

Tema 3

Organización y consulta de la información en un sistema ERP-CRM

2º DAM

SISTEMAS DE GESTIÓN EMPRESARIAL

Montse Gómez

montse.gomezmoya@gmail.com

Contenidos

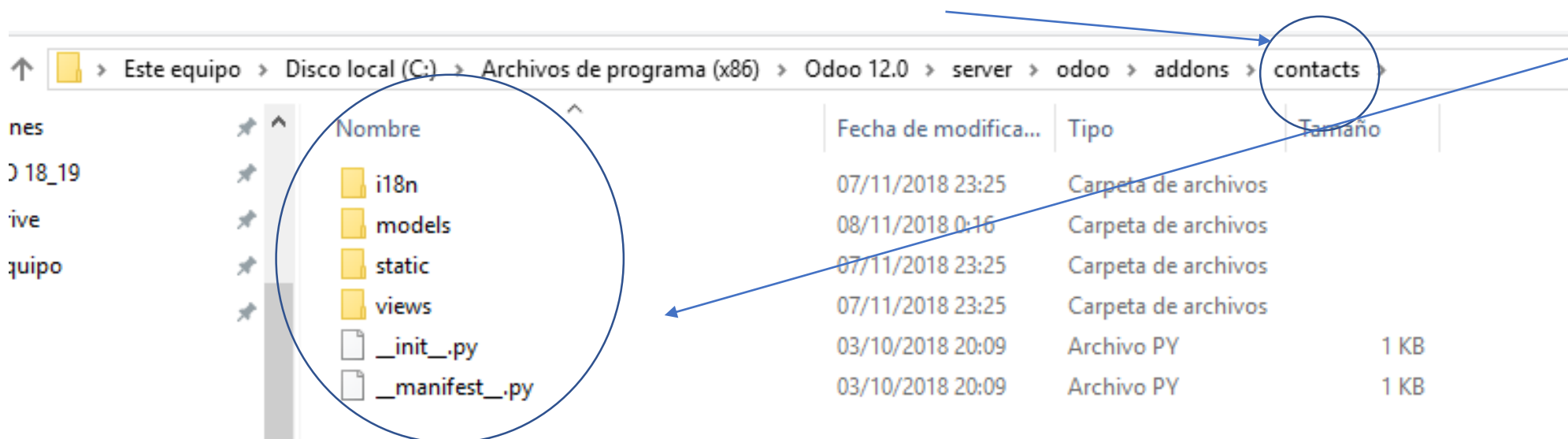
1. Organización y consulta de la información
2. Operaciones básicas con la BD. PGAdmin III
3. Interfaces de entrada de datos y de procesos. Formularios (forms)
4. Menús
5. Informes y listados (Reports)
6. Tratamiento de la información
7. Herramientas de monitorización y de evaluación del rendimiento

1. Organización y consulta de la información

- La base de datos de un sistema ERP es de gran tamaño y complejidad.
- Almacena las tablas con los datos de la aplicación, diferentes tipos de vistas, diferentes tipos de informes, las funciones que realizan operaciones sobre los datos Etc.
- Esto requiere establecer una normativa para organizar la información que los desarrolladores deben seguir a la hora de modificar el código fuente o el esquema de la base de datos. Por ejemplo, incluir un prefijo a los componentes de la base de datos para saber a que módulo pertenece.
- En los ERP escritos en lenguajes orientados a objetos, cualquier dato es accesible a través de objetos. Por ejemplo en Odoo tenemos el objeto `res.partner` para acceder a los datos de los clientes y proveedores, el objeto `account.move` (`account.invoice` en versiones anteriores) para los datos de las facturas, ... etc

Estructura de un módulo en odoo

Los módulos de odoo, ya sean oficiales o externos, están programados en Python y se empaquetan en una carpeta con el nombre del módulo que contiene una estructura similar a esta:



Los módulos tienen sus reglas, primero debe de estar en una carpeta y ésta dentro de la carpeta ADDONS, siempre deben de existir en esa carpeta dos archivos

__init__.py aquí están los imports del módulo

__manifest__.py aquí está descripción, nombre, version, autor, dependencias, archivos XML que usaras entre otras cosas.

En las carpetas tenemos otro archivo(s) (.py) con los modelos, vistas, aplicación etc

El módulo BASE

- Este módulo es el corazón de Odoo ya que en el están definidas los modelos básicos como los contactos, los usuarios, los grupos de usuarios, la compañía, los bancos, la clase de configuraciones, la moneda, los idiomas ... etc Por otra parte, también define las secuencias, las acciones y los tipos de campos. Es donde se encuentran todos los elementos que usamos luego para definir nuestros modelos y las interfaces, es en si el framework de Odoo,
 - **res_partner**: la clase donde se encuentra la definición de los contactos como clientes y proveedores.
 - **res_users**: la clase de los usuarios de Odoo, es donde se guardan datos como login, contraseñas, permisos...
 - **res_company**: la clase de la compañía y en clase multi_company_default se definen las opciones necesarias para trabajar con multicompañías.
 -
- Entre los modelos descritos hay que tener cuidado al modificar

Modelos, Tablas y Vistas de la base de datos

- Un **modelo (objeto)** guarda todos los elementos que forman parte de la aplicación (tablas, disparadores o acciones sobre esas tablas, formularios, informes, ventanas, asistentes ...)
- Dentro de los modelos distinguimos las tablas. **Una tabla es una estructura de datos organizada en filas y columnas, de manera que cada columna es un campo (o atributo) y cada fila un registro.**
 - Por ejemplo, en SQL en la tabla res.users cada registro se corresponde con un usuario, y para cada usuario se guardan una serie de atributos como nombre, login o password.
- En Odoo se accede a los modelos a través de la opción **Menu Técnico/Estructura de la base de datos/MODELOS**. En esta opción podemos encontrar toda la información que maneja la aplicación.
- En ocasiones, la base de datos está formada por tantas tablas y objetos que se vuelve compleja y difícil de manejar. En esos casos, interesa que algunos grupos o perfiles de usuarios tengan una vista parcial de esos datos. Para estos casos se utilizan las vistas.
- **Una vista es básicamente una "tabla virtual" a la que se puede acceder como si fuera una tabla del esquema, pero que realmente no lo es.** Tienen la misma estructura que las tablas: filas y columnas o campos, y se puede acceder a ellas de la misma forma, a través de consultas de acceso a datos como veremos posteriormente.
- Para administrar la base de datos podemos conectarnos directamente al gestor de la base de datos, aunque lo normal es utilizar herramientas gráficas que nos faciliten el trabajo. Entre las **herramientas para administración de bases de datos PostgreSQL se encuentra pgAdmin III.**

Definición de campos

- Los OBJETOS (MODELOS) están definidos en archivos XML.
- Un objeto está formado por CAMPOS.
- Los campos son la información que se muestra en pantalla a través de las VISTAS.

Por ejemplo, una vista de tipo formulario con 2 campos nombre y título la definiríamos así

```
<?xml version="1.0"?>
<form string="Empresa">
    <field name="name"/>
    <field name="title"/>
</form>
```

En ejemplo anterior, hemos utilizado distintos tipos de etiquetas para describir la vista :

- La **etiqueta form** indica que vamos a describir el **un formulario**
- Dentro de ella se utiliza la **etiqueta field** para cada uno de los campos que lo componen.
- Las **etiquetas empiezan con el símbolo < y terminan con el símbolo />.**
- En las vistas de tipo formulario, los campos van precedidos por una etiqueta con su nombre y se colocan de izquierda a derecha, en el orden con que son declarados en el archivo XML .

Tipos de campos.

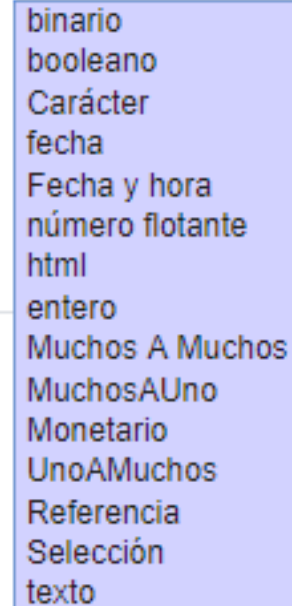
Tipo de campo

Campo ayuda

Los **objetos en Odoo contienen campos** los cuales permiten introducir datos en la base de datos, estos campos van definidos en el **atributo columns**.

