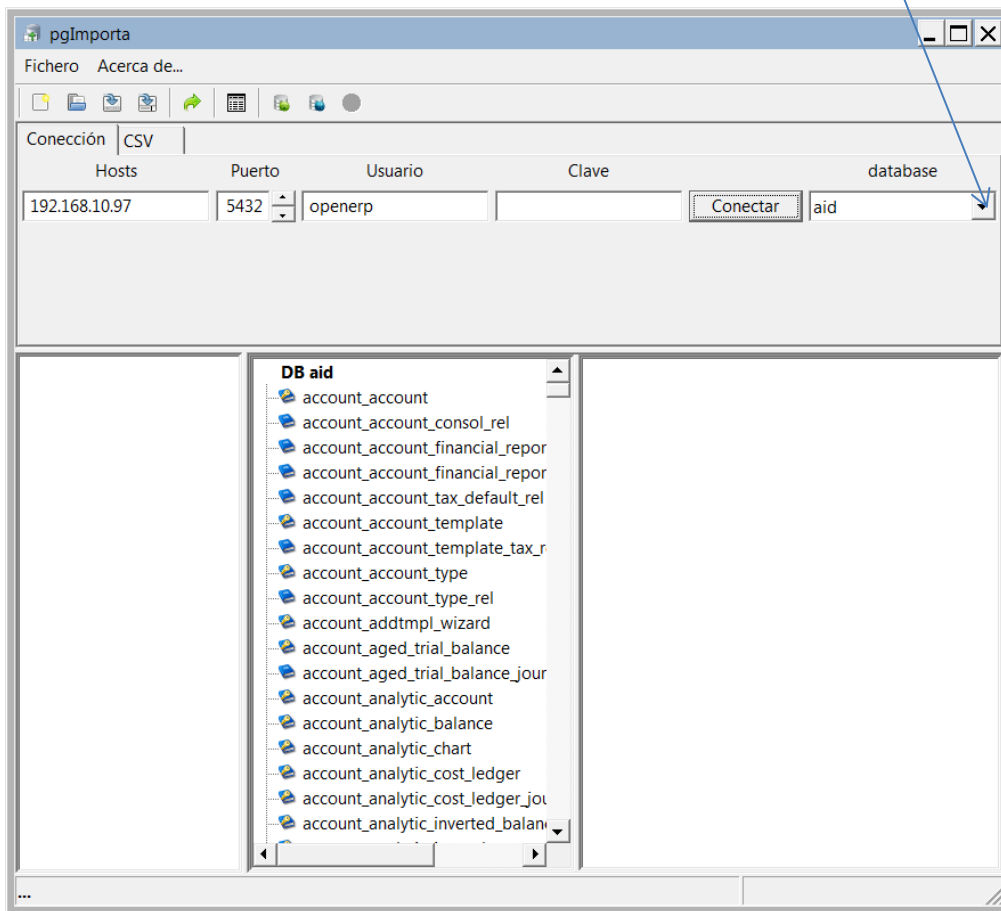
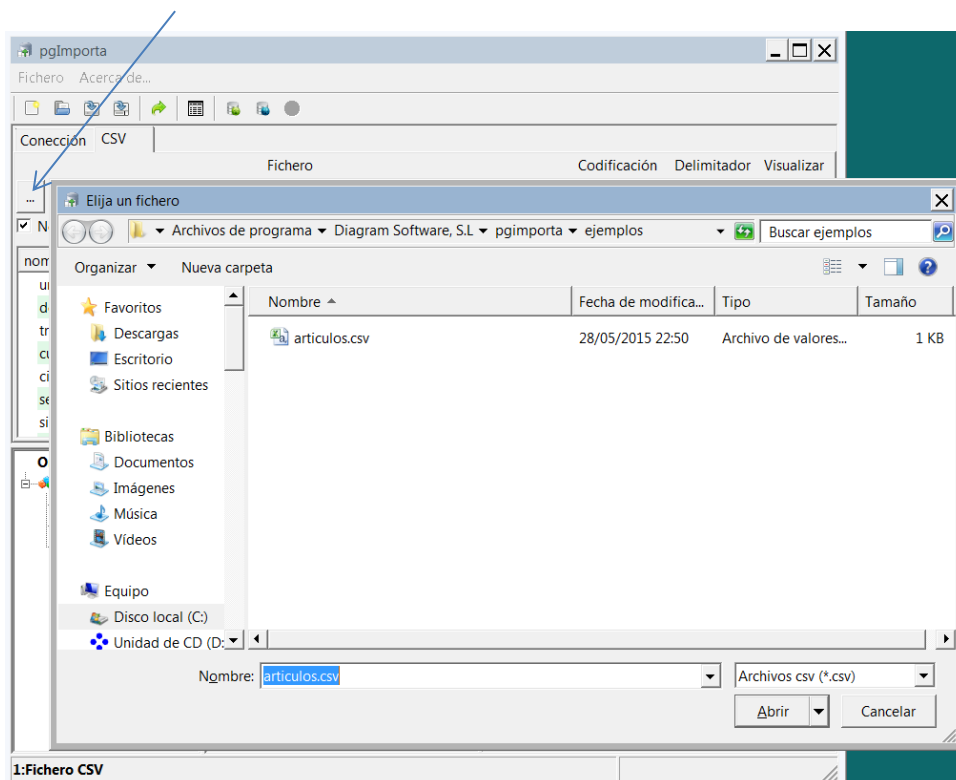


1.- Conectar con la BD de Postgres, en este caso solo es necesario el host/puerto/usuario ya que según parece no existe clave. Seguidamente se selecciona la database 'juanh'.

