UNIDAD DIDÁCTICA 3

ORGANIZACIÓN Y CONSULTA DE LA INFORMACIÓN

MÓDULO PROFESIONAL: SISTEMA DE GESTIÓN EMPRESARIAL

ÍNDICE

RESUMEN INTRODUCTORIO 3
INTRODUCCIÓN 3
CASO INTRODUCTORIO 4
1. BASES DE DATOS SOPORTE DE LA INFORMACIÓN 5
1.1 Sistemas gestores de bases de datos compatibles con el software 5
1.2 Tablas y vistas de la base de datos 8
1.3 Modelos de Odoo 10
1.4 Definición de campos 12
1.5 Consultas de acceso a datos 15
1.5.1 Ejemplo datos de cliente 17
1.5.2 Ejemplo datos de cliente combinados 19
1.6 Procedimientos almacenados de servidor 21
1.7 Cálculos de pedidos, albaranes, facturas, asientos predefinidos, trazabilidad, producción, entre otros 23
1.7.1 Pedidos y Albaranes 23
1.7.2 Facturación 24
1.7.3 Contabilidad 25
1.7.4 Operaciones: Fabricación, producción, proyectos y otros 26
2. INTERFACES DE ENTRADA DATOS, PROCESOS Y LISTADOS. FORMULARIOS 28
2.1 Formularios 29
2.2 Informes y listados de la aplicación 30
2.2.1 Diseñador integrado 31
2.2.2 Documentos Odoo 33
2.3 Búsqueda de información 34
2.4 Gráficos 36
2.5 Procesos de extracción de datos en sistemas de ERP-CRM y almacenes de datos. Automatización 37
2.5.1 Extracción de datos a través de Odoo 38
2.5.2 Extracción de datos a través de un generador de informes externo 38
2.5.3 Instalación de Jaspersoft Studio 39
2.5.4 Primer report con Jaspersoft Studio 41
2.5.5 Diseño con Jaspersoft Studio 46
2.5.6 Previsualización con Jaspersoft Studio 49
2.6 Exportación de datos e informes 51
2.6.1 Exportar a PDF 51
2.6.2 Exportar a Excel/CSV 52

2.7 Inteligencia de negocio (Business Intelligence) 53
2.7.1 BI dentro de Odoo 54
2.7.2 BI con herramientas externas 54
2.7.3 Arquitectura de BI 55
3. HERRAMIENTAS DE MONITORIZACIÓN Y DE EVALUACIÓN DEL RENDIMIENTO 55
3.1 Trazas del sistema (logs) 57
3.2 Auditorías de control de acceso a los datos 62
3.3 Incidencias: Identificación y resolución 67
RESUMEN FINAL 69

RESUMEN INTRODUCTORIO

En esta unidad nos adentramos en la importancia de los datos y la información dentro de los sistemas de gestión de empresarial.

Comenzamos con el centro de cualquier sistema, los gestores de bases de datos, haciendo un recorrido desde lo más general y recordando conceptos de bases de datos relacionales para acabar esta parte centrándonos en nuestro sistema Odoo y su base de datos PostgreSQL.

Desde el software de gestión existen diversos mecanismos de entrada y salida de información que el usuario puede utilizar. Pero también como desarrolladores introducimos los softwares de generación de informes, en nuestro caso JasperSoft Studio, en su versión community.

Por último, cerramos la visión de un software de gestión empresarial con el apartado de monitorización, acceso a la información y resolución de incidencias, tan importante en el mantenimiento de este tipo de software.

INTRODUCCIÓN

En el actual panorama empresarial, la información es un recurso estratégico para la toma decisiones rápidas y acertadas, no siendo suficiente su almacenamiento, sino que es necesario organizarlos, consultarlos y transformarlos en conocimiento útil para el día a día.

Los sistemas de gestión empresarial permiten centralizar toda esa información y facilitar su uso a través de bases de datos, formularios, listados o informes. De esta manera, distintos departamentos como ventas, compras, contabilidad o logística trabajan de forma coordinada y con datos fiables.

Estos sistemas son utilizados independientemente del tamaño de las empresas, tanto en grandes compañías como en pequeñas y medianas empresas, al facilitar la gestión en el registro de operaciones, análisis y presentación de resultados de forma clara. Además, la posibilidad de contar con herramientas de informes y de inteligencia de negocio que ofrecen nuevas oportunidades para anticipar tendencias y mejorar la eficiencia.

Su evolución va orientada hacia el tratamiento de una información que cada vez se espera sea más abundante y compleja, por lo que el cómo manejarla y aprovecharla mediante herramientas de gestión se convertirá en una competencia fundamental.

CASO INTRODUCTORIO

Como desarrollador de aplicaciones te incorporas al departamento de informática de una pequeña empresa que está en una evolución y crecimiento actual, una empresa de logística dentro del sector alimentario.

Junto a tu jefe en el departamento de informática de la empresa estáis analizando las posibilidades que tiene el ERP Odoo, el cual ya tenéis instalado en un servidor de pruebas en su versión community.

Queréis analizar las diferentes opciones que proporciona el ERP respecto a sus módulos, su interacción con los usuarios y las opciones de extracción de información directamente con el ERP.

Al tener acceso a la base de datos y ser abierta, también queréis comprobar las posibilidades que ofrece de explotación de información directamente, con herramientas ofimáticas y con algún generador de informes.

Al final de esta unidad tendrás los conocimientos para poder conocer la base de datos que hay detrás de Odoo, las posibilidades que te ofrece con Odoo y las posibilidades con un gestor externo.

1. BASES DE DATOS SOPORTE DE LA INFORMACIÓN

A la hora de en la empresa centralizar la información en un único sistema, eres el responsable de elegir qué ventajas aporta PostgreSQL frente a otros gestores de bases de datos y justificar por qué es la mejor opción para Odoo, por ejemplo, en referencia al uso de bases de datos como soporte de la información.

La información es actualmente uno de los recursos más preciados a nivel empresarial, lo hemos presentado e introducido en las anteriores unidades didácticas. De hecho, el estudio, implantación y uso de un sistema de gestión empresarial basa su importancia en el hecho de tener información de calidad, normalizada y unificada.

Y en el centro de un software de gestión empresarial se encuentra la base de datos. En la unidad 1, afirmamos que un conjunto de herramientas de gestión no es un ERP, ya que uno de los puntos importantes de un ERP es tener una base de datos centralizada. La base de datos permite a la empresa tener la información centralizada en un único lugar, por lo que introducción y explotación es coherente.

Pongamos como ejemplo que una empresa venda mochilas, bolsos y maletas. Imaginemos que una mochila tenga como código de referencia VLC-56, una tienda que venda la mochila a un cliente utiliza Word para imprimir un ticket, en el ticket el vendedor escribe vlc-56. Por otra parte, en el almacén la persona que recepciona las mochilas escribe en su Excel vlc56. Por último, desde administración, tienen una Access y escribe el asiento como VLC-56. Como observamos al tener 3 herramientas no conectadas entre sí, tenemos 3 movimientos diferentes de productos. Teniendo una base de datos centralizada, a priori, se reducen los posibles errores al utilizar todos los agentes de la empresa el mismo código.

1.1 Sistemas gestores de bases de datos compatibles con el software

Algunas veces un mismo software de gestión empresarial permite utilizar diferentes SGBD (Sistemas de Gestión de Bases de Datos), situación en la cual hay que analizar cuál de ellos es mejor en función de las necesidades de la empresa y de su coste, teniendo en cuenta que hay algunos muy potentes con versiones gratuitas. Así, por ejemplo, una tienda de bicicletas que adquiera un ERP para llevar la gestión informatizada de los circuitos compraventa, inventario y contabilidad es posible que tenga bastante con un SGBD ofimático, como por ejemplo Microsoft Access, mientras que un supermercado, con el mismo ERP, es posible que no tenga bastante con un SGBD ofimático y precise uno de mayor potencia.

Hoy, incluso para pymes, los ERP/CRM se apoyan en motores de base de datos de servidor (PostgreSQL, MySQL/MariaDB, SQL Server, Oracle), descartando los "SGBD ofimáticos" como Access, por no ser adecuados para la operación multiusuario, concurrencia, seguridad ni escalabilidad que exige un ERP/CRM.

En el caso de Odoo, por ejemplo, en su versión 18 (2024/2025), el motor compatible y recomendado es PostgreSQL. Al elegir un SGBD, deben tenerse en cuenta factores como: rendimiento, escalabilidad vertical/horizontal, alta disponibilidad, cifrado, copias y soporte a largo plazo.

ARTÍCULO DE INTERÉS

La base de datos que vamos a trabajar es relacional, pero actualmente las bases de datos NoSQL son cada vez más utilizadas y sobre todo en almacenamiento de grandes cantidades de datos. Conoce más en este artículo:

En el proceso de instalación del software de Odoo, visto en la unidad 2, ya comentábamos que el SGBD que se utiliza es PostgreSQL, un potentísimo SGBD que además es gratuito.

Para acceder a PostgreSQL disponemos de muchas herramientas. Entre ellas:

- Herramienta gráfica pgAdmin 4 (oficial, escritorio o web).
- Consola textual psql.
- Herramienta gráfica phpPgAdmin, con servidor web (necesita PHP), ha quedado sin mantenimiento y presenta incompatibilidades con PHP 8. x.
- Alternativas multiplataforma: DBeaver, DataGrip (comercial).

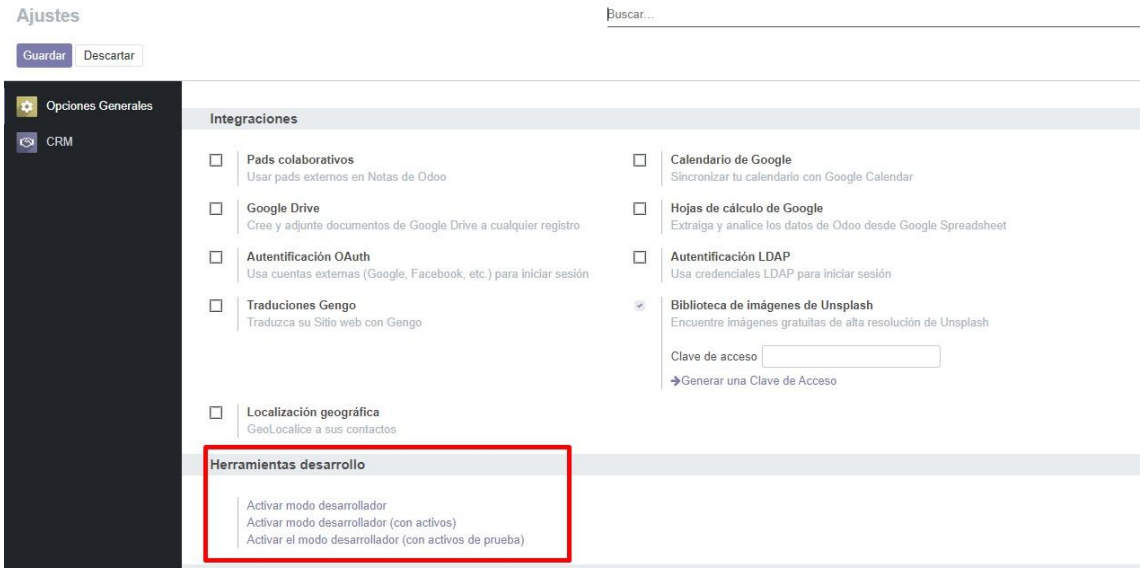
ENLACE DE INTERÉS

En este enlace encontrarás la información sobre pgAdmin 4.