CAMPOS BÁSICOS:

- **boolean:** Boolean (True o False)
 - **Sintaxis:** `fields.boolean('Field name', [, Optional Parameters])`
- **integer:** Numero entero
 - **Sintaxis:** `fields.integer('Field Name' [, Optional Parameters]),`
- **float:** Un numero de punto flotante
 - **Sintaxis:** `fields.float('Field Name' [, Optional Parameters]),`
- **char:** Un string de espacio limitado, su longitud es un parámetro requerido
 - **Sintaxis:** `fields.char('Field Name', size=n [, Optional Parameters])`
- **text:** Un campo de texto sin limite
 - **Sintaxis:** `fields.text('Field Name' [, Optional Parameters]),`
- **date:** Una fecha
 - **Sintaxis:** `fields.date('Field Name' [, Optional Parameters]),`
- **datetime:** Una fecha y una hora
 - **Sintaxis:** `fields.datetime('Field Name' [, Optional Parameters]),`
- **binary:** Dato binario
- **selection:** Esta campo permite seleccionar un dato de varios valores
 - **Sintaxis:** `fields.selection(((('n','Unconfirmed'), ('c','Confirmed')), 'field Name' [, Optional Parameters])`

A screenshot of the Odoo field type dropdown menu. The menu is light blue with a list of field types in black text. A mouse cursor is visible at the top right corner of the dropdown. The field types listed are: binario, booleano, Carácter, fecha, Fecha y hora, número flotante, html, entero, Muchos A Muchos, MuchosAUno, Monetario, UnoAMuchos, Referencia, Selección, and texto.

binario
booleano
Carácter
fecha
Fecha y hora
número flotante
html
entero
Muchos A Muchos
MuchosAUno
Monetario
UnoAMuchos
Referencia
Selección
texto

Campos relacionales

Como en cualquier base de datos, podemos establecer tres tipos de relaciones entre tablas. Estos son :

- **One2many**
- **Many2one**
- **Many2many**

Campos one2many

- **one2many:** Este campo expresa una relación uno a muchos entre dos objetos,

Sintaxis:

fields.one2many('other.object.name', 'Field relation id', 'Fieldname', optional parameter)

Optional parameters:

- *invisible: True/False*
- *readonly: True/False*

....

Ejemplo

En una relación Tabla1 (1 → N) Tabla2

Se define en la tabla principal Tabla1 del tipo:

fields.one2many('modelo de la T2', 'Campo relacionado en T2', optional parameter)

Campos many2one (muchos a uno)

- **many2one:** Asocia este objeto con un objeto padre en una relación muchos a uno, por ejemplo, de muchas marcas de coches existe una que pertenece a un vehículo en particular

Sintaxis:

fields.many2one('other.object.name', optional parameter)

Optional parameters:

- *ondelete*: ¿Qué debería suceder cuando se elimina el recurso al que apunta este campo?.
- + *Predefined value*: "cascade", "set null", "restrict", "no action", "set default"
- + *Default value*: "set null"
- *required*: True
- *readonly*: True
- *select*: True - (crea un índice en el campo de clave externa Foreign Key)

Ejemplo

En una relación Tabla1 (1 → N) Tabla2

Se define en la tabla secundaria Tabla2 del tipo:

fields.one2many('modelo de la T1', optional parameter)

Campos many2many: (muchos a muchos)

Sintaxis:

fields.many2many('other.object.name', 'relationobject', 'actual.object.id', 'other.object.id', 'Field Name')

Donde:

- other.object.name es el otro objeto que pertenece a la relación
- relation object objeto relación es la tabla que hace el enlace
- actual.object.id and other.object. son los nombres de los campos utilizados en la tabla de relaciones

▪ Ejemplo:

```
'category_ids':fields.many2many(
'res.partner.category',
'res_partner_category_rel',
'partner_id',
'category_id',
'Categories')
```

Interface en Odoo: Vistas

Los archivos XML definen la interface del ERP, lo que en Odoo se denominan “vistas”, Las vistas definen la forma en que se muestran los registros de un modelo. Cada tipo de vista representa un modo de visualización.

En lo referido a la interface tenemos :

Menu los menús que usan los actions para enlazar estas opciones al menu de OE, :

Action las actividades (action) que determina que tipos de vistas tiene una clase, el orden en que se deben mostrar ...

Search definición de las búsquedas (search) en donde se definen los campos por lo que se puede buscar y/o agrupar los datos en las vistas tree y kanban,

Vistas (list,form,etc.)

Estas vistas pueden ser de varios tipos:

- **list=listado, (tree en versiones anteriores)**
- **form=formularios,**
- **kanban=vista de íconos,**
- **calendar=vista de calendario o agenda,**
- **gant=vista tipo diagrama de Gant**
- **graph= vista en forma de gráfico estadístico**
- **Activity =actividad (nuevo tipo)**
- ...

Nombre del modelo y tipo de vista

- En odoo, cuando accedemos a un objeto(modelo) y los mostramos en un determinado modo de vista. Desde la URL podemos saber el nombre del modelo en (model=.....) y el tipo de vista que está mostrando (view_type=.....)

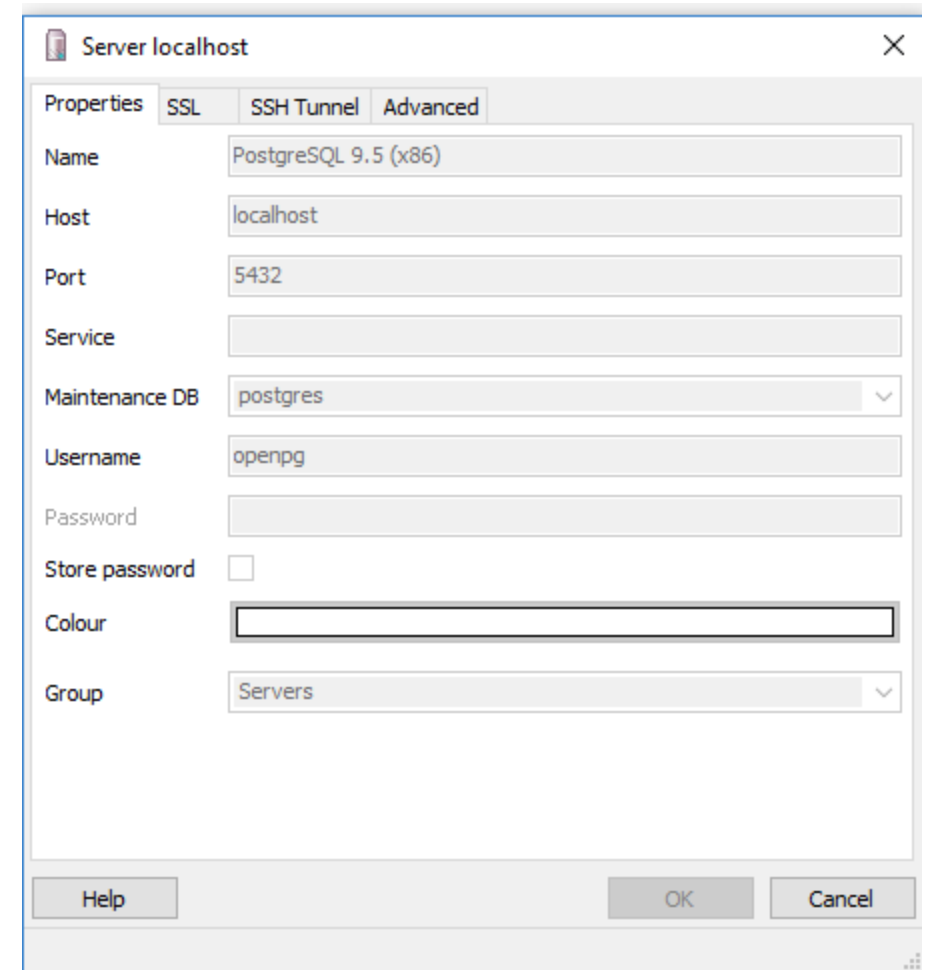
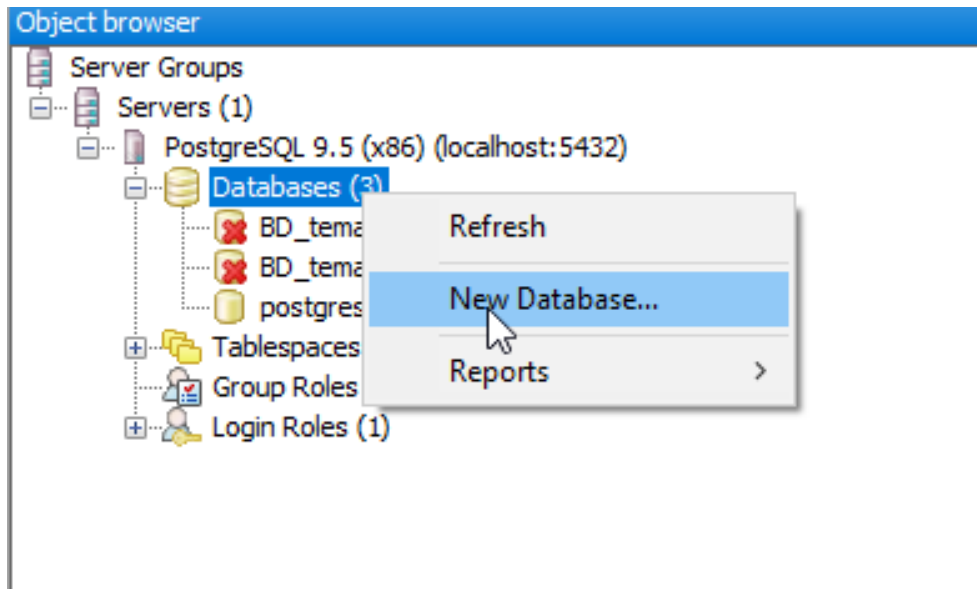
http://localhost:8069/web#action=53&model=res.partner&view_type=kanban&menu_id=133

The screenshot displays the Odoo web application interface. The browser's address bar shows the URL: `http://localhost:8069/web#action=53&model=res.partner&view_type=kanban&menu_id=133`. The URL is circled in red. The page title is 'Facturación'. The main navigation bar includes 'Clientes', 'Proveedores', 'Informes', and 'Configuración'. The user is logged in as 'Mitchell Admin'. The 'Proveedores' (Suppliers) view is shown in Kanban mode. The view includes a search bar, a filter dropdown, and a list of suppliers. The suppliers listed are:

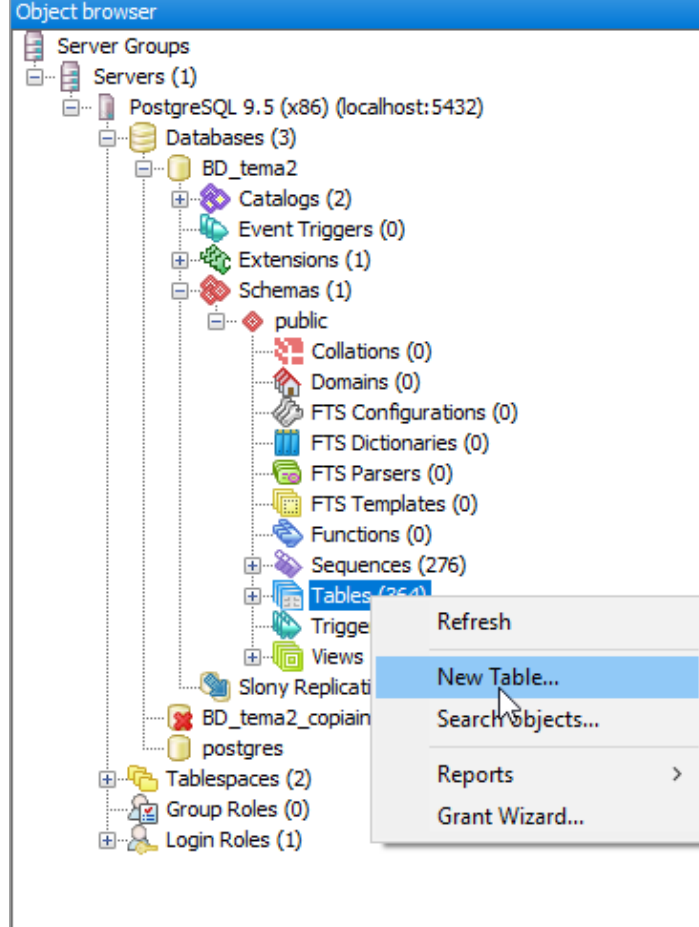
- Azure Interior**: Services, Jonesboro, Estados Unidos, azure.interior24@example.com
- Gemini Furniture**: Consulting Services, Vendor / Desk Manufacturers, Tampa, Estados Unidos, gemini.furniture39@example.com
- Ready Mat**: Vendor / Desk Manufacturers, Vendor / Office Supplies, Columbia, Estados Unidos, ready.mat28@example.com
- Wood Corner**: Vendor / Desk Manufacturers, Vendor / Office Supplies, White Tanks, Estados Unidos, wood.corner26@example.com

2. Operaciones básicas con la BD. Pgadmin III

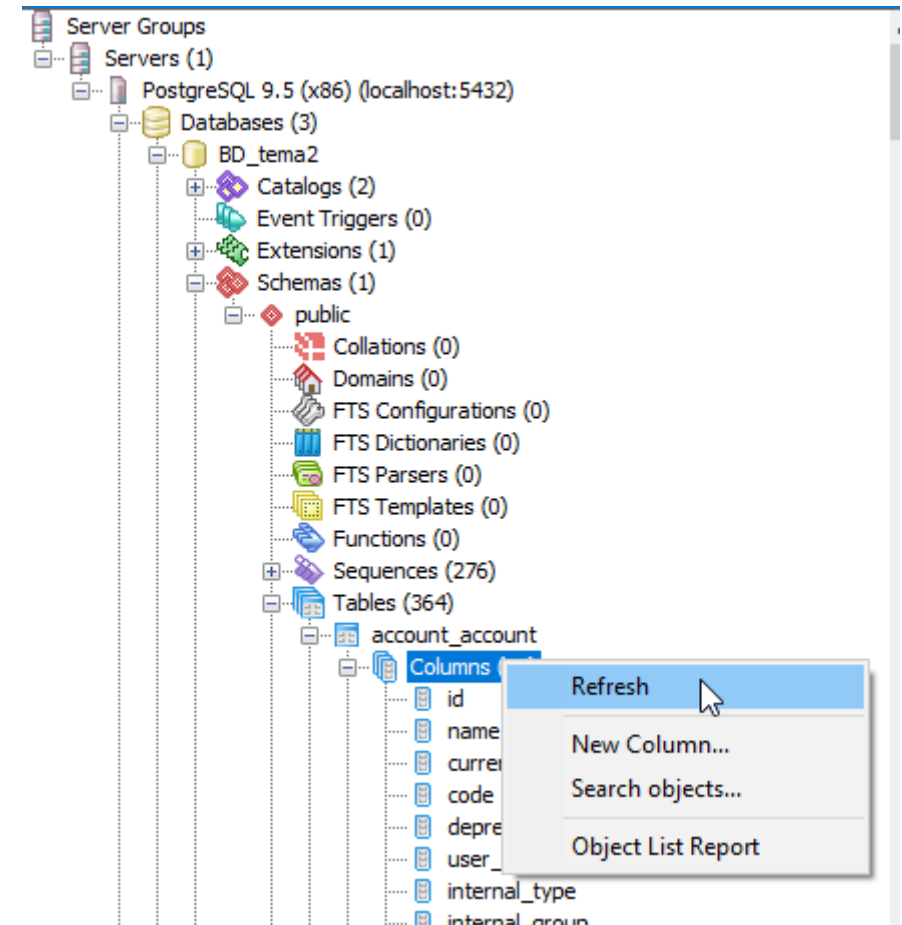
- **Edit/ Propieties** – podremos cambiar el usuario de acceso a la BD
- **Crear nueva BASE DE DATOS**



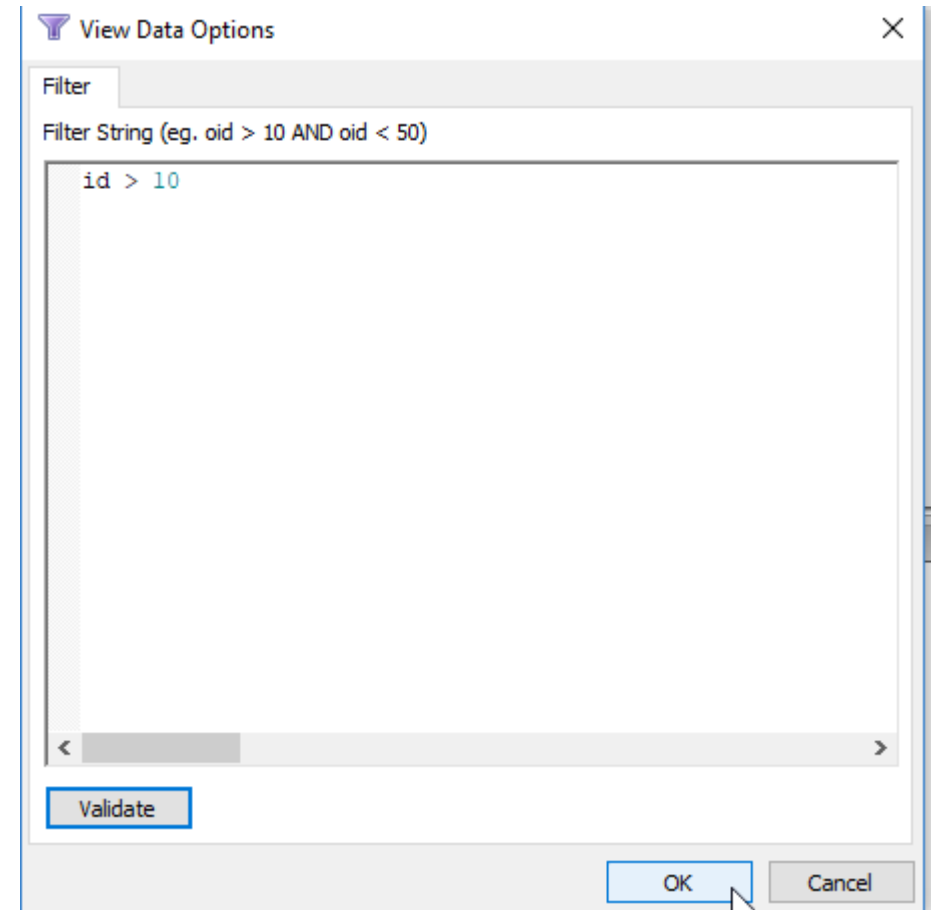
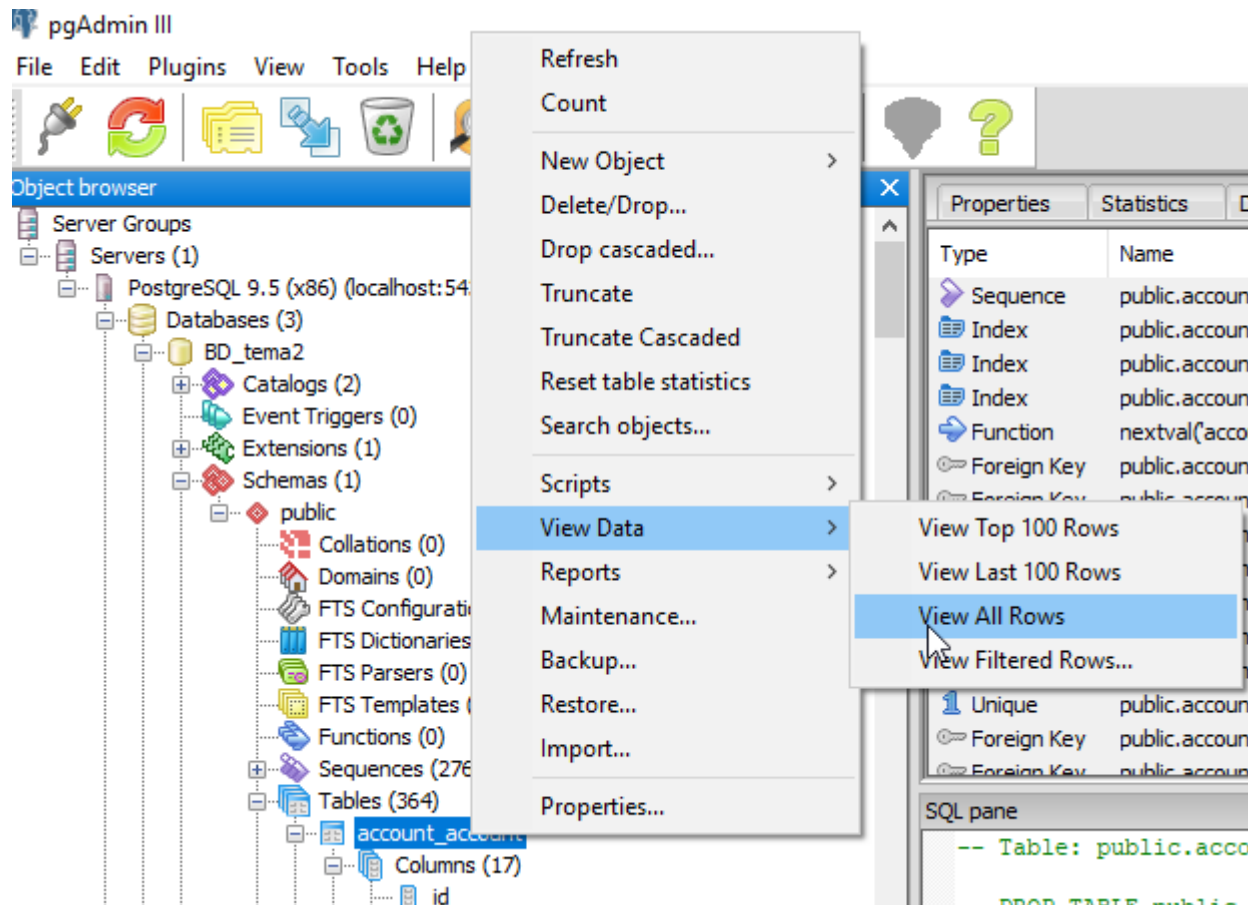
- **Crear nueva TABLA**