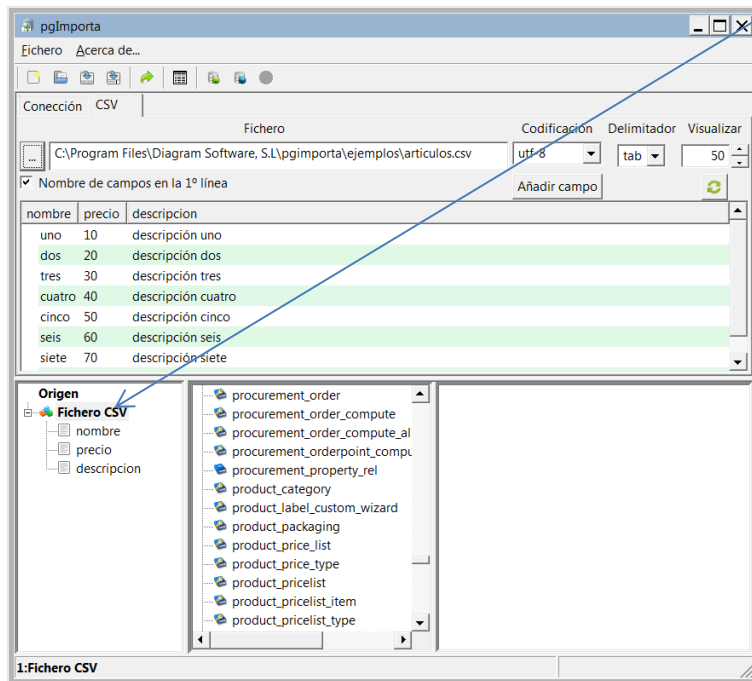


2.- Seleccionar el fichero csv 'articulos.csv', situado en el directorio de instalación\ejemplos.

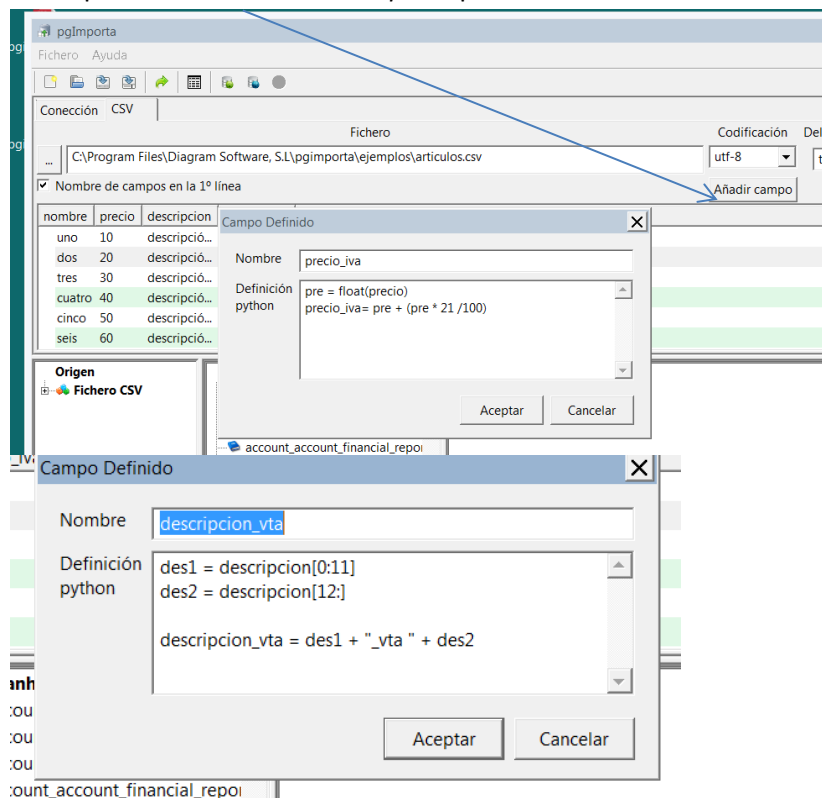


3.- Seleccionar: codificación 'utf-8', delimitador 'tab' y marcar 'Nombre de campos en la 1ª línea' (ya que este trae los nombre de los campos). Notar que los campos aparen en **Origen** en **Fichero CSV**

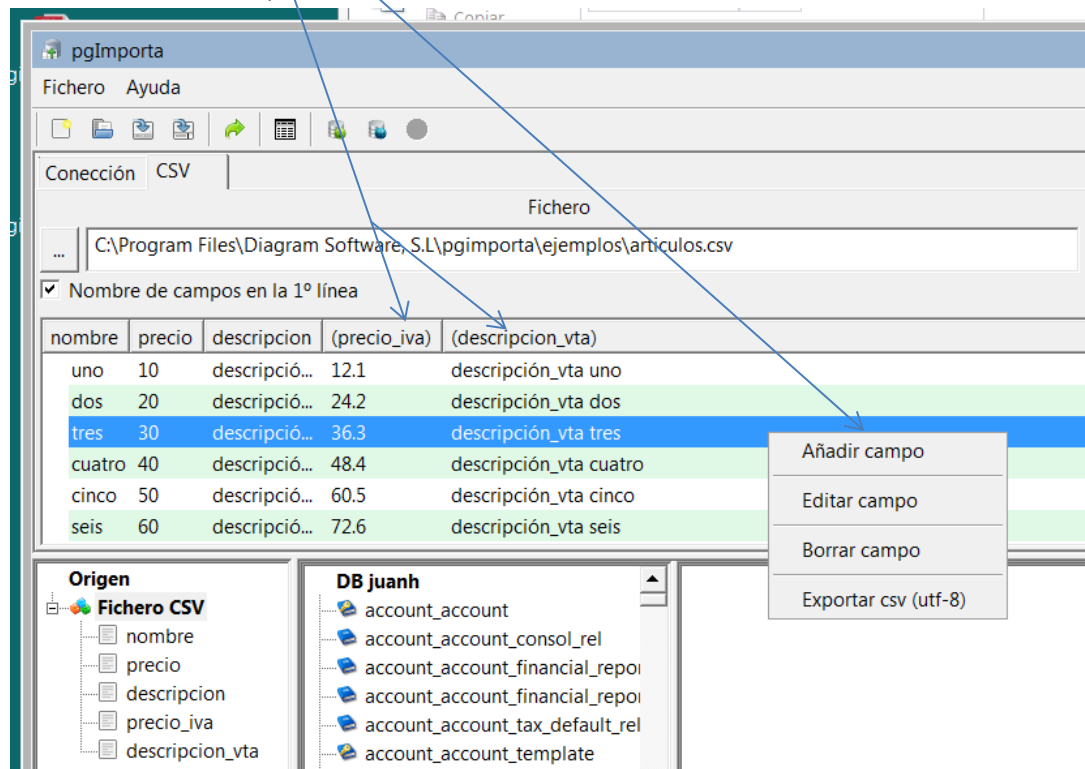
Nota: utf-8 para los creados en linux/mac, cp1252 para los de windows y cp437 los exportados desde diagram (se puede indicar cualquier otra codificación).