VÍDEO DE INTERÉS

Visualiza al detalle el paso a paso de instalación de PostgreSQL en Windows 10.

Será necesario para muchas de las acciones que realicemos activar el modo desarrollador, desde Ajustes → Herramientas de desarrollador → Activar modo desarrollador, o añadiendo `?debug=1` a la URL.



Herramientas de desarrollo

Fuente: Elaboración propia

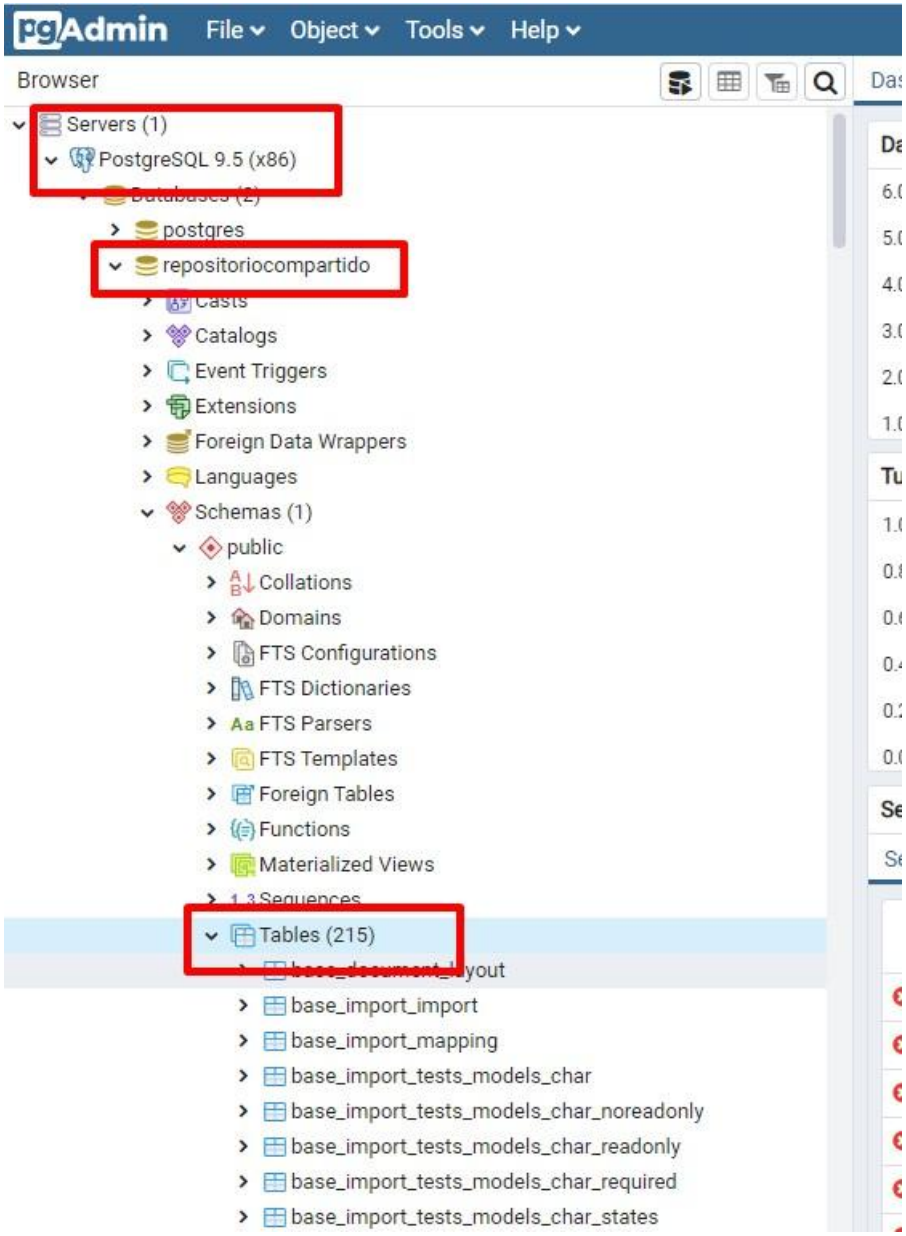
VÍDEO DE INTERÉS

Conoce en detalle cómo activar el modo desarrollador.

1.2 Tablas y vistas de la base de datos

Tal y como hemos visto en el apartado anterior, Odoo instala un servidor PostgreSQL, si no lo teníamos previamente ya instalado, y sobre este crea una nueva base de datos y tablas dentro de la base de datos.

Utilizando un gestor de base de datos como pgAdmin, podemos explorar las diferentes tablas dentro de la base de datos creada tal y como vemos en la imagen.



Tablas de la base datos

Fuente: Elaboración propia

Son muchas las tablas que tenemos dentro de nuestra base de datos y no es el objetivo analizar una a una todas las tablas. De forma general observamos:

- Las tablas suelen incluir un prefijo relacionado con el módulo.
- Las tablas "core" de Odoo incluyen prefijos como `res_`, `ir_`, `mail_`.
- Los módulos instalados añaden sus propias tablas (p. ej., CRM, Ventas, Inventario).

En Odoo, la tabla clave para contactos (clientes, proveedores) es `res_partner`; muchas relaciones se expresan con campos `many2one` que terminan en `_id` (p. ej., `country_id`).

EJEMPLO PRÁCTICO

Trabajas como técnico de aplicaciones en una empresa y te encargan analizar qué usuarios están activos dentro del sistema y comprobar qué permisos tienen asignados, diferenciando entre los que son administradores y los que son simples usuarios. Esta información es clave para preparar una auditoría interna y garantizar que no existen accesos indebidos.

El objetivo es tener un listado claro de qué usuarios están activos y podremos cruzar la información con los permisos asignados desde la propia interfaz de Odoo.

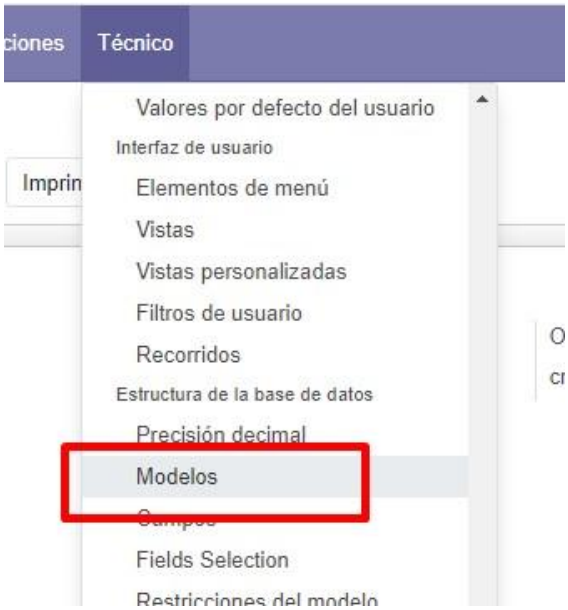
Comenzarás para que en Odoo, los datos principales de los usuarios se almacenen en la tabla `res_users`. Además, los permisos se gestionen a través de los grupos asignados a cada usuario. Desde pgAdmin, se puede lanzar una consulta SQL para obtener la lista de usuarios activos:

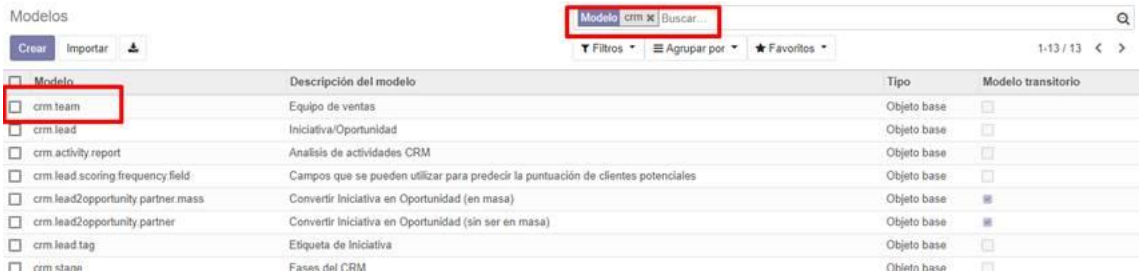
```
SELECT id, login, active
FROM res_users
WHERE active = true;
```

1.3 Modelos de Odoo

Dentro de Odoo, tenemos el concepto de modelos que, a su vez, se relaciona tanto con las bases de datos como con las vistas. Si lo relacionamos con las arquitecturas MVC, es justamente la “M” de modelo.

Accedemos a los modelos definidos a través del menú técnico, tal y como vemos en las imágenes.





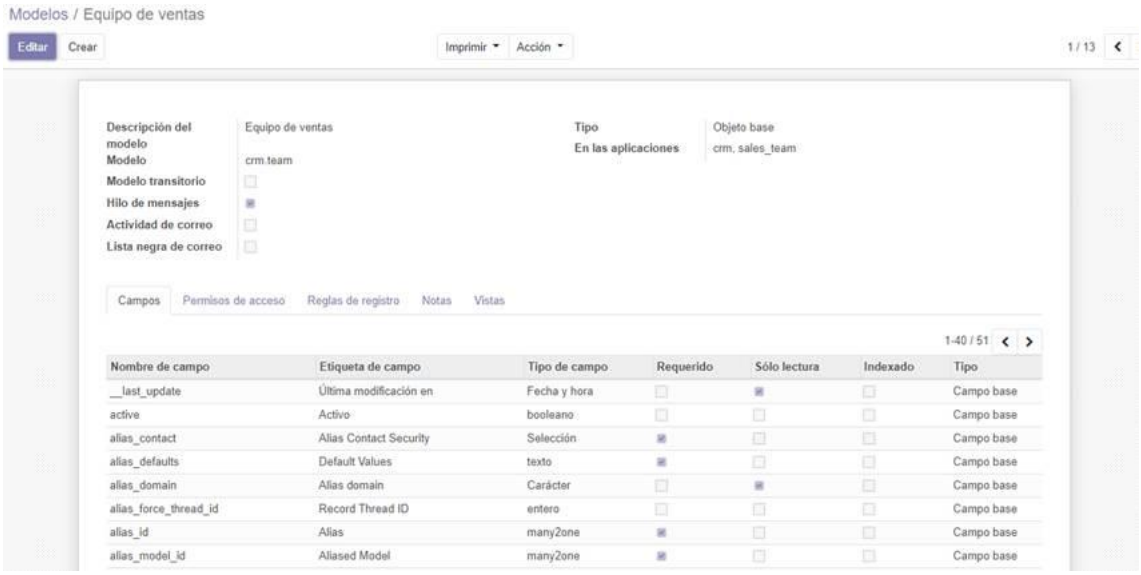
Modelos en Odoo

Fuente: Elaboración propia

VÍDEO DE INTERÉS

Conoce una explicación muy detallada sobre los modelos dentro de Odoo.

Dentro de un modelo nos encontramos con diferentes pestañas y zonas que se explican perfectamente en el anterior vídeo.