- **Añadir un CAMPO a la tabla (columns)**



- **Ver datos de una tabla** . Podemos mostrar todos, los 100 primeros, los 100 últimos o filtrar por un criterio



Consultas de acceso a datos

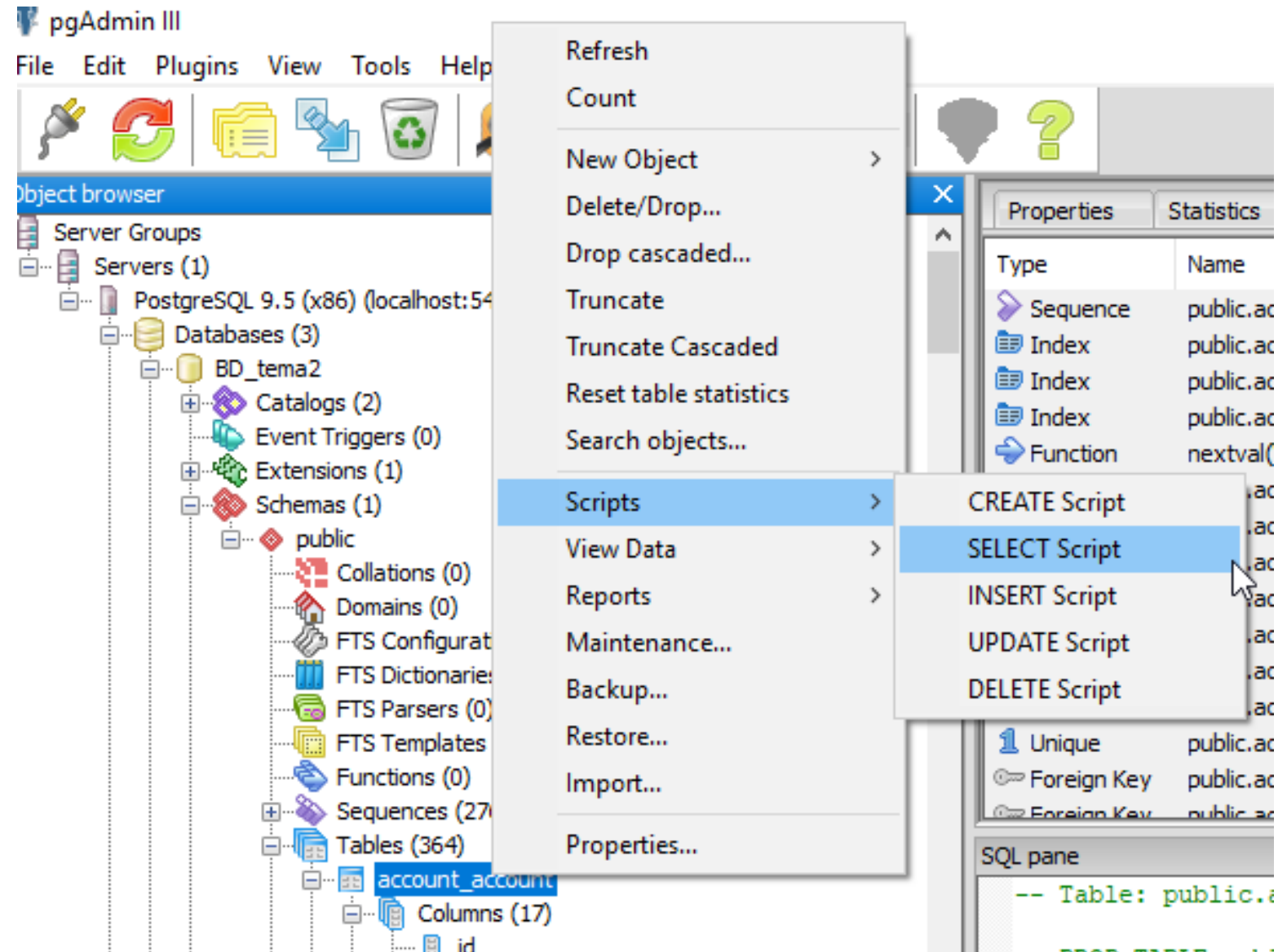
- Las consultas de acceso a datos nos permiten acceder a la información que guardan las tablas y vistas de la base de datos. Las consultas de acceso a datos sirven para indicar al sistema de gestión de la base de datos que devuelva un extracto de la información en forma de un conjunto de registros.
- Los pasos para crear una consulta son
 - Seleccionar las tablas o vistas sobre la que se consulta
 - Establecer la relación entre las tablas o vistas
 - Seleccionar los campos a mostrar
 - Ejecutar la consulta
- Las consultas pueden actuar sobre una o varias tablas o vistas, y se pueden guardar para ser utilizadas posteriormente. En ocasiones la aplicación permite realizar consultas de acceso a datos, o bien podemos conectarnos directamente al sistema gestor de base de datos.
- Las consultas de acceso a datos se pueden construir escribiendo el código en el lenguaje de consulta utilizado, como por ejemplo SQL, o bien mediante asistentes y constructores gráficos si se trata de consultas poco complejas.

Para ampliar PostgreSQL:

<http://www.postgresqltutorial.com/>

- Servidor PostgreSQL y objetos de la BD
<http://www.postgresqltutorial.com/postgresql-server-and-database-objects/>
- Tipos de campos en Postgresql
<http://www.postgresqltutorial.com/postgresql-data-types/>
- Crear una tabla <http://www.postgresqltutorial.com/postgresql-create-table/>
- Seleccionar datos de una tabla
<http://www.postgresqltutorial.com/postgresql-select/>
- Trigger (activadores) en PostgreSQL
<http://www.postgresqltutorial.com/introduction-postgresql-trigger/>
- Crear un trigger : <http://www.postgresqltutorial.com/creating-first-trigger-postgresql/>

- Crear una consulta



Query - BD_tema2 on openpg@localhost:5432 *

File Edit Query Favourites Macros View Help

BD_tema2 on openpg@localhost:5432

SQL Editor Graphical Query Builder

BD_tema2

- Catalogs
 - information_schema
 - pg_catalog
- Schemas
 - public
 - account_account
 - account_account_ac
 - account_account_ta
 - account_account_ta
 - account_account_ta
 - account_account_ter
 - account_account_ter
 - account_account_ter
 - account_account_ter
 - account_account_tyr
 - account_account_tyr
 - account_analytic_acc
 - account_analytic_dis

account_account

- ☒ id
- ☒ name
- ☐ currency_id
- ☐ code
- ☐ deprecated
- ☐ user_type_id
- ☐ internal_type
- ☐ internal_group
- ☐ last_time_entries_checked
- ☐ reconcile
- ☐ note
- ☒ company_id
- ☐ group_id
- ☐ create_uid
- ☐ create_date
- ☐ write_uid
- ☐ write_date

Columns

	Restricted Value	Operator	Value	Connector
1	account_account.id	>	10	AND

Columnas seleccionadas

Ejecutar consulta

- Editor SQL
- Editor Gráfico

Tabla/s de la consulta

- Columnas seleccionadas
- Criterios
- Orden

ACTIVIDADES

1. Crea un duplicado de tu BD y lo llamas BD_Pruebas, después entra a PGADMIN, localiza alguna de las tablas de tu BD_Pruebas (elige entre las tablas de clientes, usuarios, proveedores)
 - visualiza sus registros,
 - añade algún registro,
 - modifica algún registro,
 - comprueba desde Odoo que los cambios desde pgadmin se han realizado
 - vuelve a PGADMIN y elimina el registro añadido,
 - realiza un par de consultas sobre la BD. Explica la tabla elegida, las consultas realizadas y el proceso seguido para ello, acompáñalo de pantallazos del diseño y del resultado de las consultas.