4.- Se pueden crear *nuevos campos* a partir de los existentes en el csv

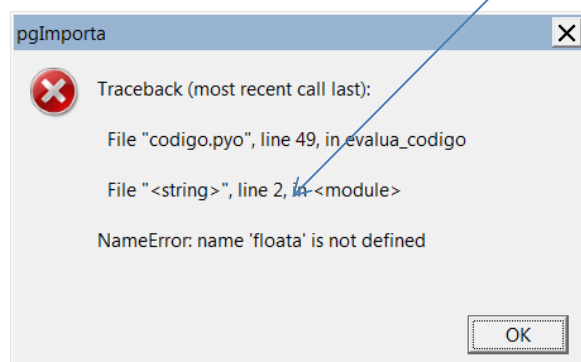
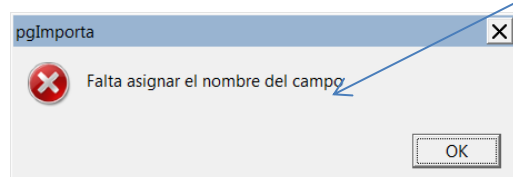


5.- Pulsando el botón derecho del ratón se puede añadir/editar/borrar y exportar, el fichero exportado contendrá además los campos definidos, estos se ponen entre parentesis para diferenciarlos de los que trae el csv.



Nota: La definición de los campos es en lenguaje python. Se puede operar con cualquier campo de csv, el valor de este siempre es un 'string'. La definición siempre debe de contener el nombre del campo asignando a este algún valor sino se obtendra un error.

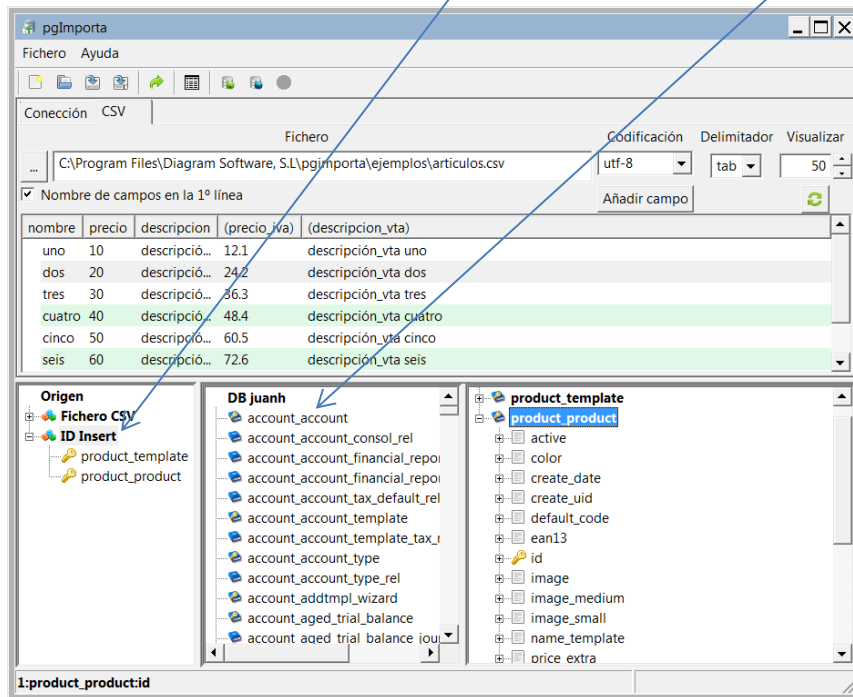
Si existe algún error en la definición tener en cuenta que el número de línea que muestra es una mayor, aquí el error estará en la línea 1.



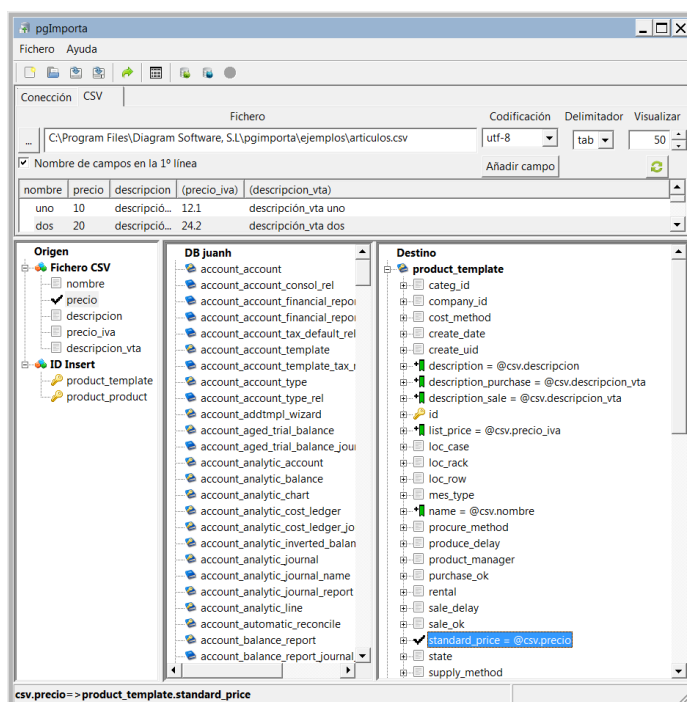
6.- Seleccionar la tabla o tablas a realizar la importación, en este caso artículos, para ello seleccionamos *product\_template* y *product\_product* haciendo sobre click en **DB juanh**.