Modelos en Odoo. Equipo de ventas

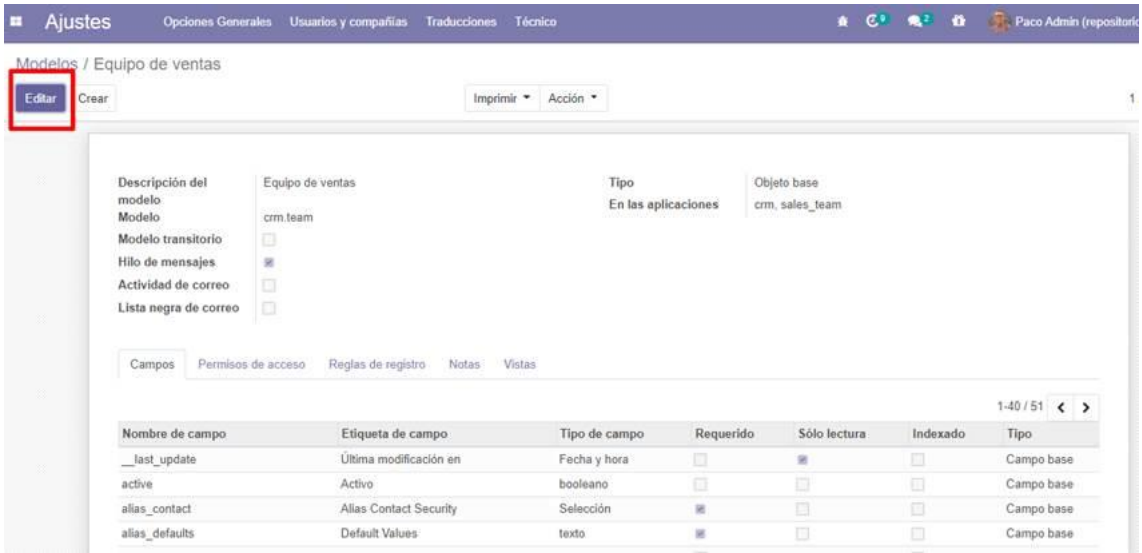
Fuente: Elaboración propia

Analicemos algunas de las opciones:

- Descripción del modelo: palabra o frase corta que trata de explicar para qué sirve, qué hace o qué información guarda esta tabla.
- Modelo: nombre de la tabla.
- Campos:
  - Etiqueta de campo: nombre que "debería" aparecer como cabecera de columna.
  - Tipo de campo: tipo de valor que puede guardar cada campo.
  - Requerido: si lo marcamos, no se puede guardar el registro hasta que no tenga ese campo relleno.
  - Solo lectura: está deshabilitado, no puede editarse.
- Permisos de acceso: permite crear grupos y a cada grupo darle permisos:
  - Para leer.
  - Para escribir.
  - Para crear.
  - Para eliminar.

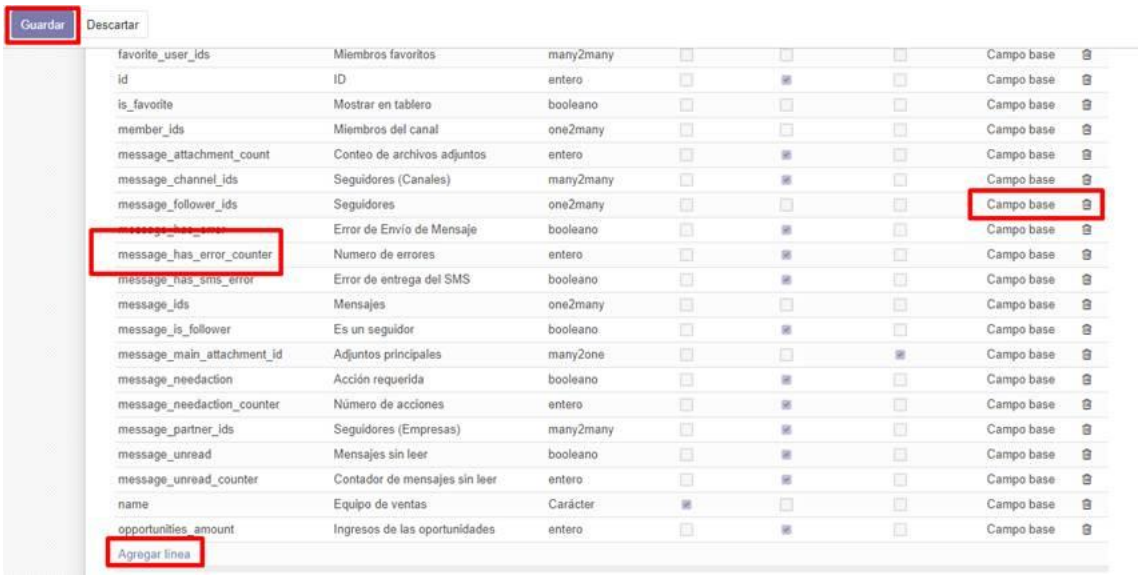
1.4 Definición de campos

Los campos nos permiten almacenar información dentro de un modelo. Tal y como acabamos de ver en el apartado anterior, cada modelo viene con una serie de campos definidos. Modificar o crear un nuevo campo siempre se realizará a través de las opciones de desarrollador.



Edición de modelo

Fuente: Elaboración propia



Guardar | Descartar

favorite_user_ids	Miembros favoritos	many2many				Campo base
id	ID	entero		■		Campo base
is_favorite	Mostrar en tablero	booleano				Campo base
member_ids	Miembros del canal	one2many				Campo base
message_attachment_count	Conteo de archivos adjuntos	entero		■		Campo base
message_channel_ids	Seguidores (Canales)	many2many		■		Campo base
message_follower_ids	Seguidores	one2many				Campo base
[illegible]	Error de Envío de Mensaje	booleano		■		Campo base
message_has_error_counter	Numero de errores	entero		■		Campo base
message_has_sms_error	Error de entrega del SMS	booleano		■		Campo base
message_ids	Mensajes	one2many				Campo base
message_is_follower	Es un seguidor	booleano		■		Campo base
message_main_attachment_id	Adjuntos principales	many2one			■	Campo base
message_needaction	Acción requerida	booleano		■		Campo base
message_needaction_counter	Número de acciones	entero		■		Campo base
message_partner_ids	Seguidores (Empresas)	many2many		■		Campo base
message_unread	Mensajes sin leer	booleano		■		Campo base
message_unread_counter	Contador de mensajes sin leer	entero		■		Campo base
name	Equipo de ventas	Carácter	■			Campo base
opportunities_amount	Ingresos de las oportunidades	entero		■		Campo base

Agregar línea

Nuevo campo

Fuente: Elaboración propia

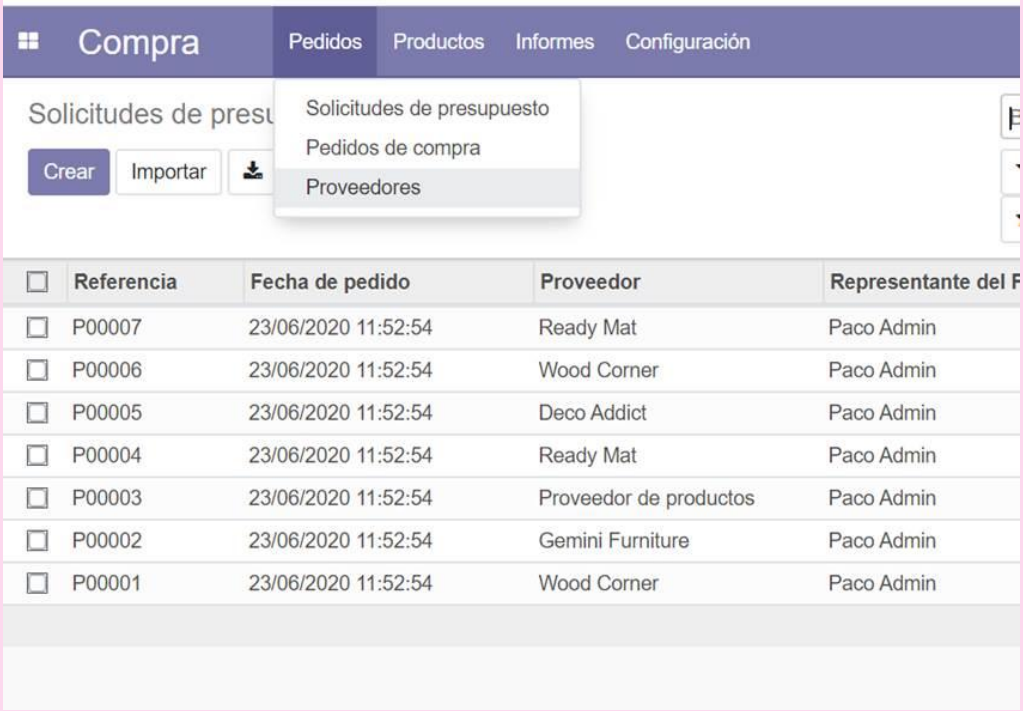
EJEMPLO PRÁCTICO

Como técnico informático especialista en el desarrollo de aplicaciones, vas a encargarte del análisis de funcionalidades de Odoo y su implantación en los almacenes de alimentación donde se pretende introducir un nuevo campo a los productos que vengan del proveedor.

Se necesita implementar un campo que determine si el proveedor trabaja o no con la trazabilidad definida en el reglamento 178/2002.

¿Cómo podemos incorporar ese nuevo campo al formulario de proveedor?

1. El primer paso es activar el modo desarrollador.
2. En segundo lugar, iremos al formulario en concreto que queramos añadir el campo nuevo tal y como muestra la figura.



Nuevo campo

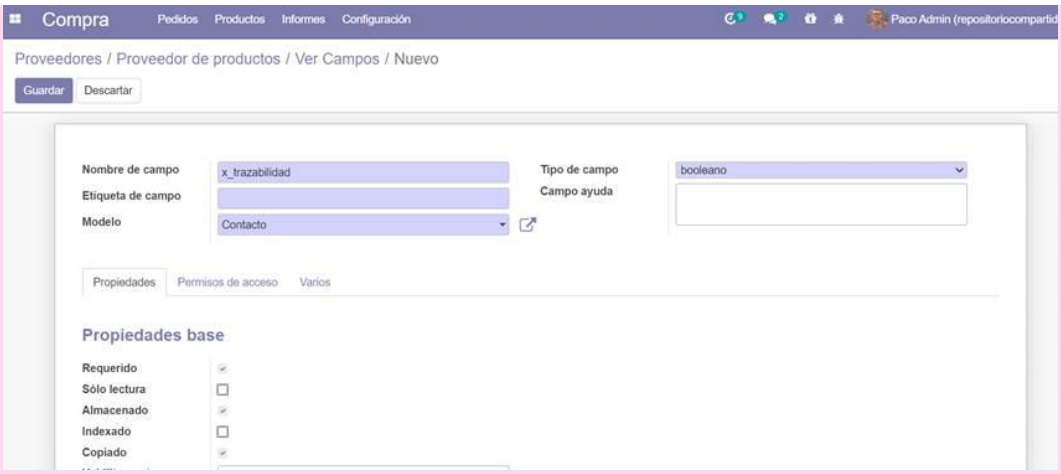
Fuente: Elaboración propia

3. Dentro del proveedor abrir los campos, como vemos en la figura.

Ver campos de proveedor

Fuente: Elaboración propia

4. Crear un nuevo campo con el nombre x_trazabilidad, de tipo booleano y requerido, tal y como muestra la imagen.



Nuevo campo de proveedor

Fuente: Elaboración propia

VÍDEO DE INTERÉS

En este vídeo podrás ver una explicación detallada de cómo generar un nuevo campo dentro del modelo.