Nota : para saber el nombre interno del modelo/tabla que queremos gestionar, visualízala en odoo y revisa la URL (model=)

3. Interfaces de entrada de datos y de procesos. Formularios (forms)

El servidor de Odoo se basa en una arquitectura donde el acceso a los datos y la lógica de negocio (Modelo) están separados de la presentación de los datos y la interfaz de usuario (Vista) a través de un componente de intermedio con acceso a ambos denominado Controlador. Este tipo de diseño se llama **Modelo-Vista-Controlador (MVC)**.

- El **Modelo** es la capa donde se trabaja con los datos y se accede a la información basada en objetos para modificarlos, actualizarlos, crearlos, etc.
- La **Vista** es la capa que nos presenta la información mediante la interfaz de usuario, trabaja con los datos pero no accede directamente a ellos. Cada vista del sistema Odoo está definida en **documentos XML**, las cuales son usadas por el marco de trabajo (framework) **del cliente web para generar vistas HTML de datos**.
- El **Controlador** es la capa que sirve de enlace entre la Vista y el Modelo, y contiene el código necesario (Python) para responder a las acciones que se solicitan en la aplicación por parte del usuario.

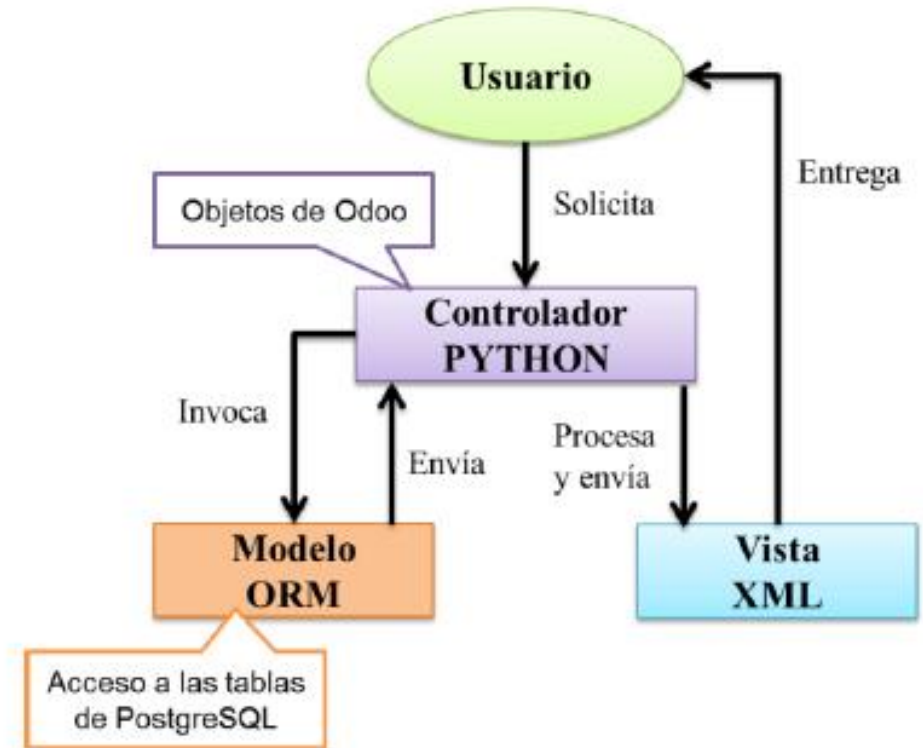


Figura 2.3 Diseño Modelo-Vista-Controlador del sistema Odoo


```
<record id="vista_película_form" model="ir.ui.view">
```

```
<field name="name">motion.película.form</field>
```

```
<field name="model">motion.película</field>
```

```
<field name="type">form</field>
```

```
<field name="arch" type="xml">
```

```
<form string="Películas">
```

```
<group col="2" colspan="2"><separator colspan="2" string="Informacion  
General"/>
```

```
<field name="nombre" select="1"/>
```

```
<field name="codigo" select="1"/>
```

```
<field name="nombre_d" select="1"/>
```

```
</group>
```

```
<group col="2" colspan="2">
```

```
<separator string="Informacion Secundaria" colspan="2"/>
```

```
<field name="fecha"/>
```

```
<field name="costo"/>
```

```
<field name="numero"/>
```

```
<field name="costo_a"/>
```

```
</group>
```

```
<notebook colspan="4">
```

```
<page string="Notas">
```

```
<field colspan="4" nolabel="1" name="notas"/>
```

```
</page>
```

```
</notebook>
```

```
</form>
```

```
</field>
```

```
</record>
```

Ejemplo:
Definición de formulario

El formulario resultante será

Menu Nueva película

Informacion General

Nombre película:

Codigo:

Nombre director:

Informacion Secundaria

Fecha de adquisicion:

Costo de la película:

Numero de copias:

Costo alquiler:

Notas:

No record selected Estado: sockets//localhost:8070 [admin] Administrator Solicitudes: No request

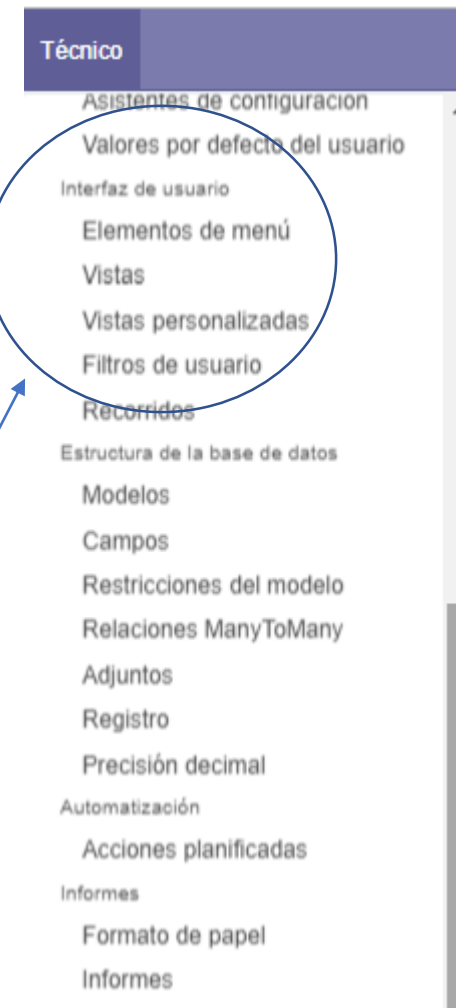
Podemos definir una o varias interfaces para cada objeto o tipo de recurso, para describir qué campos se muestran y cómo. Dependiendo de cómo se distribuyan los campos nos encontramos, entre otras, con los siguientes tipos de interfaces:

Formularios: en este tipo de interfaces se muestra un sólo registro, cuyos campos se distribuyen en la pantalla siguiendo siempre el mismo criterio, normalmente de izquierda a derecha y de arriba a abajo, de acuerdo con el orden en el que son descritos en la vista.

Árboles: este tipo de interfaces se utiliza cuando queremos mostrar un conjunto de registros en modo lista, y es útil para mostrar varios registros a la vez y realizar búsquedas sobre ellos.

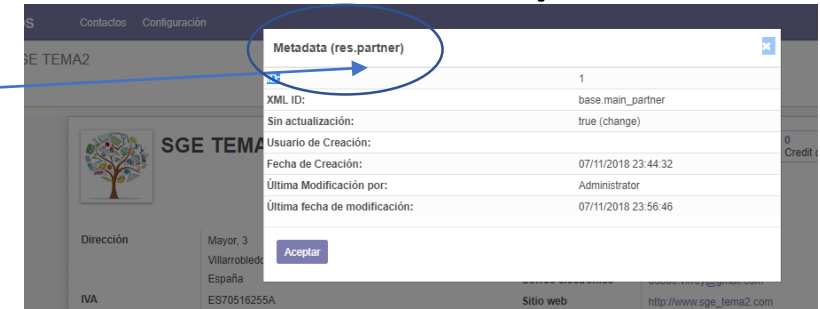
Gráficos: los gráficos son otra forma de ver los datos de un formulario, y pueden mostrarse en varios formatos y tipos para una mejor visualización de la información.

Podemos personalizar las vistas, desde el **Menú Técnico /Interfaz de usuario/Vistas** podemos seleccionar un formulario, editamos la vista y escribimos directamente el código XML. Guardamos, cerramos y recargamos el formulario para ver los cambios.