Notar que en el **Origen** aparecen en **ID Insert** ambas tablas, estos campos contendrán los valores de los Id generados al hacer el insert o update de la tabla.

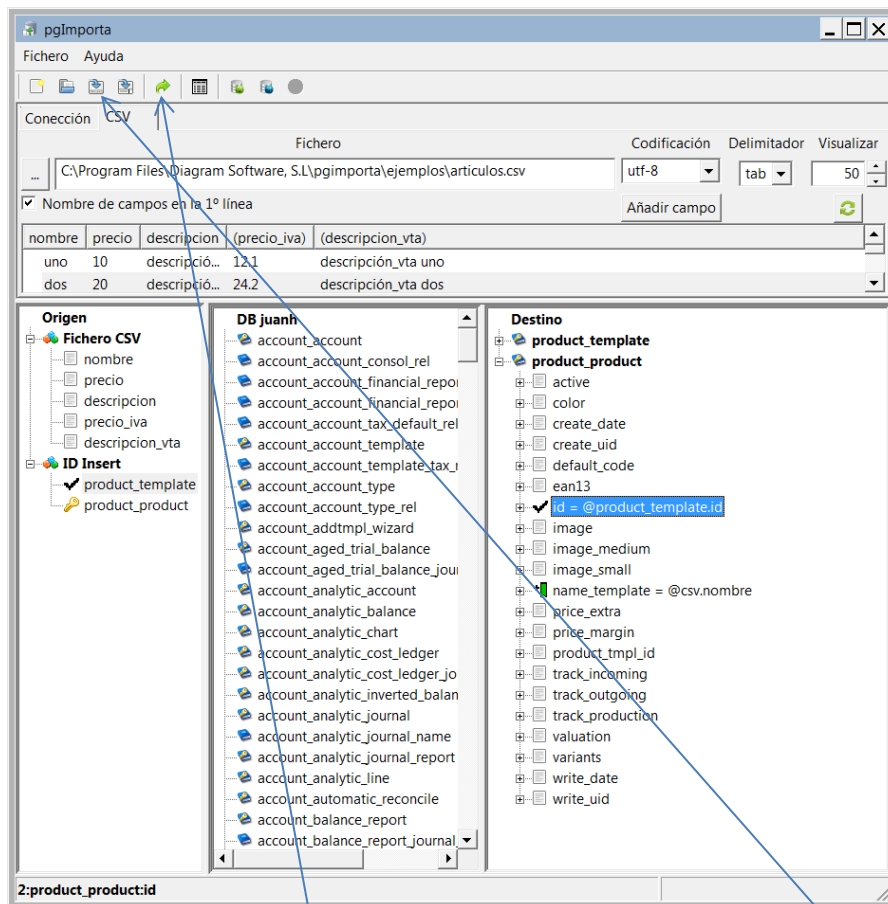
Nota: Ver apéndice para saber que tabla/s hay que seleccionar según que importar.



7.- Asignar los valores del **Origen** al **Destino**, para ello se marca el campo **Origen** y doble click en el **Destino** o al contrario. Quedará de esta forma la asignación de todos los campos del fichero CSV a *product\_template*:

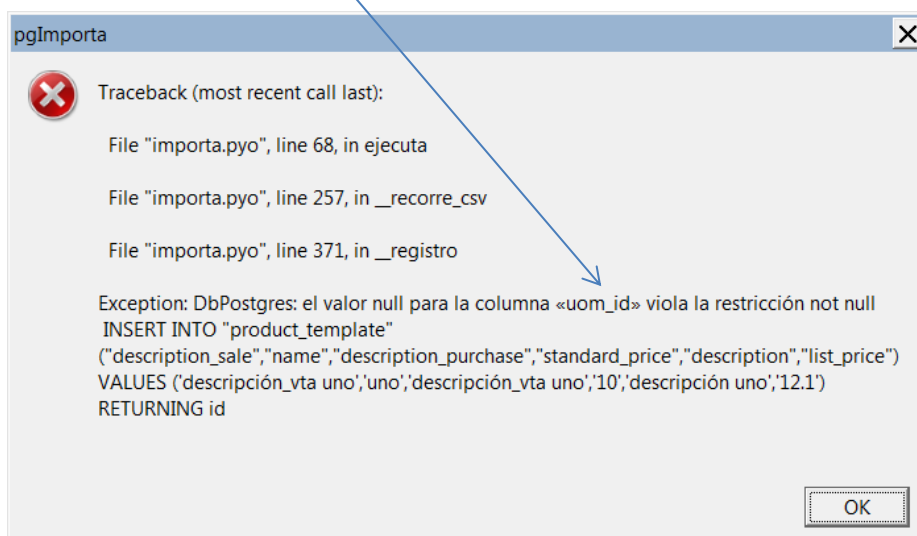


8.- Asignar los campos a *product\_product*, en este caso solo se asigna el id de *product\_template* (que resulta al hacer el insert de esta tabla) y el nombre.



Con esto hacemos la importación para ver si es posible, pero antes grabamos todo lo realizado, *por si las moscas* ☺.