1.5 Consultas de acceso a datos

Para realizar consultas sobre las tablas de una base de datos utilizamos el lenguaje SQL. Aunque ya se vio en el curso anterior, vamos a hacer un repaso sobre este lenguaje.

El lenguaje SQL consta de una treintena de sentencias, cada una de las cuales solicita una acción específica por parte del SGBD tal como la creación de una nueva tabla, la recuperación de datos o la inserción de nuevos datos en la base de datos.

SENTENCIA	DESCRIPCIÓN
Manipulación de datos	
SELECT	Permite recuperar datos de la BD
INSERT	Permite añadir nuevas filas de datos de la BD
DELETE	Permite suprimir filas de la BD
UPDATE	Permite modificar datos existentes en la BD
Definición de datos	
CREATE TABLE	Permite añadir una base de datos a la BD
DROP TABLE	Permite suprimir una tabla de la BD
ALTER TABLE	Permite modificar la estructura de una tabla existente
CREATE VIEW	Permite añadir una nueva vista a la BD
DROP VIEW	Permite suprimir una lista de la BD
CREATE INDEX	Permite construir un índice para una columna
DROP INDEX	Permite suprimir el índice para una columna
CREATESYNONUM	Permite definir un alias para un nombre de tabla
DROP SYNONUM	Permite suprimir un alias para un nombre de tabla
COMMENT	Permite definir comentarios para una tabla
LABEL	Permite definir el título de una columna
Control de acceso	
GRANT	Permite conceder privilegios de acceso a usuarios
REVOKE	Permite suprimir privilegios de acceso a usuarios
Control de transacciones	
COMMIT	Permite finalizar la transacción actual
ROLLBACK	Permite abortar la transacción actual

Las sentencias SQL, independientemente de la utilización, deben comenzar con un verbo seguido de una palabra clave que haga referencia a lo que la sentencia hace. Estas sentencias pueden ser CREATE, INSERT, UPDATE, etc.

A continuación, la sentencia incorpora una o más cláusulas que permiten, por ejemplo, indicar los datos sobre los que se va a ejecutar la sentencia. Estas cláusulas, que algunas son obligatorias y otras opcionales, empiezan también siempre con una palabra clave como WHERE, INTO, FROM, etc.

La cláusula SELECT especifica los ítems de datos a recuperar por la consulta. Los ítems se especifican generalmente mediante una lista de selección, separados por comas.

Cada ítem de selección de la lista genera una única columna de resultados de consulta, en orden de izquierda a derecha.

SELECT item1, item2, ... FROM table_name

Un ítem de selección puede ser:

- El nombre de uno de los campos de la tabla.
- Un valor constante.
- Una expresión SQL.

La palabra reservada FROM nos permite, por otro lado, seleccionar la tabla sobre la cual se filtrará y extraerá la información.

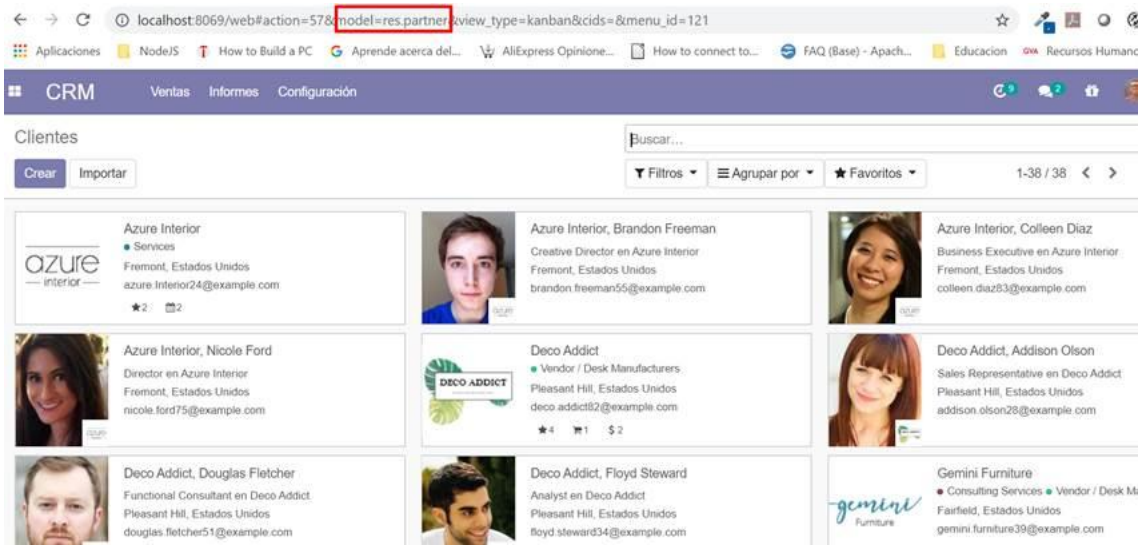
En los casos de realizar consultas directamente a la base de datos de Odoo, debe evitarse utilizar `UPDATE/DELETE` directos, excepto en scripts de migración controlados. Para extracción analítica, es preferible usar vistas de solo lectura o el API del ORM.

ARTÍCULO DE INTERÉS

Accede, en este enlace, una serie de artículos tutoriales sobre SQL.

1.5.1 Ejemplo datos de cliente

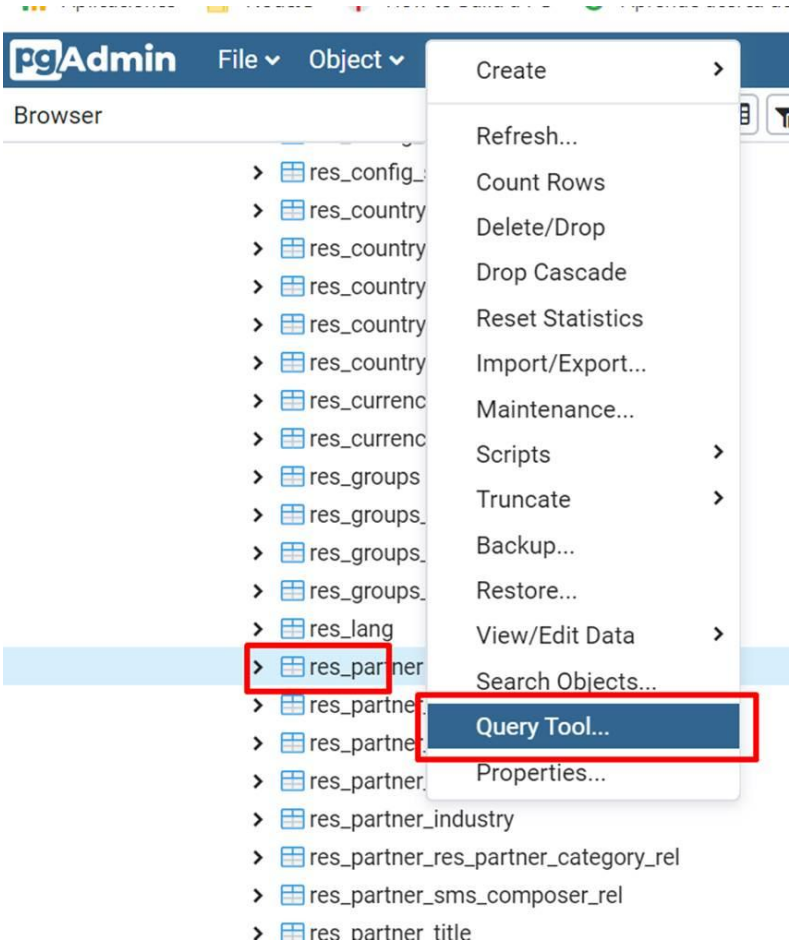
Odoo tiene una base de datos potente y extensa donde se almacena la información de los diferentes módulos sobre los que trabajamos. Una de esas tablas es la de clientes dentro del módulo de CRM. Si observamos cuando estamos en la vista de clientes, la URL nos da una pista de cuál es la tabla que se está utilizando.



Lista de clientes

Fuente: Elaboración propia

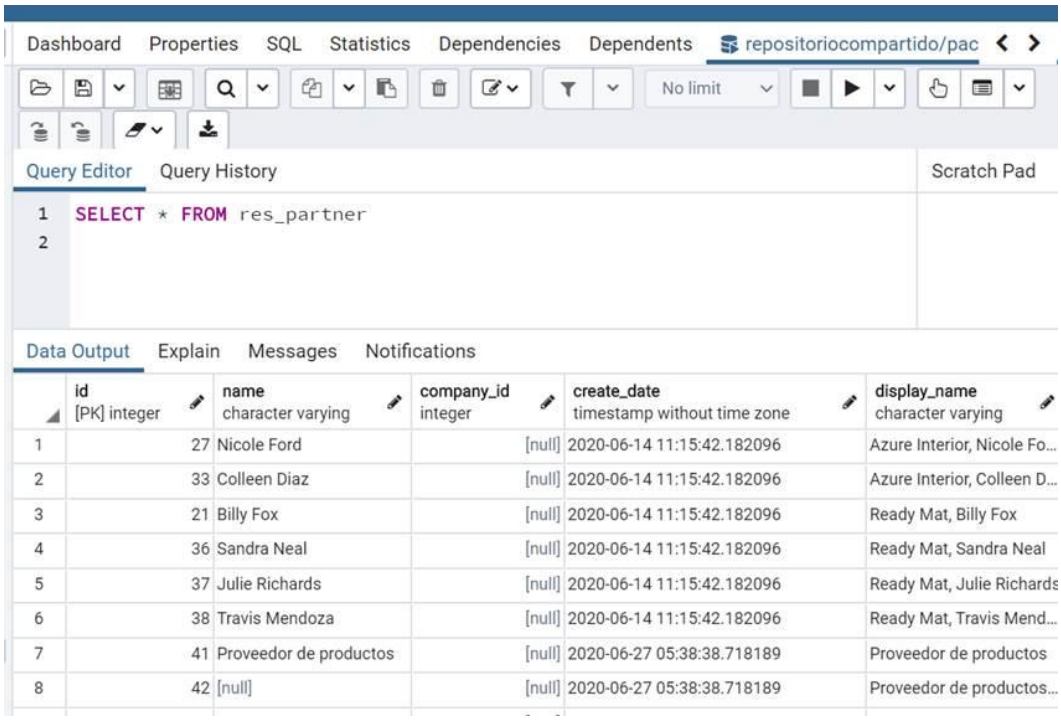
De esta forma, a través de pgAdmin o cualquier herramienta de administración de PostgreSQL, podemos lanzar la consulta de todos los clientes y sus datos.



Query tool en pgAdmin

Fuente: Elaboración propia

A partir de la herramienta QueryTool, lanzamos la siguiente consulta que vemos en la imagen, obteniendo todos los datos de los clientes.