Añadir un campo a una tabla y a un formulario desde Odoo

- Con el **modo desarrollador activado** :
- Nos vamos al formulario en el que queremos añadir el campo y desde el **menú desarrollador / Ver Metadatos** visualizamos el **nombre del modelo que tenemos que modificar**
 - Por ejemplo, para los contactos el modelo es res.partner
- (también lo podemos ver en la URL: ...model=....)



1º) Añadimos el campo en la tabla del modelo

- **Ajustes / Menú Técnico / Estructura de la BD / Modelos / Buscamos el modelo** que tenemos que modificar y los editamos. Accedemos a los campos de la **tabla** para al final de todo pulsar en **“Añadir Registro”** esto nos creará un **nuevo campo que siempre tendrá el prefijo “x_”**, por ejemplo : **x_nombrenuevocampo** ponemos nombre (solo caracteres, dígitos y subrayado), etiqueta, tipo de datos, tamaño... etc por último guardar y cerrar.

2º) Añadimos el campo creado a la vista formulario

- **Abrimos el formulario**, nos posicionamos sobre el campo anterior o posterior al que queremos insertar, nos posicionamos encima y nos fijamos en el nombre del campo
- **Desde el Menú Desarrollador / Editar Vista : Formulario** veremos el código de la vista del formulario, aquí buscamos el nombre del campo vecino, ahora **añadimos encima o debajo el campo que creamos en la tabla para que lo muestre en la vista del formulario**
`<field name=“x_nombrenuevocampo”/>`
- **Guardamos el formulario, actualizamos la vista y comprobamos** que aparece el nuevo campo de la tabla en el formulario

```

<field name="company_type" widget="radio" class="oe_
<h1>
  <field name="name" default_focus="1" placeholder
</h1>
<div class="o_row">
  <field name="parent_id" widget="res_partner_many
    <field name="company_name" attrs="{ 'invisibl
    <button name="create_company" type="object"
      <span class="fa fa-plus-square"/>
      Create company
    </button>
  </div>
</div>
<group>
  <group>
    <field name="type" attrs="{ 'invisible': [('paren
    <label for="street" string="Address"/>
    <div class="o_address_format">
      <div attrs="{ 'invisible': ['|', ('parent_id'
        <field name="street" placeholder="Street..."
        <field name="street2" placeholder="Street 2.
        <field name="city" placeholder="City" class=
        <field name="state_id" class="o_address_stat
        <field name="zip" placeholder="ZIP" class="o
        <field name="country_id" placeholder="Countr
      </div>
      <field name="vat" placeholder="e.g. BE0477472701
      <field name="x_nuevocampo"/>
    </group>
  </group>
    <field name="function" placeholder="e.g. Sales D
    <field name="phone" widget="phone"/>
    <field name="mobile" widget="phone"/>

```

Editar

Crear

Contactos / SGE TEMA2

Guardar

Descartar



☐ Individual ☒ Compañía

SGE TEMA2

Dirección

Mayor, 3

Calle 2...

Villarrobledo

Estado

16660

España

IVA

ES70516255A

nuevo campo

prueba

Contactos y direcciones

Notas internas

Ventas y Compras

Facturación

Añadir

ACTIVIDADES

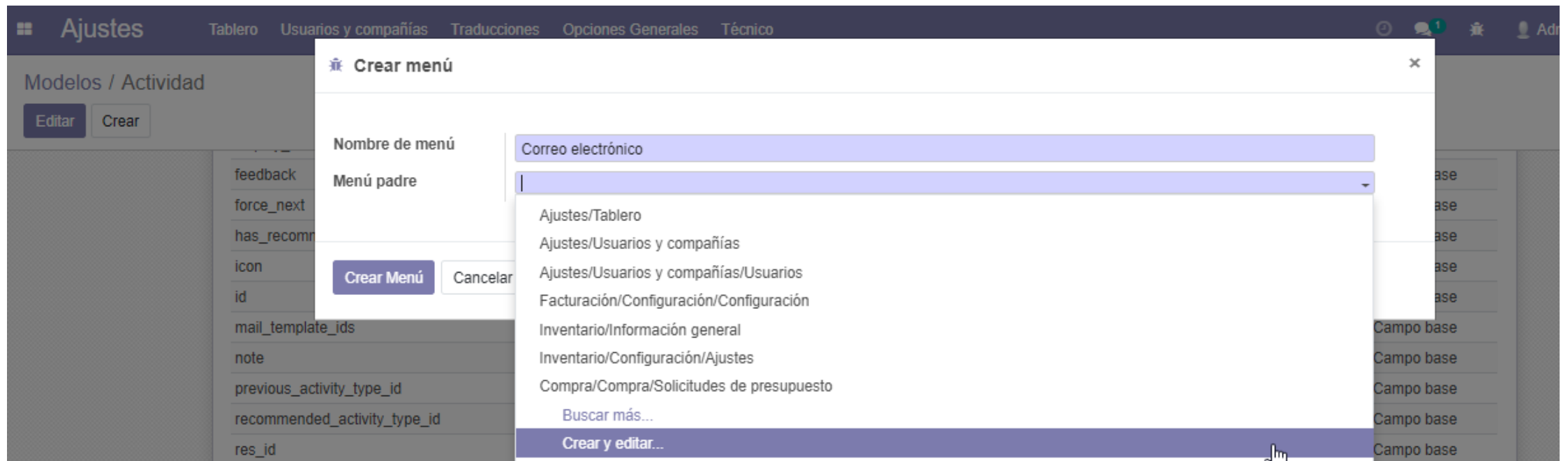
- 2. Añade un nuevo campo a la tabla de clientes (partners). Explica el tipo de campo y el proceso seguido para crearlo.
- 3. Añade el nuevo campo al formulario de clientes. Explica el proceso seguido y muestra un pantallazo con el resultado obtenido
- 4. Ahora realiza el 2 y el 3 para la tabla de empleados de tu ERP (si no tienes la aplicación que gestiona empleados instalada instálala)

4. Menus

- En el menú del cliente Odoo se van añadiendo los menús de las aplicaciones que se instalan, con las opciones para las que cada usuario tiene acceso según sus permisos.
- Podemos personalizar los menús de las aplicaciones y crear nuestros propios menús en la interfaz de Odoo.

Se pueden crear menús de dos formas diferentes :

A) Desde **Menú Técnico/Estructura de la base de datos / Modelos** se selecciona el objeto del modelo que queremos que se ejecute en la opción de menú y se pulsa **en Crear Menú**, se pone un nombre y se selecciona o crea un menú padre y listo. Recargamos y ya tendremos el nuevo menú accesible desde la barra de menús de Odoo.



- B) Desde **Menú Técnico / Interfaz de Usuario / Elementos de Menú** pulsando en **CREAR**. Nos sale la siguiente ventana en la que ponemos el nombre al menú, elegimos su **menú padre**, el **tipo de acción** y el **objeto asociado** a la acción.

Ajustes Tablero Usuarios y compañías Traducciones Opciones Generales Técnico

Elementos de menú / Nuevo

Guardar Descartar

Menú	Menu_Montse	Ruta completa	Menu_Montse
Menú padre	<div>Ajustes/Tablero</div>	Acción	<div>ir.actions.act_window</div>
Secuencia	10	Archivo de icono web	<div>ir.actions.act_window</div>
		Imagen de icono web	<div>Abono</div>

Permisos de acceso Submenús

Nombre del grupo

Agregar registro

Los menús también se pueden editar desde :

Menú Técnico / Interfaz de Usuario / Elementos de Menú

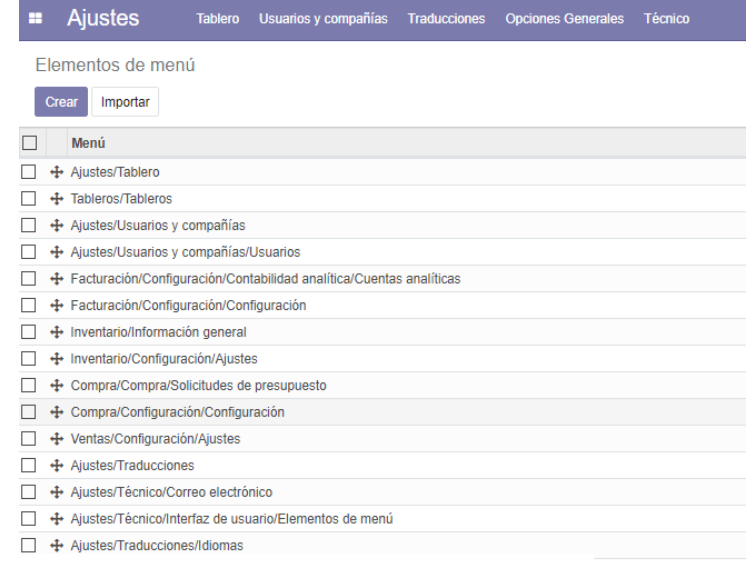
Si queremos meter una opción de menú en nuestro menú personalizado y no sabemos que acción es, lo mejor es seleccionar y editar la opción de menú que queremos clonar y ver el tipo de acción y nombre de la acción

Los menús tienen asociada la acción que deben realizar. Hay diferentes tipos de acciones, las más importantes:

- **Window:** Abre una vista en una nueva ventana.
- **Report:** Imprime un informe.
- **Wizard:** Ejecuta un asistente para realizar un determinado trabajo o proceso.