Obtenemos el siguiente error, la columna "uom\_id" no puede ser null:



Se debe asignar un valor a ese campo y volver a realizar el proceso hasta que estén todos los campos que no puedan ser nulos asignados. Como no sé exactamente qué valor darle, lo que hago es crearme un artículo de muestra en el openErp

Productos / muestra

Guardar Descartar 1 / 2

Nombre del producto muestra

Categoría Todos los productos

☒ Puede ser vendido ☒ Puede ser comprado

Información Abastecimientos Inventario Ventas Contabilidad

Tipo de producto Consumible Referencia interna

Precio de venta 1,00 Código EAN13 e.g. 5901234123457

describe the product characteristics...

Y consulto los valores con el "visor sql", para ello se marca *produc\_template* y pulso en el visor (con botón izquierda y pulsamos en visor sql)

pgImporta

Fichero Ayuda

Conexión CSV

Host 192.168.10.97 Puerto 5432 Usuario openerp

Origen

- Sql Inicial
  - muestra
    - categ\_id
    - company\_id
    - cost\_method
    - create\_date
    - create\_uid
    - description
    - description\_p
    - description\_s
    - id
    - list\_price
    - loc\_case
    - loc Rack
    - loc\_row
    - mes\_type
    - name
    - procure\_meth
    - produce\_dela
    - product\_man
    - purchase\_ok
    - rental
    - sale delay

product\_template

1:product\_template:id

pgImporta (SQL)

Sql: SELECT \* FROM product\_template ORDER BY "id" LIMIT 100

purchase_ok	procure_method	supply_method	loc_case
True	make_to_stock	buy	None
True	make_to_stock	buy	None

Campos

- categ\_id = 1
- company\_id = 1
- cost\_method = standard
- create\_date = 2015-06-25 16:04:51.7657
- create\_uid = 1
- description = None
- description\_purchase = None
- description\_sale = None
- id = 7
- list\_price = 1.00
- loc\_case = None
- loc Rack = None
- loc\_row = None
- mes\_type = fixed
- name = muestra
- procure\_method = make\_to\_stock
- produce\_delay = 1.0
- product\_manager = None
- purchase\_ok = True
- rental = False
- sale\_delay = 7.0
- sale\_ok = True
- standard\_price = 0.00
- state = None
- supply\_method = buy
- type = consu
- uom\_id = 1
- uom\_po\_id = 1
- uos\_coeff = 1.000
- uos\_id = None

Pulsar en **Origen** el botón derecho y creo un **Sql Inicial**, en los valores por defecto pongo los valores de la muestra (esto nos vale para importar a otra BD que no tenga el artículo muestra), para cambiar de campo se pulsa el tabulador y enter para aceptar, también se pueden modificar los defectos desde el **Origen** (botón derecho).

categ\_id=1, cost\_method=standard, type=consu, uom\_id=1, uom\_po\_id=1

Sql Inicial

Nombre:

SQL:

loc_row	loc_rack	type	rental	uom_po...	produce...	sta
None	None	consu	None	1	None	Nc

Defecto:

Continuar: ☒ Siempre ☐ Si se cumple ☐ Si no se cumple

Aceptar Cancelar

muestra

- categ\_id
- company\_id
- cost\_method
- standard
- create\_date
- create\_user
- description
- description\_short
- id

DB juanh

- account\_account
- account\_account\_consol\_rel
- account\_account\_type\_rel
- account\_addtmpl\_wizard

cost\_method

Valor:

Aceptar Cancelar

Al igual que antes, con el botón derecho, creo el **Valor Fijo** verdadero=True

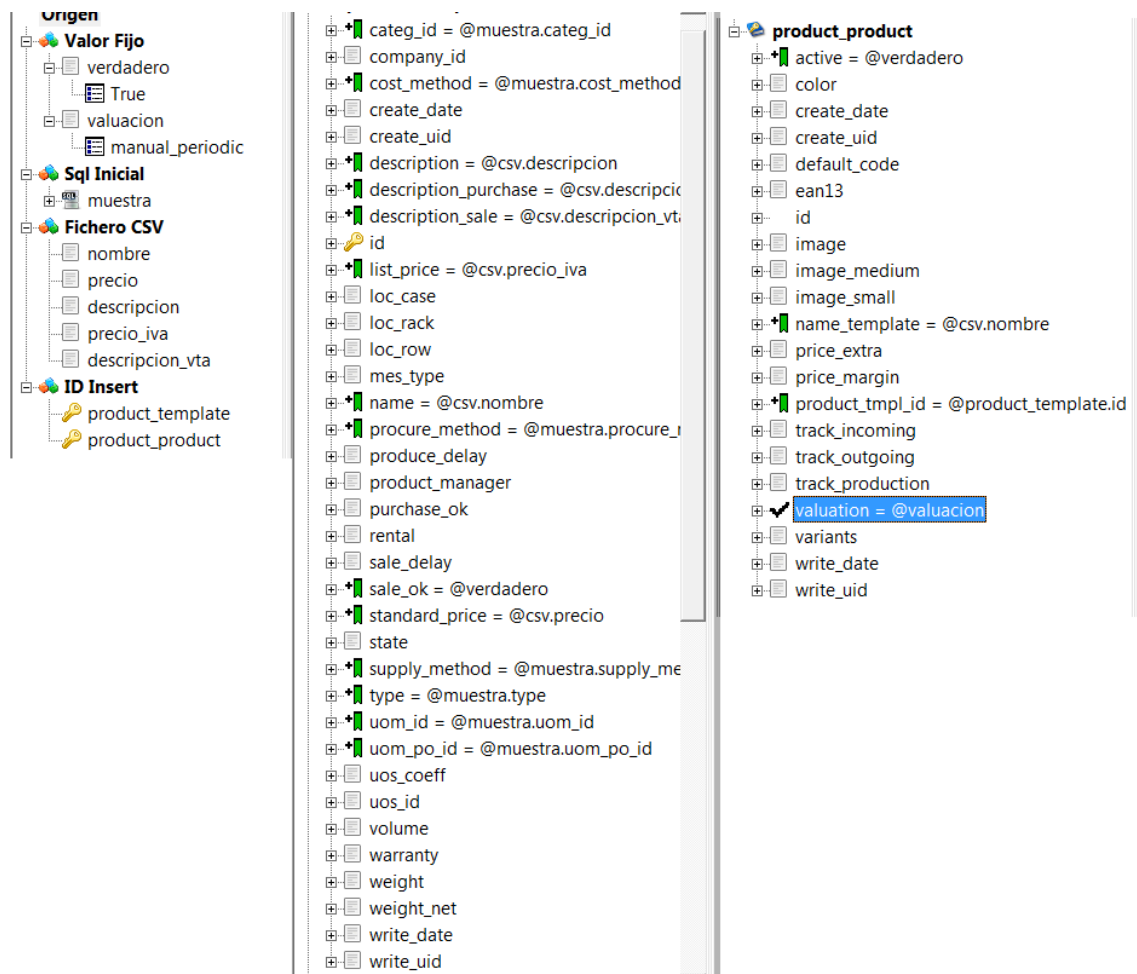
Valor Fijo

Nombre:

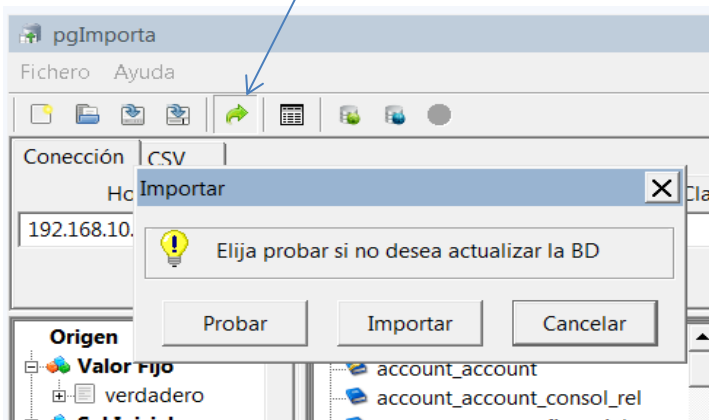
Valor:

Aceptar Cancelar

9.- Asigno el resto de campos, quedaría de la siguiente forma. En el directorio de instalación\ejemplos se encuentra “artículos.pgi” con todo lo realizado.




10.- Proceder a la importación, podemos comprobar lo realizado desde el visor sql o desde el openErp. Si se elige probar o existe algún error la BD no se actualiza.





Datos importados:



**NubeaERP**  
global cloud solutions

[MENSAJERÍA](#)
[VENTAS](#)
[CONTABILIDAD](#)
[PURCHASES](#)
[ALMACÉN](#)
[INFORMES](#)
[CONFIGURACIÓN](#)

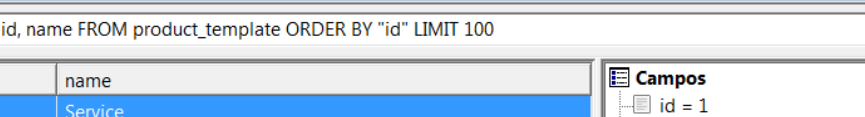
✉

---

## Productos

Crear

<div style="text-align: center;"></div> <p><b>cinco</b></p> <p>Precio: 60,50 A mano: 0,000 Unidad(es) Previsto: 0,000 Unidad(es)</p>	<div style="text-align: center;"></div> <p><b>cuatro</b></p> <p>Precio: 48,40 A mano: 0,000 Unidad(es) Previsto: 0,000 Unidad(es)</p>	<div style="text-align: center;"></div> <p><b>dos</b></p> <p>Precio: 24,20 A mano: 0,000 Unidad(es) Previsto: 0,000 Unidad(es)</p>
<div style="text-align: center;"></div> <p><b>muestra</b></p> <p>Precio: 1,00 A mano: 0,000 Unidad(es) Previsto: 0,000 Unidad(es)</p>	<div style="text-align: center;"></div> <p><b>ocho</b></p> <p>Precio: 96,80 A mano: 0,000 Unidad(es) Previsto: 0,000 Unidad(es)</p>	<div style="text-align: center;"></div> <p><b>seis</b></p> <p>Precio: 72,60 A mano: 0,000 Unidad(es) Previsto: 0,000 Unidad(es)</p>
<div style="text-align: center;"></div> <p><b>Servicio</b></p> <p>Precio: 75,00</p>	<div style="text-align: center;"></div> <p><b>siete</b></p> <p>Precio: 84,70 A mano: 0,000 Unidad(es) Previsto: 0,000 Unidad(es)</p>	<div style="text-align: center;"></div> <p><b>tres</b></p> <p>Precio: 36,30 A mano: 0,000 Unidad(es) Previsto: 0,000 Unidad(es)</p>
<div style="text-align: center;"></div> <p><b>uno</b></p> <p>Precio: 12,10 A mano: 0,000 Unidad(es) Previsto: 0,000 Unidad(es)</p>		



pgImporta (SQL)

Sql: SELECT id, name FROM product\_template ORDER BY "id" LIMIT 100

id	name
1	Service
7	muestra
20	uno
21	dos
22	tres
23	cuatro
24	cinco
25	seis
26	siete
27	ocho

**Campos**

- id = 1
- name = Service

11.- Si se vuelve a realizar la importación se duplicarán los datos, para evitarlo se crea dos nuevas **Sql por Registro**, marcando en “Continuar: si no se cumple” si no se quiere modificar el registro, “Continuar: si se cumple” y marcando “Modificar registro seleccionado en lugar de insertar nuevo” si se quiere actualizar solo los registros ya existentes o “Continuar: siempre” y marcando “Modificar registro...” si se quiere actualizar los existentes y crear los no existentes.