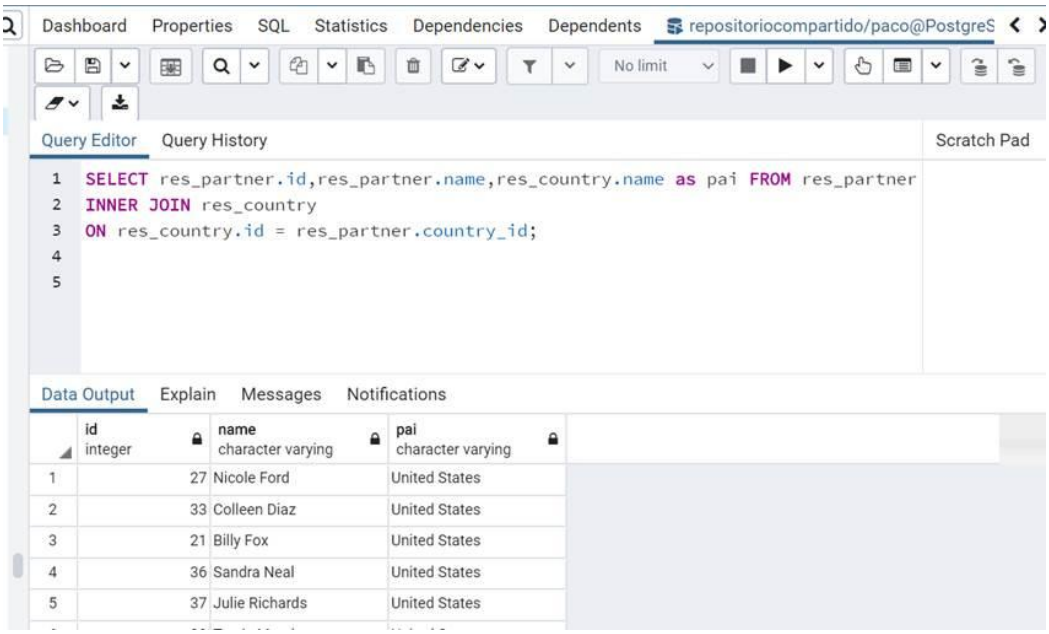
	id [PK] integer	name character varying	company_id integer	create_date timestamp without time zone	display_name character varying
1	27	Nicole Ford	[null]	2020-06-14 11:15:42.182096	Azure Interior, Nicole Fo...
2	33	Colleen Diaz	[null]	2020-06-14 11:15:42.182096	Azure Interior, Colleen D...
3	21	Billy Fox	[null]	2020-06-14 11:15:42.182096	Ready Mat, Billy Fox
4	36	Sandra Neal	[null]	2020-06-14 11:15:42.182096	Ready Mat, Sandra Neal
5	37	Julie Richards	[null]	2020-06-14 11:15:42.182096	Ready Mat, Julie Richards
6	38	Travis Mendoza	[null]	2020-06-14 11:15:42.182096	Ready Mat, Travis Mend...
7	41	Proveedor de productos	[null]	2020-06-27 05:38:38.718189	Proveedor de productos
8	42	[null]	[null]	2020-06-27 05:38:38.718189	Proveedor de productos...

Consulta SQL en pgAdmin
Fuente: Elaboración propia

1.5.2 Ejemplo datos de cliente combinados

Como hemos dicho, Odoo posee una estructura de datos e información muy extensa. Muchos de los datos se encuentran relacionados entre tablas a través de diferentes claves.

En el siguiente ejemplo usaremos la tabla res_country para complementar la información de un cliente y, de esa forma, extraer los datos del nombre de un país para un determinado cliente, como vemos en la siguiente figura.



Consulta combinada SQL en pgAdmin

Fuente: Elaboración propia

La consulta utilizada en concreto es:

```
SELECT res_partner.id,res_partner.name,res_country.name as pais FROM
res_partner
INNER JOIN res_country
ON res_country.id = res_partner.country_id;
```

VÍDEO DE INTERÉS

Visualiza un ejemplo de acceso a las tablas y consultas combinadas.

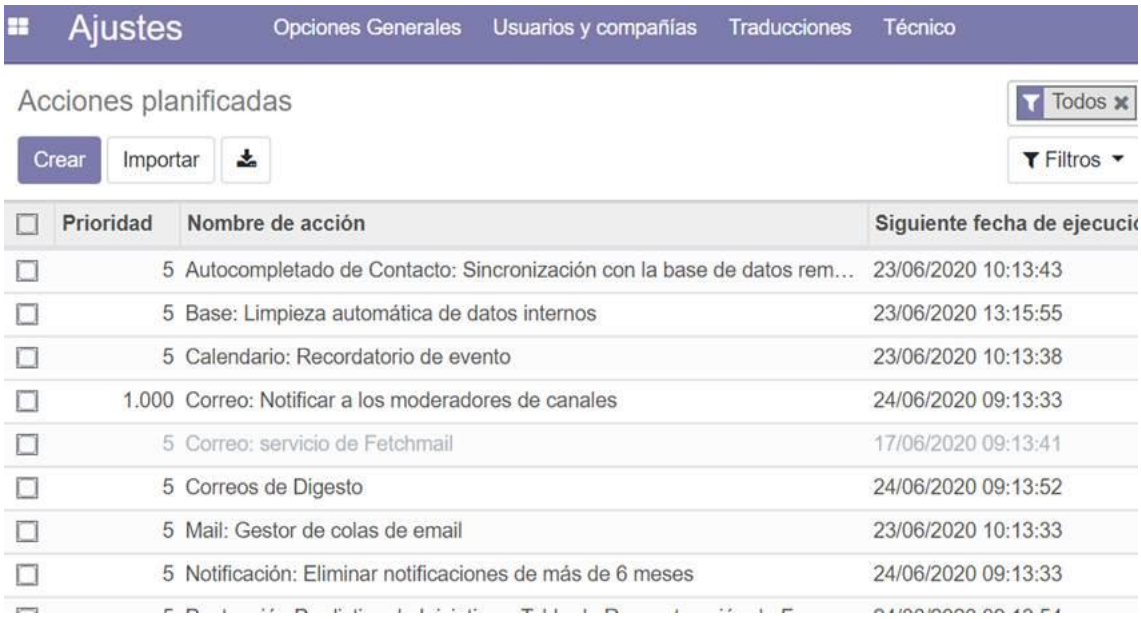
1.6 Procedimientos almacenados de servidor

Los procedimientos almacenados nos permiten realizar una serie de acciones programadas. Al tener dos servidores, Odoo y PostgreSQL, tenemos la opción de programar dos tipos de acciones en servidor:

- Aquellas que tengan que ver con el servidor Odoo. Las tareas programadas se gestionan como Acciones planificadas (cron) desde Ajustes ⍰ Técnico (modo desarrollador).

- Aquellas que tengan que ver únicamente con los datos. En PostgreSQL, la lógica de negocio suele residir en módulos Odoo (Python/ORM), dejando la base para integridad y rendimiento (índices, constraints).

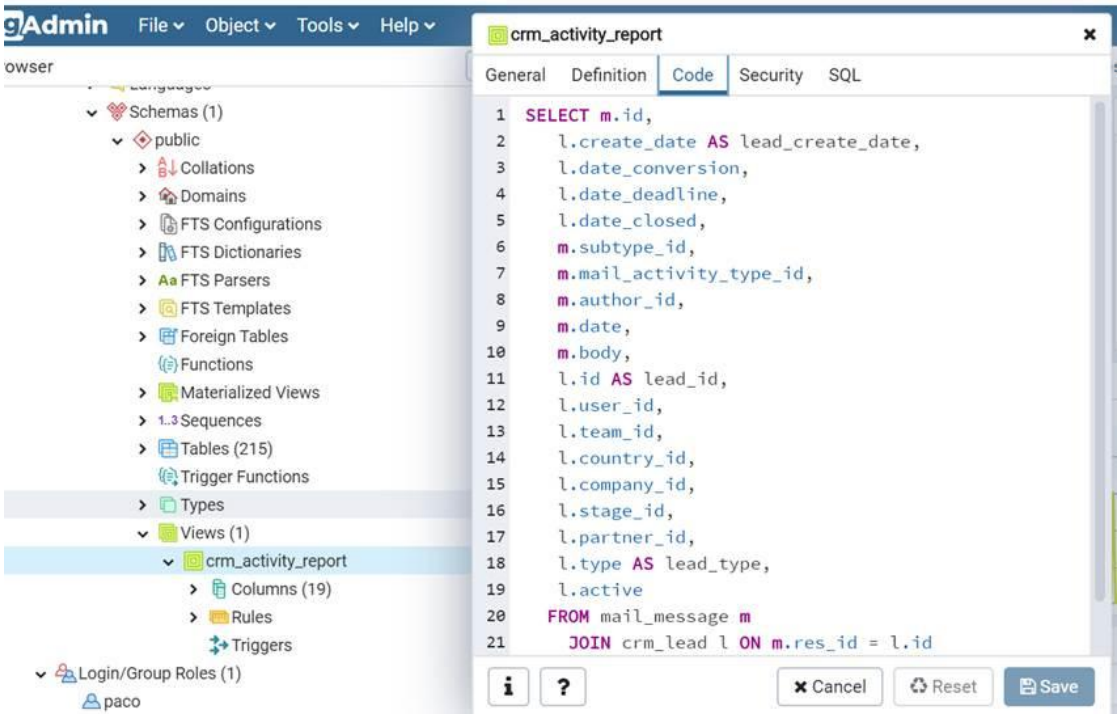
En el caso de procedimientos almacenados dentro del servidor Odoo, encontraremos las actualmente definidas para poderlas editar o crear nuevas a través de *Automatización -> Acciones planificadas*, tal y como se muestra en la imagen.



Procedimientos almacenados en el servidor

Fuente: Elaboración propia

Para el caso de los procedimientos almacenados en el servidor PostgreSQL, existen varios mecanismos para programar procedimientos y vistas. En la imagen se muestra una de esas vistas generadas en la instalación.



Procedimientos almacenados en la base de datos

Fuente: Elaboración propia

EJEMPLO PRÁCTICO

En la empresa de logística donde trabajas como técnico de aplicaciones, se está implantando Odoo, la dirección quiere optimizar los procesos de reposición. Solicitan que cada noche, de forma automática, se genere un listado de productos cuyo stock esté por debajo del mínimo establecido para que el equipo de compras lo revise al día siguiente.

¿Cómo implementarías la generación del listado?

En Odoo, este tipo de tareas programadas tendrás que configurarla desde Ajustes → Técnico → Acciones planificadas.

Como técnico, debes crear una nueva acción planificada que ejecute la consulta sobre el stock de productos, filtrando aquellos con cantidades menores al mínimo configurado.

Deberás programar la tarea para que se ejecute cada noche a las 00:00 y genere un listado o notificación interna para el equipo de compras. Garantizando el proceso sea automático y que los pedidos de reposición se gestionen de manera ágil.

1.7 Cálculos de pedidos, albaranes, facturas, asientos predefinidos, trazabilidad, producción, entre otros

Son muchas las funciones que nos permiten realizar los sistemas de gestión empresarial de entre las que podemos destacar: contabilidad, operaciones de compra y venta, trazabilidad, etc.

Cuando instalamos Odoo, instalando los módulos de Compras, Ventas y CRM, se instalan de base muchos módulos que nos resultarán necesarios para la gestión de una empresa.

Flujo de trabajo en Facturación/Contabilidad
Fuente: Elaboración propia

1.7.1 Pedidos y Albaranes

Dentro de los diferentes flujos de trabajo nos encontramos con diferentes documentos, dentro de las compras y ventas tenemos diferentes documentos que son reflejo los unos de los otros dependiendo de si estamos hablando de clientes o de proveedores:

- Dentro de una empresa, la compra de productos se refiere a la interacción con los diferentes proveedores y sus productos. En este flujo de proceso nos encontramos:
    - Solicitud de presupuesto, el cual es un documento sobre cálculo económico de una serie de productos bajo las condiciones pactadas con el proveedor.
    - Pedidos de compra, que se trata de un documento de pedido de compra de los productos, bien sobre una solicitud de presupuestos, bien sin este.