Por tanto, cuando creamos un menú, debemos indicar el tipo de acción que tiene asociada. Si la acción no está creada debemos utilizar el botón **Nuevo** para crearla. Los campos a introducir variarán según la acción de que se trate. Por ejemplo, para crear una nueva acción que abra una vista debemos introducir al menos los siguientes datos:

- **Nombre de la acción.**
- **Tipo de la acción**, en este caso, `ir.actions.acc_window`.
- **Datos de la vista:** tipo (árbol, formulario, etc.) y nombre.



ACTIVIDADES

- 5. Crea un nuevo menú llamado MENU_TUNOMBRE y crea en él al menos 3 opciones de menú. Explica que opciones has elegido y el proceso seguido para ello. Por último, muestra un pantallazo del menú creado con sus opciones.
- 6. Modifica alguno de los menús predefinidos del ERP añadiendo una opción de menú nueva dentro. Indica que cambio has realizado y muestra un pantallazo donde se vea el menú anterior y el menú modificado con la nueva opción añadida.

5. Informes y listados (Reports).

Los informes y listados de la aplicación son una forma de mostrar los datos para mejorar su visualización. Muchos informes y listados están creados, y son accesibles desde los distintos menús de la aplicación.

Algunos de estos documentos pueden ser:

- Informes contables.
- Albaranes.
- Pedidos.
- Recibos.
- Reclamaciones a proveedores y/o clientes.
- Informes estadísticos.
- Generación de etiquetas para un conjunto de registros.
- Informes de agrupamiento: permiten mostrar los registros existentes para un mismo valor de un campo.
- Impresión de documentos del sistema.

Los informes y listados que incorpora la aplicación se pueden utilizar tal cual se proporcionan, o bien adaptar su diseño a la imagen corporativa de la empresa. También se pueden **añadir nuevos informes** instalándolos como módulos independientes. En el caso de **Odoo** los módulos extra que contienen exclusivamente informes de algún tipo llevan en su nombre la etiqueta "report".

Menú técnico : Informes

Técnico

Asistentes de configuración
Valores por defecto del usuario
Interfaz de usuario
Elementos de menú
Vistas
Vistas personalizadas
Filtros de usuario
Recorridos
Estructura de la base de datos
Modelos
Campos
Restricciones del modelo
Relaciones ManyToMany
Adjuntos
Registro
Precisión decimal
Automatización
Acciones planificadas
Informes
Formato de papel
Informes

Nos muestra todos los informes del sistema, su **nombre**, **modelo al que están asociados**, **nombre de la plantilla donde está definido** Etc. Desde aquí podemos:

- **EDITAR** un informe de cualquier módulo instalado en el sistema
- **CREAR** uno nuevo (aunque esto requiere varios pasos, la creación de otros objetos y su conexión) – Con la herramienta Odoo STUDIO se pueden generar fácilmente informes personalizados aunque desde la v12 ya es de pago. Hay un módulo gratuito de plantillas de informes que nos permite elegir entre varias plantillas diferentes para nuestros albaranes, facturas ,..... etc

Ajustes						
Tablero Usuarios y compañías Traducciones Opciones Generales Técnico						
Informes						
Buscar...						
Crear Importar						
Filtros Agrupar por Favoritos						
1-33 / 33						
<input type="checkbox"/>	Nombre	Modelo	Tipo de acción	Nombre de plantilla	Tipo de informe	Guardar como Prefijo del Adjunto
<input type="checkbox"/>	Auditoría de Libros	account.common.journal.report	ir.actions.report	account.report_journal	PDF	
<input type="checkbox"/>	Código de barras de ubicación	stock.location	ir.actions.report	stock.report_location_barcode	PDF	
<input type="checkbox"/>	Código de barras del paquete (PDF)	stock.quant.package	ir.actions.report	stock.report_package_barcode_small	PDF	
<input type="checkbox"/>	Código de barras del paquete con contenido	stock.quant.package	ir.actions.report	stock.report_package_barcode	PDF	
<input type="checkbox"/>	Delivery Slip	stock.picking	ir.actions.report	bi_professional_reports_templates.report_deliveryslip	PDF	
<input type="checkbox"/>	Factura PRO-FORMA	sale.order	ir.actions.report	sale.report_saleorder_pro_forma	PDF	
<input type="checkbox"/>	Facturas	account.invoice	ir.actions.report	account.report_invoice_with_payments	PDF	(object.state in ('open','in_payment','paid')) and ('INV'+(object.number or "").replace('/','')+ '.pdf')
<input type="checkbox"/>	Facturas sin pago	account.invoice	ir.actions.report	account.report_invoice	PDF	(object.state in ('open','in_payment','paid')) and ('INV'+(object.number or "").replace('/','')+ '.pdf')
<input type="checkbox"/>	Guía técnica	ir.module.module	ir.actions.report	base.report_irmodulereference	PDF	
<input type="checkbox"/>	Información general del modelo	ir.model	ir.actions.report	base.report_irmodeloverview	PDF	
<input type="checkbox"/>	Inventario	stock.inventory	ir.actions.report	stock.report_inventory	PDF	

Editar un informe

Al seleccionar un informe para editar, activando las **vistas QWEB** se puede visualizar y modificar el XML del informe (el PDF normalmente está definido en la vista Qweb **nombredelmodelo.document**)

Informes / Facturas

Editar

Crear

Nombre
Facturas
Tipo de informe
PDF
Formato de papel

Modelo
account.invoice
Nombre de plantilla
account.report_invoice_with_payments
Nombre del reporte
impreso
(object._get_report_base_filename())

Seguridad Propiedades avanzadas

Nombre del grupo

Eliminar del
menú
"Imprimir"

</> Vistas QWeb

Informes / Facturas sin pago / Vistas / report_invoice_document

Guardar

Descartar

2 / 4 < >

Nombre de la vista

report_invoice_document

Campo hijo

Tipo de vista

QWeb

Vista heredada

Modelo

Secuencia

16

Activo

☒

Modo de herencia de
la vista

Datos del modelo

ID externo

report_invoice_document

account.report_invoice_document

Estructura Permisos de acceso Vistas heredadas

Editar traducciones

```
<?xml version="1.0"?>
<t t-name="account.report_invoice_document">
  <t t-call="web.external_layout">
    <t t-set="o" t-value="o.with_context({'lang': lang})"/>
    <t t-set="address">
      <address t-field="o.partner_id" t-options="{&quot;widget&quot;: &quot;contact&quot;, &quot;fields&quot;: [&quot;address&quot;,
      <div t-if="o.partner_id.vat" class="mt16"><t t-esc="o.company_id.country_id.vat_label or 'Tax ID'"/>: <span t-field="o.partner_
    </t>
    <div class="page">
      <h2>
        <span t-if="o.type == 'out_invoice' and o.state in ('open', 'in_payment', 'paid')>Invoice</span>
        <span t-if="o.type == 'out_invoice' and o.state == 'draft'">Draft Invoice</span>
        <span t-if="o.type == 'out_invoice' and o.state == 'cancel'">Cancelled Invoice</span>
      </h2>
    </div>
  </t>
</t>
</t>
```

Ajustes

Tablero

Usuarios y compañías

Traducciones

Opciones Generales

Técnico

2

🔍

Informes / Facturas sin pago / Vistas

Crear

Importar

Activo x Buscar...

Filtros

Agrupar por

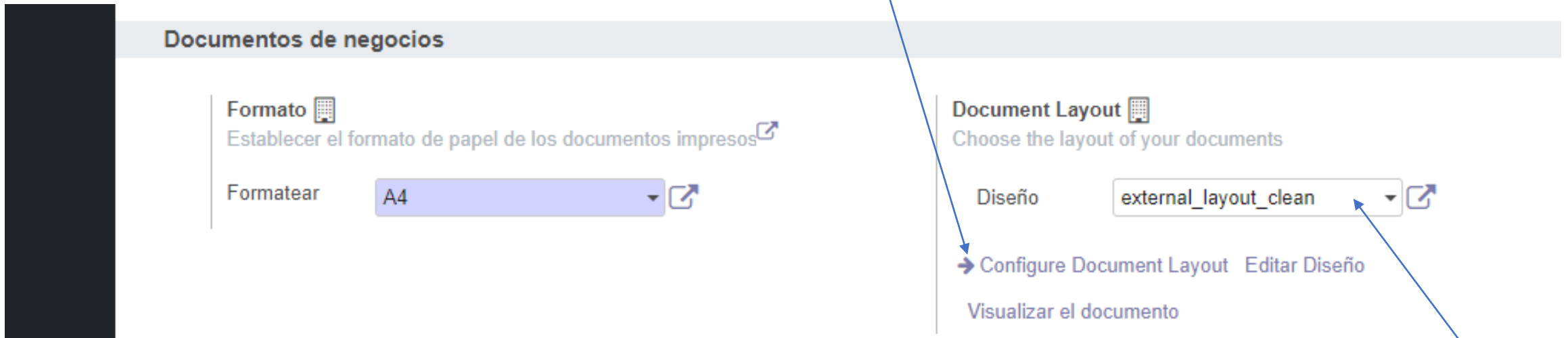
★ Favoritos

	Nombre de la vista	Tipo de vista	Modelo	ID externo
<input type="checkbox"/>	report_invoice	QWeb	account.report_invoice	
<input checked="" type="checkbox"/>	report_invoice_document	QWeb	account.report_invoice_document	
<input type="checkbox"/>	report_invoice_document_with_payments	QWeb	account.report_invoice_document_with_payments	
<input type="checkbox"/>	report_invoice_with_payments	QWeb	account.report_invoice_with_payments	

- Los informes impresos que utilizamos en Odoo están **diseñado en XML**.
- La mayoría de los módulos (carpeta addons) contiene una **carpeta llamada REPORTS** en la que se encuentran estos archivos XML que podemos modificar manualmente si disponemos de los conocimientos suficientes de XML
- Durante la instalación de un módulo, cada archivo XML de la carpeta reports del módulo genera un registro en la **tabla ir_act_report_xml** de la base de datos de la empresa y se podrán explorar con **pgAdminIII**,
- Estos registros XML, también se pueden explorar directamente desde Odoo accediendo a **Menú Técnico>Acciones>Informes**
- Alternativamente, podemos instalar módulos externos diseñados como un conjunto de plantillas predefinidas que cambian el diseño de los informes de las aplicaciones instaladas.
- El **módulo Odoo Studio** incorpora herramientas para crear y personalizar informes, aunque desde la v12 está en la versión Enterprise.
- Siempre que actualicemos un módulo se perderán las modificaciones que hayamos realizado en sus informes.