The dialog box titled "Sql por Registro" has a close button (X) in the top right corner. It contains the following fields and controls:

- Nombre:** A text box containing "template\_modifica".
- SQL Query:** A text box containing the query: `SELECT id FROM product_template WHERE name='@csv.nombre'`. To the left of this box is a green circular icon with a white refresh symbol.
- Defecto:** An empty text box.
- Continuar:** A group box containing three radio buttons: "Siempre" (selected), "Si se cumple", and "Si no se cumple".
- Checkboxes:** A checked checkbox labeled "Modificar registro seleccionado en lugar de insertar nuevo".
- Buttons:** "Aceptar" and "Cancelar" buttons at the bottom right.

The dialog box titled "Sql por Registro" has a close button (X) in the top right corner. It contains the following fields and controls:

- Nombre:** A text box containing "product\_modifica".
- SQL Query:** A text box containing the query: `SELECT id FROM product_product WHERE product_tmpl_id='@template_modifica.id'`. To the left of this box is a green circular icon with a white refresh symbol.
- Defecto:** A table with two columns. The first column is labeled "id" and the second column contains the value "None".
- Continuar:** A group box containing three radio buttons: "Siempre" (selected), "Si se cumple", and "Si no se cumple".
- Checkboxes:** A checked checkbox labeled "Modificar registro seleccionado en lugar de insertar nuevo".
- Buttons:** "Aceptar" and "Cancelar" buttons at the bottom right.

#### NOTAS:

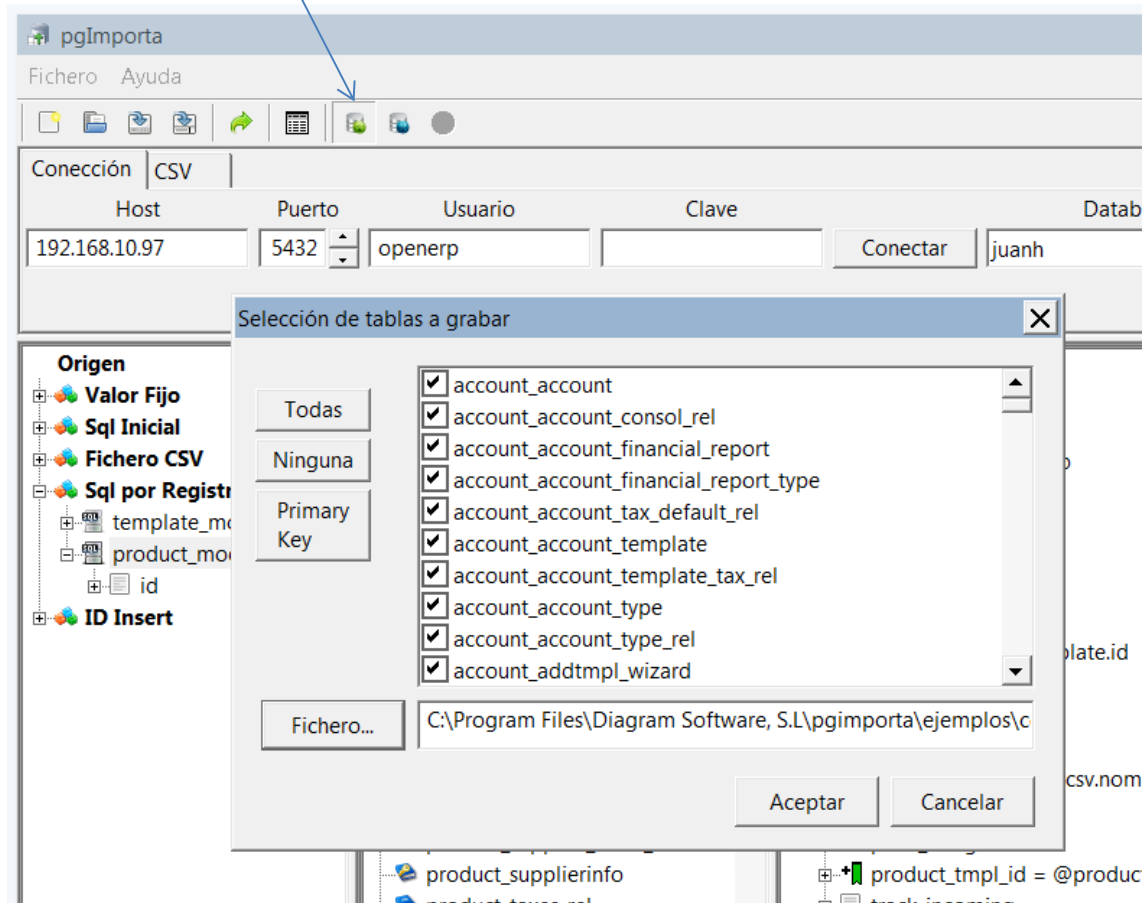
El orden del **Origen** se realiza de forma secuencial según el orden introducido. Al igual que las tablas **Destino** se actualizan también de forma secuencial.

Las **Sql Inicial** se ejecután antes de iniciar la importación, **Sql por Registro** se ejecután por cada registro del csv, en ambas se pueden indicar los campos anteriormente cargados, para ello se antepone la @. Se puede indicar cualquier sentencia sql válida.