- La venta dentro de una empresa es uno de los procesos más importantes, ya que alrededor de este proceso giran el resto de procesos. En las ventas nos encontramos con:
    - Presupuestos. En este caso es el cliente el que nos solicita un documento con una aproximación del cálculo para una serie de productos y de servicios.
    - Pedido. Al igual que ocurría con los de compra, el cliente nos solicita ya en firme un conjunto de productos y de servicios que pueden provenir de un presupuesto o no.

Si el flujo es completo, se genera un presupuesto que se convierte en un pedido, tanto de compra como de venta, y a su vez estos generan el albarán de venta, que para el flujo de logística es imprescindible ya que es el documento que proporciona el seguimiento de la venta y entrega al cliente. En el caso del albarán de compra, no somos nosotros quienes generamos dicho documento, sino que es el proveedor quien nos genera dicho documento.

ARTÍCULO DE INTERÉS

Lee este interesante artículo sobre las diferencias entre albarán y factura.

1.7.2 Facturación

Odoo permite varias combinaciones de creación de facturas desde pedido o albarán. Un pedido con varios albaranes, un albarán con una factura, un albarán con varias facturas, etc.

Además, se pueden configurar todas las formas de cobro o pago que utilice la empresa (giros, pagarés, transferencia…):

- Configuración de distintas formas de pago de clientes o cobro de proveedores.
- Generación de facturas automáticas desde pedido o albarán.

- Generación automática de efectos de cobro y pago.
- Generación de remesas de recibos, órdenes de pago o transferencia.
- Importación de extractos del banco.
- Envío telemático de remesas bancarias.
- Gestión de bancos propios de la empresa, bancos de clientes y de proveedores.

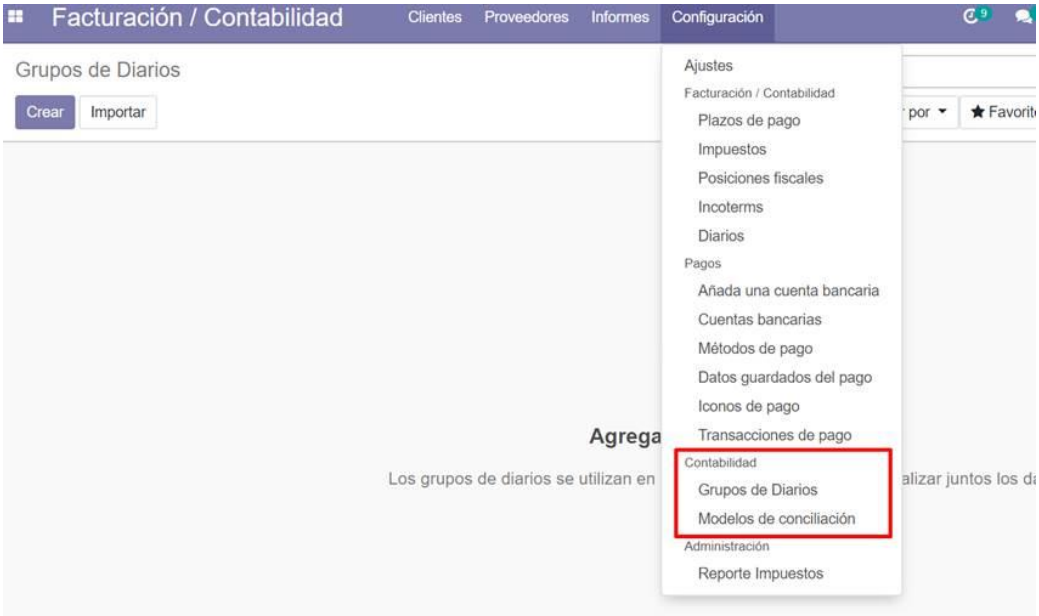
1.7.3 Contabilidad

La contabilidad en Odoo se encuentra dentro del módulo Facturación/Contabilidad y permite múltiples divisiones de la compañía y múltiples compañías, así como varios idiomas y monedas.

La mayoría de los asientos contables son creados de forma automática por el sistema, por lo tanto, se evita la introducción manual de datos y los errores que esto conlleva.

Odoo permite gestionar la contabilidad analítica presupuestaria y de costes, así como:

- Definir la estructura del plan de contabilidad.
- Indicar impuestos (IVA, recargo de equivalencia, retenciones IRPF).
- Establecer plazos de pago.
- Definir procesos automáticos para el cierre del año.
- Generar informes anuales de pérdidas y ganancias.



Configuración Contabilidad
Fuente: Elaboración propia

En la imagen vemos una captura de pantalla de la configuración de la contabilidad, pero como podemos apreciar tenemos todas las opciones necesarias para la realización de las transacciones relacionadas con esta.

1.7.4 Operaciones: Fabricación, producción, proyectos y otros

Dentro de lo que se denomina operaciones, encontramos una serie de módulos dentro de Odoo que, aunque no son imprescindibles para todas las empresas, sí que son importantes.

Muchas empresas no realizan producción o almacenaje de producto, por lo que los módulos relacionados con producto y stock no serán imprescindibles. Pero para las empresas que tienen fabricación o tienen almacenamiento y distribución son muy importantes.

El flujo completo de una empresa que realice fabricación podría ser el siguiente (cabe mencionar que la fabricación es uno de los procesos quizá más particulares de una empresa):

- **Planificación de la fabricación.** Para ello, se requiere planificar los recursos humanos, de materia prima y de fabricación. Para esto se requieren los módulos de proyectos y fabricación. Incluiremos aquí los procesos de trazabilidad de producto, si son requeridos por ley o simplemente necesarios para un correcto funcionamiento.

- **Almacenaje.** Se tenga uno o varios almacenes, se necesita planificar el stock tanto de la materia prima, del producto en proceso, el producto finalizado y el producto con pedido a cliente.

- **Logística y distribución.** Una vez realizado un pedido a cliente es necesario realizar una correcta planificación de la distribución de los pedidos. Aquí o, en procesos anteriores, deberemos introducir también el proceso de atención al cliente y devoluciones, ya que impactará en nuestros almacenes.

EJEMPLO PRÁCTICO

El departamento de compras de la empresa en la que trabajas necesita trabajar con información fiable y bien organizada sobre los proveedores y los productos que estos ofrecen. No se trata únicamente de saber a quién se compra, sino de poder analizar condiciones, comparativas de precios o incluso medir el volumen de operaciones con cada proveedor.

Como técnico del departamento de informática, serás el encargado de identificar de qué forma Odoo almacena esos datos y cómo podemos extraerlos de la base de datos para transformarlos en informes útiles. En este caso, la cuestión pasa por localizar en qué tabla se encuentran los datos principales de los proveedores. Una vez identificada, podremos crear consultas SQL que nos permitan generar informes claros y adaptados a las necesidades del área de compras, mejorando la capacidad de negociación y la toma de decisiones estratégicas.

¿En qué tabla encontramos la información principal de proveedor?

1. Si vamos al listado de proveedores, vemos que se está usando la misma tabla que la de clientes.

Listado de proveedores
Fuente: Elaboración propia

2. En este caso, para solo elegir los proveedores, deberemos usar el campo supplier_rank>0

```
SELECT *
FROM res_partner
WHERE supplier_rank>0
```

2. INTERFACES DE ENTRADA DATOS, PROCESOS Y LISTADOS. FORMULARIOS

Llega la hora en la que los usuarios van a comenzar a introducir información en Odoo, por lo que deberás analizar qué formularios, listados o informes se adaptan mejor a sus necesidades y proponer posibles mejoras, de los que ofrece Odoo.

Dentro de cualquier aplicación, y en concreto cuando hablamos de un ERP, la interacción con el usuario es importantísima, hasta el punto de que una implantación de cualquier ERP puede no tener éxito porque no se tengan bien definidos los procesos y la interacción con el usuario final.

Podemos definir dos grandes apartados dentro de la interacción con los usuarios cuando definimos la implantación de un ERP:

- La introducción de datos, que primordialmente será a través de formularios tal y como veremos, pero que puede llegar a través de otros muchos mecanismos y que se escapan al ámbito de este módulo. Hablamos de la entrada de datos automática desde otros sistemas o la recogida de información desde sensores, entre muchas opciones.

- Los procesos relacionados, que se definen como las restricciones y pasos que se deben seguir para introducir la información dentro del sistema.

Acabamos de introducir y hablar de los procesos de entrada de datos de cara al usuario. Tan importante o más, es el apartado de procesos de salida o listados de información. En este gran apartado nos encontramos con:

- Las vistas o listados integrados en el sistema.
- Los informes, también integrados o informes externos, a partir de la información almacenada en la base de datos.

VÍDEO DE INTERÉS

Visualiza un resumen de las diferentes tipologías de interfaces que nos encontramos dentro de Odoo.

2.1 Formularios

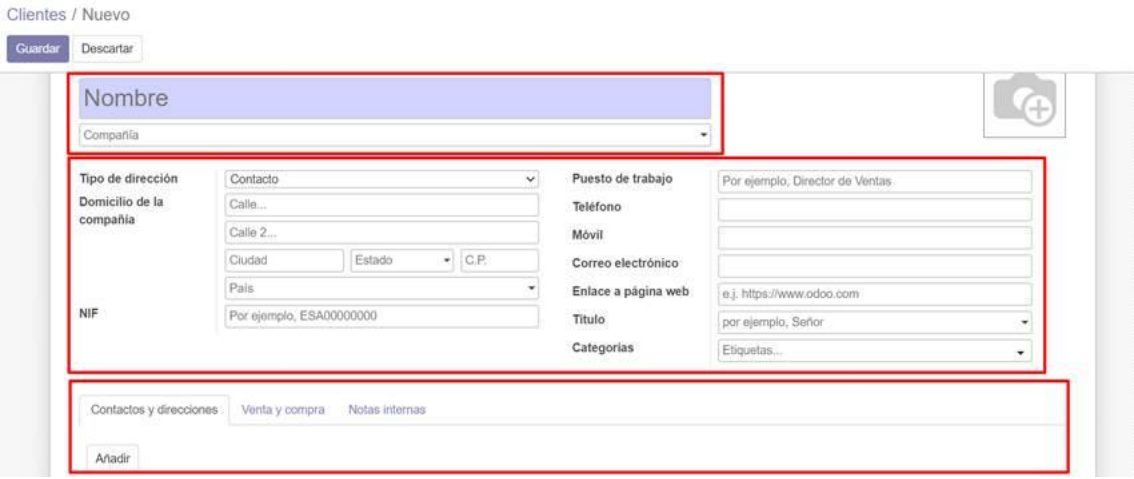
Los formularios dentro de una aplicación cliente-servidor como es Odoo es el mecanismo principal de interacción con el usuario, ya que permite introducir información en el sistema de acuerdo con las definiciones establecidas.

De forma general nos encontramos con 3 tipos de formularios para una determinada tabla o maestro:

- Dar de alta, permitiendo crear un nuevo elemento.
- Modificar datos.
- Eliminar elemento.

Un ejemplo dentro del módulo de CRM que hemos instalado en la unidad 2 es el de creación de un nuevo cliente a través del menú *CRM-> Ventas-> Clientes-> Nuevo* y que, tal y como vemos en la imagen, es el claro ejemplo de formulario donde nos encontramos las siguientes partes:

- Identificado único del maestro, en este caso representado por el nombre de la compañía. Es cierto que puede haber duplicidades y que internamente se resuelven mediante identificadores numéricos.
- Cabecera de información, donde nos encontramos con la información más importante relacionada con el maestro.
- Información auxiliar, donde se define información propia del maestro o información relacionada con otros maestros.



