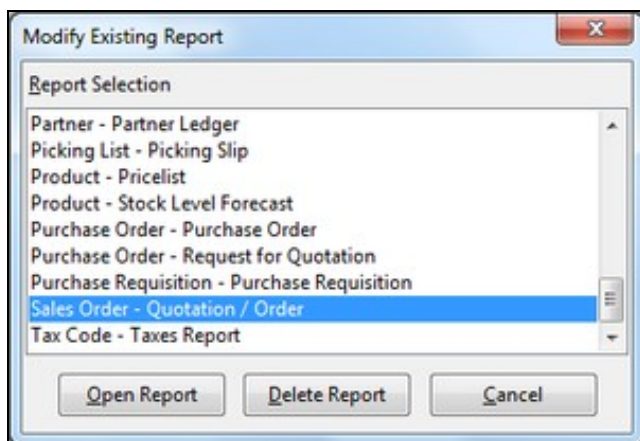
É importante non confundirmos as vistas da BD coas vistas de OpenERP.

1.7 Edición de informes con LibreOffice

Instalamos o módulo *OpenOffice Report Designer*. Tamén **hai que descargar o plugin** en formato ZIP para Writer e seguir as instrucións de configuración.



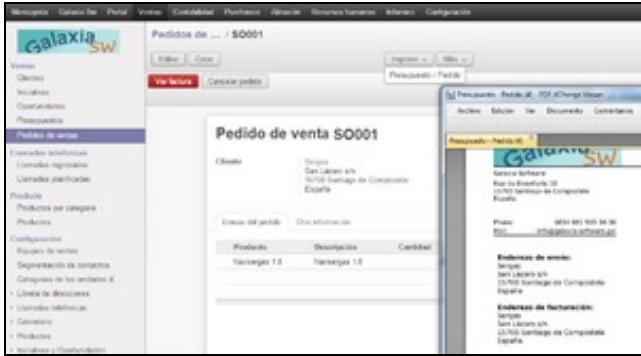
- Iniciamos Writer.
- En *Ferramentas, Xestor de extensións, Engadir*, escollemos o ficheiro *base_report_designer_installer.zip* previamente descargado. Se non ten extensión .zip, poñémoslla nós.
- Reiniciamos Writer. Aparece unha nova barra de ferramentas e un novo menú.
- Executamos *Server parameters* e conectámonos co servidor OpenERP utilizando a conta *admin*.
- Executamos *Modify Existing Report* e escollemos o informe de orde de vendas.



- Neste exemplo traducimos *Enderezo de envío* e *Enderezo de facturación*, pero pódense realizar outras accións como inserir campos.



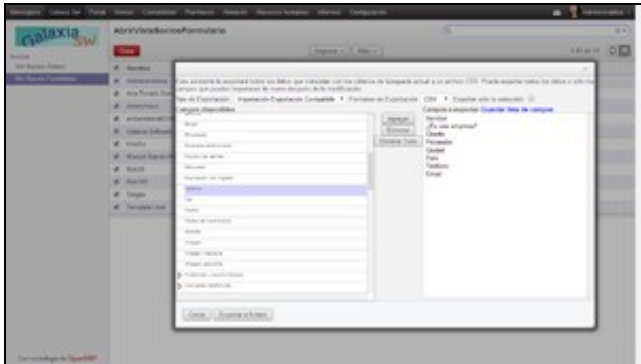
- Finalmente enviamos o informe ao servidor executando *Send to server*.
- En OpenERP, no menú *Ventas, Pedidos de ventas*, escollemos un pedido, e facemos clic en *Imprimir*.



1.8 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, facemos clic en *Más, Exportar* e seleccionamos os campos do noso interese.



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