Personalizar plantilla de informes

- En los **Ajustes / Opciones Generales / Configure Document Layout** se puede personalizar el encabezado y el pie de la plantilla general en la que se basan los documentos de Odoo. Esta plantilla se aplica por defecto como base para los informes que se generen en las aplicaciones.



- En el **desplegable Diseño** también se podrá seleccionar y editar cualquier plantilla concreta de documentos de Odoo

Actividades

7. Muestra desde PGADMIN la lista de informes de la BD que estás utilizando. Después muestra la misma lista desde Odoo

8. Configura la plantilla de informes con una plantilla y le añades la frase “SGE. Tema 3” en el encabezado y 3 líneas en el pie contengan “Empresa de prueba para SGE” – “Tu nombre” - “IES Virrey Morcillo”. Crea una factura y muestras el informe impreso, comprueba que se aplican tus cambios

9. Accede a la vista QWEB del informe de las facturas sin pago, (**account.move - report.invoice.document**) y revisa su contenido en XML.

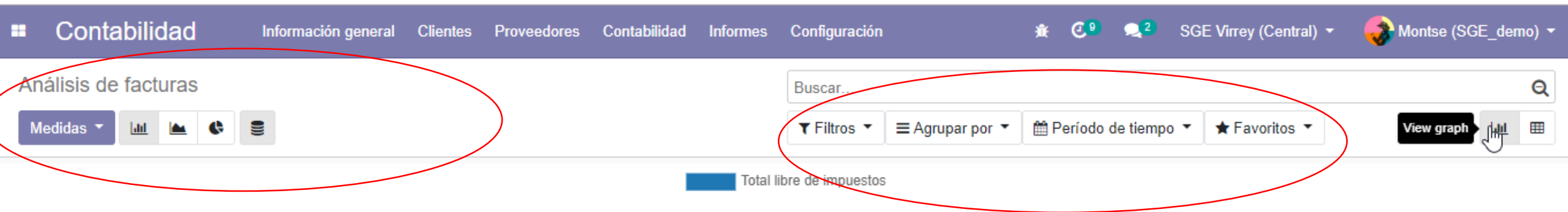
(Opcional : Si manejas XML podrías editar y modificar lo que quieras del informe, por ejemplo añadir el campo que creaste en la tabla de clientes o cambiar la ubicación y/o textos del informe)

5. Gráficos

- Los **gráficos y las tablas** representan e interpretan información procedente de diferentes fuentes, de forma clara, precisa y ordenada. Casi todos los tipos de información puede organizarse en una tabla de datos y ser representada en algún tipo de gráfico.
- Según las características y la cantidad de datos, conviene utilizar uno u otro gráfico.. Los gráficos permiten visualizar la información contenida en las tablas de manera rápida y sencilla, demostrando con mayor claridad la relación que estos datos tienen entre sí.
- En la mayoría de aplicaciones, los gráficos para análisis de datos, están en un menú Informes.



- Por ejemplo, en Contabilidad / Informes – Facturas tendremos acceso al análisis de las facturas aplicando distintos filtros y mostrando la información con distintos tipos de gráficos



Actividades

- 10. Genera 4 facturas, a continuación accede al menú Informes / Facturas de Facturación y realiza un análisis de las facturas representando la información con diferentes tipos de gráficos.

6. Tratamiento de la información

El tratamiento de la información se lleva a cabo a través de distintos procesos. Antes de poder realizar ningún proceso es necesario introducir la información propia de la compañía. Por ejemplo, si vamos a llevar la contabilidad de la empresa habrán de configurarse las cuentas contables, se tendrán que definir las cuentas bancarias y los diarios de compras, ventas y caja o banco. También habrán de definirse los impuestos.

Por ejemplo, si tenemos instalado el módulo de contabilidad, cuando creemos una empresa cliente o proveedor, deberemos asociarle una serie de cuentas contables, como las de cobro al cliente y pago al proveedor, por lo que deberemos tener creadas dichas cuentas antes de poder dar de alta a nuestros clientes y proveedores, e igual con los productos.

Dentro de la información propia de la compañía deberemos crear los clientes, proveedores y productos. En cualquiera de los casos, existe una información mínima que tendremos que introducir antes de poder empezar a trabajar.

Procesos empresariales: pedidos, albaranes, facturas, asientos predefinidos, trazabilidad y producción.

Entre los procesos a realizar por un sistema de planificación empresarial destacan los siguientes:

- Contabilidad. Incluye los procesos donde se reflejan las operaciones económicas realizadas por la empresa, la determinación de los costes de la empresa y los presupuestos del ejercicio fiscal. Se proporcionan asientos predefinidos para la introducción rápida de asientos sin necesidad de tener conocimientos de contabilidad.
- Operaciones de compra:
 - Crear una orden de compra o pedido de compra.
 - Recibir los bienes.
 - Controlar la factura de compra.
 - Registrar el pago al proveedor.
- Operaciones de venta:
 - Crear una orden de venta o pedido de venta y recibir la conformidad del cliente.
 - Preparar los bienes a enviar al cliente y realizar el albarán y la entrega.
 - Realizar la factura de venta.
 - Registrar el cobro al cliente o pago del cliente.
- Trazabilidad: se llama así al proceso de la entrada del producto hasta la salida del mismo.

Es posible que por las necesidades de la empresa no sea necesario utilizar todos los procesos del ERP, por ejemplo, puede ser usado sólo como un CRM, o sólo como un programa de contabilidad, pero su verdadera potencia se alcanza con la integración de todas sus funciones.

7. Herramientas de monitorización y de evaluación del rendimiento.

Las **herramientas de monitoreo** de aplicaciones proporcionan a los administradores la información que necesitan para **descubrir rápidamente, aislar y resolver problemas que impactan negativamente en el rendimiento de una aplicación**. Pueden ser específicas para una aplicación en particular, o monitorizar varias aplicaciones en la misma red, recogiendo datos sobre el uso del CPU cliente, demandas de memoria, rendimiento de la base de datos y ancho de banda.

Estas herramientas no son específicas para sistemas ERP sino que permiten la monitorización de otros tipos de sistemas informáticos.

Además las propias aplicaciones ERP tienen sus propias herramientas de auditoria, monitoreo y control del rendimiento.

ACTIVIDADES

11. Busca información sobre las siguientes herramientas de monitorización y/o rendimiento y añade alguna otra a la lista.

- ManageEngine Applications Manager
- **Ingeo:**
- **PandoraCMS:**
- **SocialMention:**

Auditoría de control de acceso a los datos.

Trazas del sistema en Odoo

La actividad de los programas, sobre todo si se trata de programas que se ejecutan en servidores, queda registrada en ficheros del sistema llamados logs. En ocasiones podemos querer examinar los ficheros de trazas del sistema para realizar un control de acceso a los datos, ya que estos ficheros de trazas van almacenando toda la actividad y eventos que ocurren en el equipo: quién entra, qué comandos ejecuta, qué errores muestran las aplicaciones, etc.

En el caso de Odoo las trazas del sistema se almacenan en un fichero que registra toda la actividad e incidencias del servidor : el archivo **odoo.log** que está en la carpeta **C:\Program Files (x86)\Odoo 13.0\server**

Actividades

12. Visualiza las trazas del servidor Odoo en los últimos 15 minutos.

13. Realiza las siguientes tareas documentando los procedimientos que realizas :

- A. Añade algún campo en dos de las tablas de Odoo.
- B. Modifica las vistas de los modelos modificados para que los nuevos campos sean accesibles.
- C. Accede a las tablas de BD en PostgreSQL y modifica el nuevo campo en los registros de la primera tabla.
- D. Genera un archivo de importación de la segunda tabla, modifica los registros con 'valores en el nuevo campo y realiza una importación.
- E. Cambia la plantilla de documentos, pon un nuevo tipo, encabezado y pie.
- F. Por último, crea un menú llamado MIMENU con acceso a las vistas formulario de las dos tablas modificadas y a un informe en el que se vean los cambios realizados en la plantilla general (por ejemplo una factura).

Para ampliar :

- Curso XML
 - <https://desarrolloweb.com/manuales/manual-introduccion-xml.html>
- Tutorial PostgreSQL
 - <http://www.postgresqltutorial.com/>