Formulario de clientes
Fuente: Elaboración propia

Como recomendaciones técnicas a tener en cuenta, tenemos:

- Validación, marca campos críticos como requeridos y añade restricciones (Python/SQL) en el módulo cuando el negocio lo exija.
- Seguridad, controla permisos con grupos, ACL y record rules, evita "campo editable" si el usuario no debe alterarlo.
- Flujos, muchos formularios incorporan "estados" (por ejemplo, presupuesto → pedido) y acciones servidor (cron/automatizaciones) que orquestan el proceso.
- Para localizar modelos y campos desde la interfaz, activa el modo desarrollador (Ajustes → Herramientas de desarrollador → Activar) o añade `?debug=1` a la URL.

2.2 Informes y listados de la aplicación

Dentro de Odoo nos encontramos con múltiples opciones para la visualización de información:

- **Vistas.** Son muy parecidos a los formularios solo que en este caso nos muestra información almacenada en la base de datos y relacionada con uno o varios módulos. Las vistas pueden ser de múltiples formatos y tipos, como los de tipo Kanban que mostramos a continuación en la imagen de lista de clientes, o de tipo gráfico, pivot, calendario, etc. Son configurables y sirven como "listados" interactivos.

- **Informes**. Los informes son de tipo report dentro de la nomenclatura Odoo y permiten definir documentos que serán imprimidos o descargados por el usuario. En el caso de documentos imprimibles, usa reportes QWeb (PDF) o la hoja de cálculo integrada para cuadros de mando.

Odoo a lo largo de sus versiones ha ido evolucionando también en las posibilidades de creación y desarrollo de informes:

- Tenemos herramientas sencillas, integradas, como el diseñador de presupuestos dentro de un módulo en concreto.

- Tenemos un lenguaje para las vistas denominado QWeb, que combina HTML, tags y Python.

ENLACE DE INTERÉS

Lee una pequeña introducción sobre QWeb.

2.2.1 Diseñador integrado

Tal y como decíamos, Odoo incorpora un diseñador para algunos de sus informes y listados. Es el caso del Diseño de Presupuestos o el Diseño de Facturas, donde es posible insertar un pivot o gráfico directamente en una Spreadsheet de Odoo ("Insert in Spreadsheet") y reutilizarlo como informe vivo.

Diseñador integrado

Fuente: Elaboración propia

En la imagen, dentro del módulo de Ventas, vemos como tenemos la opción de modificar el diseño de presupuesto que se le envíe a un cliente. Dentro de este diseñador tendremos las opciones de:

- Cambiar el diseño del detalle del presupuesto.
- Cambiar el logo.
- Cambiar la tipografía.
- Cambiar los colores.
- Añadir textos.

Diseñador integrado. Facturas

Fuente: Elaboración propia

VÍDEO DE INTERÉS

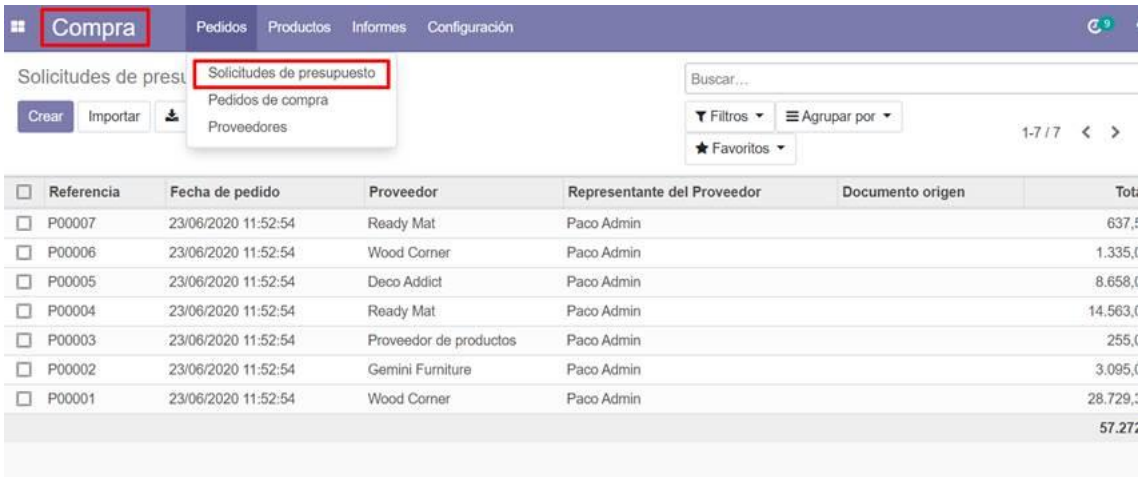
Visualiza en qué consiste el diseñador de informes.

2.2.2 Documentos Odoo

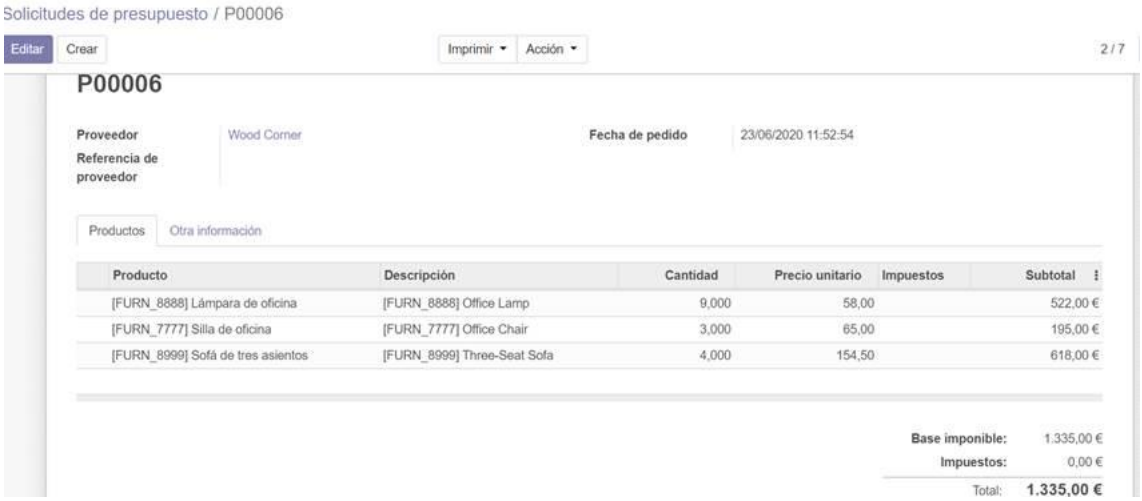
Dentro de Odoo nos encontramos con muchos módulos que producen los documentos comunes y necesarios para el trabajo en un proceso habitual de trabajo.

Recuerda que la numeración y plantillas de documentos (ventas, compras, facturas) se gestionan desde sus respectivos módulos y que el reporte PDF se define con QWeb Reports.

En el caso de compras tenemos los presupuestos y los pedidos de compra. Un presupuesto es un documento que nos presenta un proveedor con un cálculo del detalle de los productos y su precio para una compra en concreto. Dentro del módulo de compras, en el apartado de presupuestos, podemos ver un ejemplo, tal y como muestran en las imágenes.

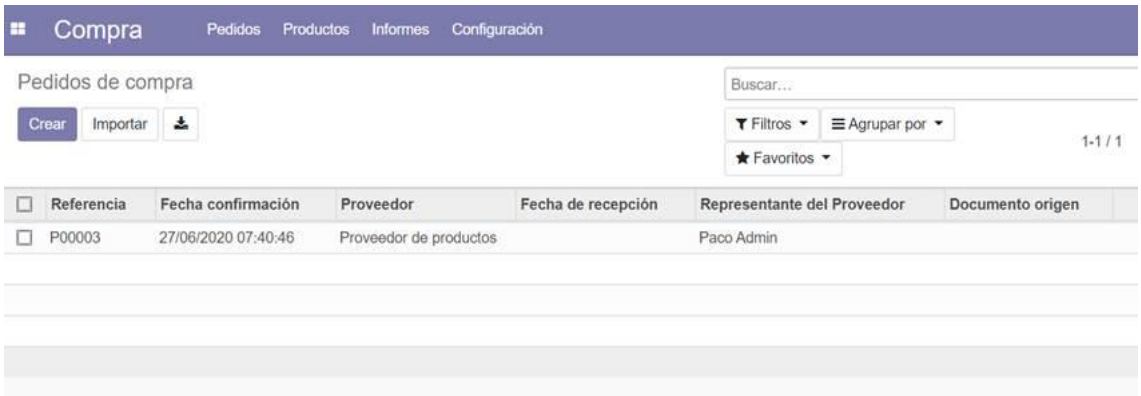


Flujo de compra
Fuente: Elaboración propia



Presupuesto de compra
Fuente: Elaboración propia

Dentro del módulo de compra también podremos ver los pedidos de compra, que son documentos parecidos a los de presupuesto, solo teniendo en cuenta que se han avanzado dentro del flujo de compra por parte de nuestra empresa al proveedor.



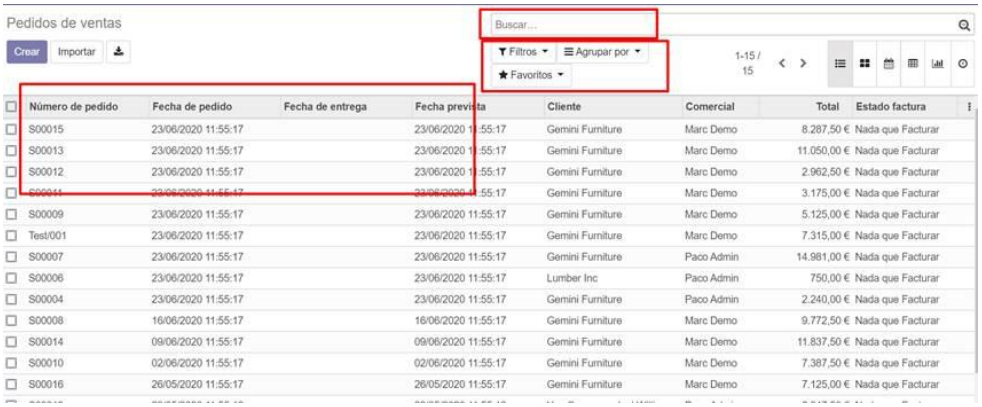
Pedido de compra
Fuente: Elaboración propia

En el caso de las ventas tendremos una correspondencia muy parecida:

- Presupuesto de venta, consiste en un documento que permite presentar al cliente un previo del servicio, su precio final y el detalle.
- Pedidos, es el siguiente paso de un presupuesto de venta donde se formaliza el presupuesto en un pedido de servicio o producto por parte del cliente
- Facturas, es el documento de la empresa y cliente donde se formaliza legalmente la compra por parte del cliente.

2.3 Búsqueda de información

A lo largo de las diferentes ventanas de la aplicación nos encontraremos el mismo sistema homogéneo de presentación de la información y búsqueda de la misma.



Filtrado
Fuente: Elaboración propia

Como vemos en la imagen, en el ejemplo que tenemos de Pedidos de Venta, tenemos hasta 3 tipos de búsquedas, filtrado y ordenación de la información:

- Búsqueda de la información, mediante un buscador de texto que permite realizar la búsqueda contenida en cualquier campo.
- Filtrados que dependerán del módulo y la información que estemos visualizando.
- Agrupaciones que permiten un segundo nivel de filtrado.
- Ordenaciones directas a través de las columnas de visualización de la información.