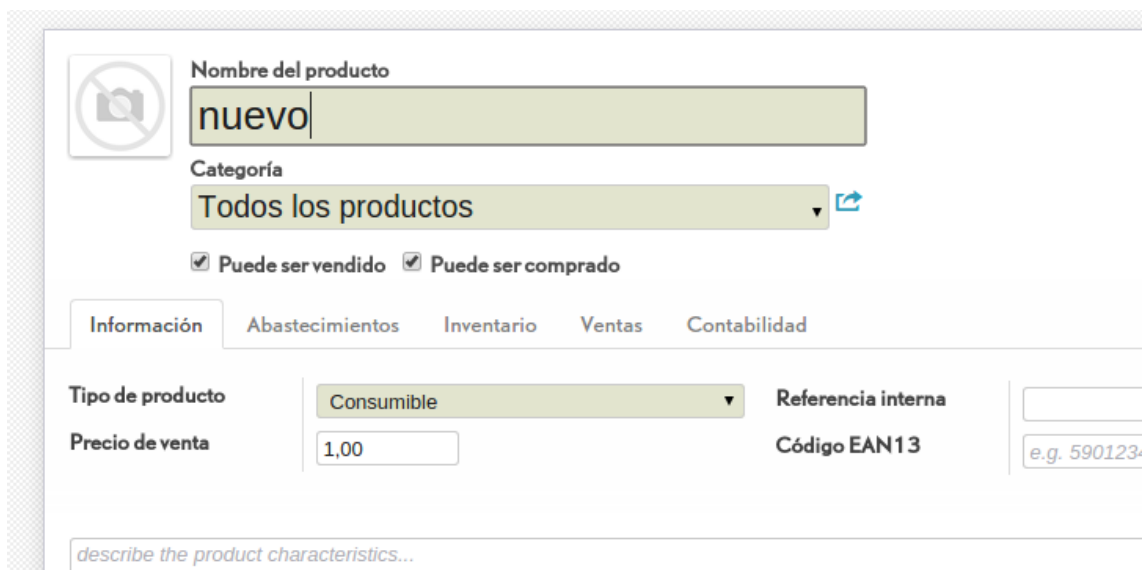
En las Sql las comillas simples " se usan para los valores y las comillas dobles "" para los campos de la tabla.

**Apéndice.** Como saber que tablas modificar si quiere actualizar los artículos.

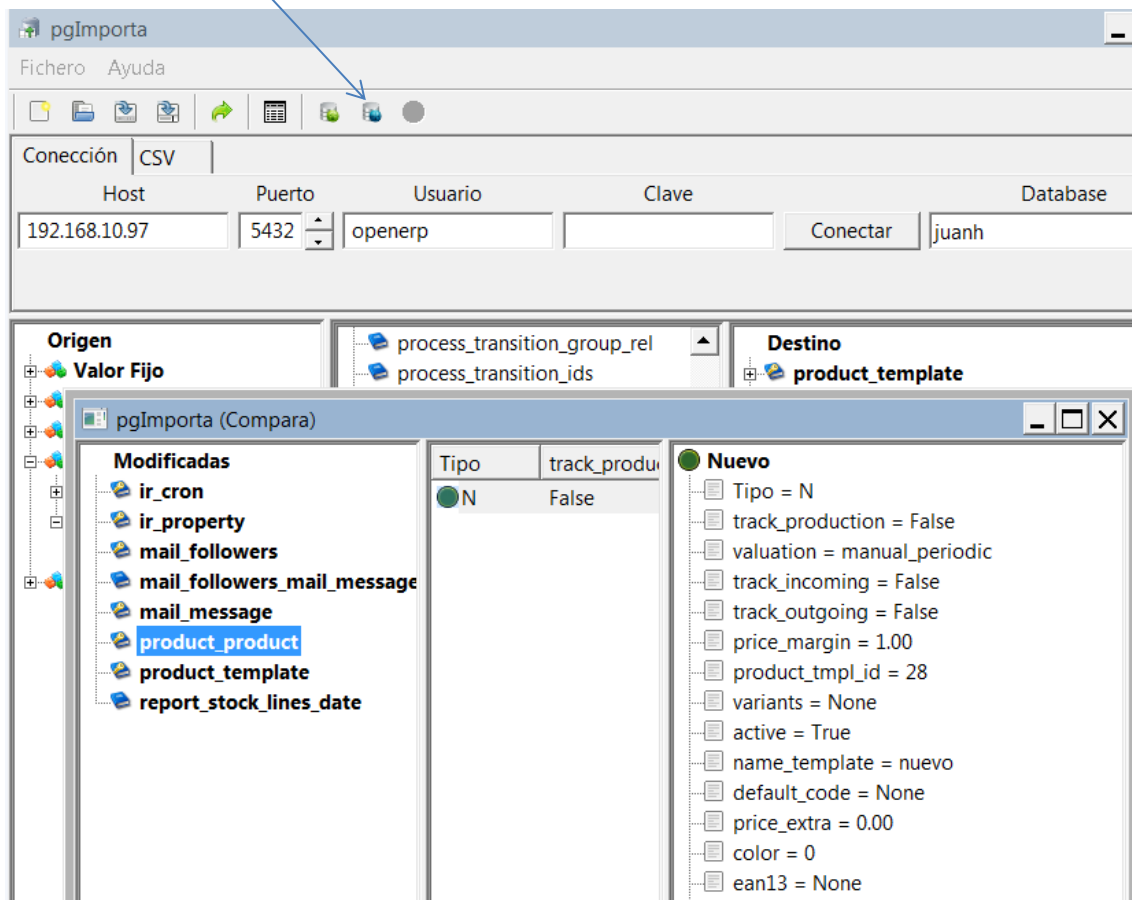
1.- Pulsar en *Grabar BD*, seleccionar *Todas* y en fichero indicar el nombre donde grabar todas las tablas seleccionadas (ojo usar una BD pequeña porque se graban las tablas completamente)



2.- Crear un artículo en el openErp



3.- Pulsar en *Comparar DB*, mostrará los cambios que se ha producido en las tablas. Analizar los cambios producidos para saber qué hacer ☺ .



4.- Se puede ejecutar desde la línea de comandos:

C:\> pgimporta -h

Esto crea el 'pgimporta.log' indicando los parámetros a ejecutar:

```

INFO:  -f fichero.pgi
INFO:  -c datos fichero csv: fichero,nombres1linea(1=si,0=no),encode,delimitador
INFO:  -d datos database: host,database,puerto,usuario,clave
INFO:  -p solo probar (no actualiza la BD)
INFO:  -l fichero log
INFO:  -t tipo log (0 defecto / 1 detallado)
INFO:  -h Muestra ayuda

```

**That's all folks.**