Indicar que los filtros y agrupaciones se traducen internamente a dominios (`[('campo','=','valor')]`). Las vistas Pivot agregan datos para análisis, en Pivot/Graph puedes Insert in Spreadshee para crear cuadros de mando actualizables.

EJEMPLO PRÁCTICO

En la empresa de alimentación que trabajas como técnico de aplicaciones, se generan diariamente decenas de pedidos de venta. El responsable de logística necesita localizar con rapidez todos los pedidos pendientes de entrega de un cliente concreto, pero también quiere poder agruparlos por fecha para organizar las rutas de reparto.

¿Qué pasos deberás seguir?

En Odoo:

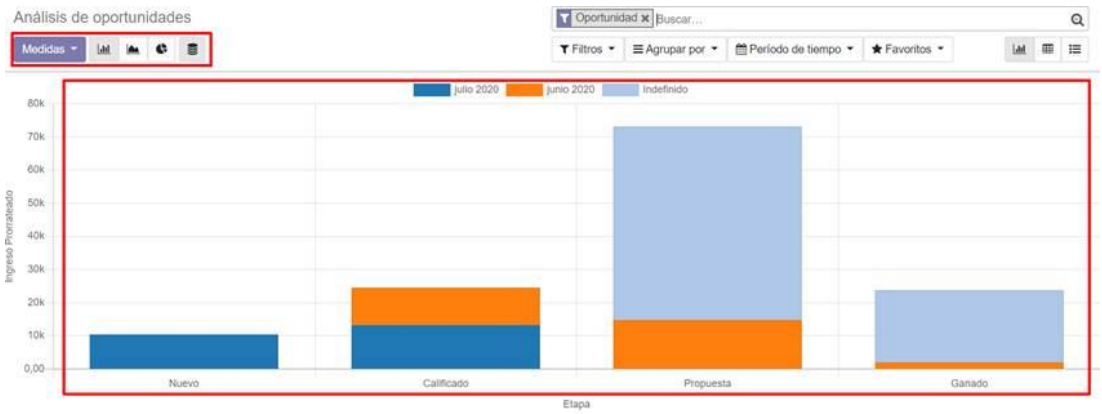
1. Dentro del módulo de Ventas, puedes utilizar el buscador y los filtros avanzados, mostrando al responsable de logística cómo aplicar un filtro por cliente y, a continuación, agrupar los resultados por fecha de pedido.

2. 2. De esta forma, obtendrás un listado dinámico que muestra solo los pedidos pendientes de entrega del cliente elegido, ordenados por día, lo que facilita la planificación de la distribución. Además, este filtro puede guardarse para reutilizarlo en futuras búsquedas.

2.4 Gráficos

Otra forma de presentar la información es a través de gráficos de diferentes tipos. Como podemos ver la imagen, tenemos en todos los gráficos una serie de zonas comunes:

- Medida: nos permite variar la variable de evaluación de la gráfica.
- Tipo de gráfico: nos permite representar la información mediante diferentes tipos de gráficos como barras, acumulados, líneas y otros.
- Gráfico: es la zona central donde aparece la información reflejada.



Ejemplo de gráfico

Fuente: Elaboración propia

EJEMPLO PRÁCTICO

Desde la dirección comercial de la empresa en la que trabajas como técnico de aplicaciones, te solicitan un informe visual para poder comparar rápidamente las ventas mensuales de cada comercial.

El objetivo es tener una representación clara de la evolución de ventas a lo largo del tiempo y poder detectar diferencias significativas en el rendimiento de cada miembro del equipo.

¿Qué pasos deberás seguir para obtener el informe solicitado?

Dentro de Odoo, puedes utilizar la vista de gráfico en el módulo de Ventas. Para este caso, la mejor opción es un gráfico de barras agrupadas por comercial y por mes, ya que permite comparar de manera sencilla la evolución temporal de las ventas de cada persona.

Deberás configurar el gráfico seleccionando “Ventas mensuales” como medida y “Comercial” como criterio de agrupación.

Este informe quedará disponible en el panel de control y podrá actualizarse de forma dinámica en cualquier momento.

2.5 Procesos de extracción de datos en sistemas de ERP-CRM y almacenes de datos. Automatización

Como hemos visto en unidades anteriores, los sistemas de gestión empresarial normalmente constan de la siguiente infraestructura:

- Uno o varios servidores de aplicaciones que permiten la interacción con los usuarios.
- Las vistas de clientes que se ejecutan normalmente con un navegador pero que también puede ejecutarse a través de herramientas propietarias o clientes propietarios.
- Uno o varios servidores de datos donde se almacena la información que gestiona el servidor de aplicaciones.

La extracción de información consiste en la preparación y presentación de la información almacenada en la base de datos. La extracción, por lo tanto, se puede realizar desde varias perspectivas y con diferentes objetivos:

1. Utilizando las vistas, listados, informes y gráficas que proporciona el propio software.
2. Utilizando un generador de informes externo.

Además, la extracción se puede presentar de diversas formas, dependiendo del usuario final y de la finalidad misma del informe:

1. **Por pantalla**. En este caso el mejor mecanismo es el propio software y el mecanismo será a través de vistas, listados o gráficos que pueden ser en muchos casos dinámicos, ya que se pueden cambiar parámetros que, a su vez, modifiquen la presentación.

2. **En pdf, papel o correo**. En este caso se puede utilizar tanto el propio software como softwares externos que nos permitan realizar estas tareas.

2.5.1 Extracción de datos a través de Odoo

A partir del esquema que planteamos Odoo tiene la siguiente infraestructura:

- Tiene un servidor de aplicaciones desarrollado con HTML-JS-Python-CSS.
- Tiene un servidor de datos basado en PostgreSQL.
- Los clientes se ejecutan en un navegador.

Ya hemos visto algunas de los mecanismos a través de los cuales extraemos la información en Odoo:

- Vistas.
- Listados.
- Gráficos.
- Informes.

En la próxima unidad, además, trabajaremos los mecanismos de desarrollo para modificar y crear nuevas vistas e informes.

2.5.2 Extracción de datos a través de un generador de informes externo

En este caso las opciones pasan por atacar directamente a la base de datos, y ya que, en nuestro caso, en Odoo, es una base de datos abierta, podremos gestionar las conexiones.

Los generadores de informes externos pueden ser desde una herramienta ofimática con capacidades para realizar conexiones a bases de datos como Excel, Access u otras, hasta herramientas específicas.

Dentro de las herramientas específicas podemos destacar dos herramientas, una propietaria y otra con una versión community:

- **Crystal Reports:** Es una aplicación propietaria comprada por la empresa SAP y que nos permite la posibilidad de generar y servir informes mediante una arquitectura cliente-servidor.

- **Jasper Reports:** Es una herramienta que tiene una versión community y es la que introduciremos y utilizaremos para realizar informes específicos.

Alternativas modernas de BI conectando directamente PostgreSQL: Power BI (conector PostgreSQL vía Npgsql), Metabase y Apache Superset.

ENLACE DE INTERÉS

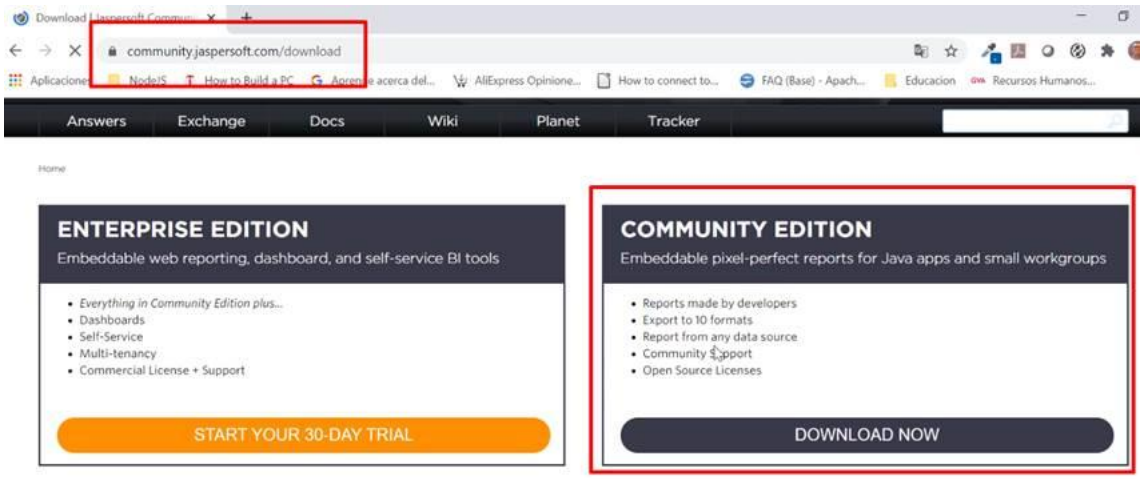
Accede a la web del proveedor.

2.5.3 Instalación de Jaspersoft Studio

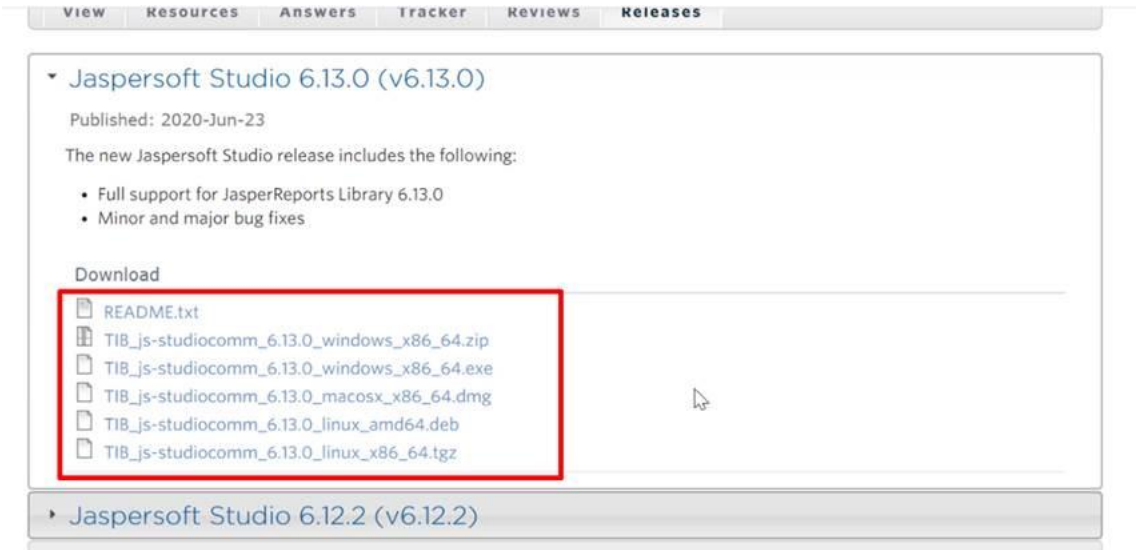
Dentro de Jaspersoft nos encontramos con diferentes softwares. En concreto, nosotros trabajaremos con Jaspersoft Studio, que se propone como el sustituto de Jaspersoft Reports que aún es utilizable.

En la actualidad está ya disponible la versión 7.0.3, muy similar a la versión 6.13.0 que es la que vamos a ver los pasos de en la instalación, que pasan por:

1. Registro en JasperSoft
2. Descarga de la versión adecuada
3. Instalación

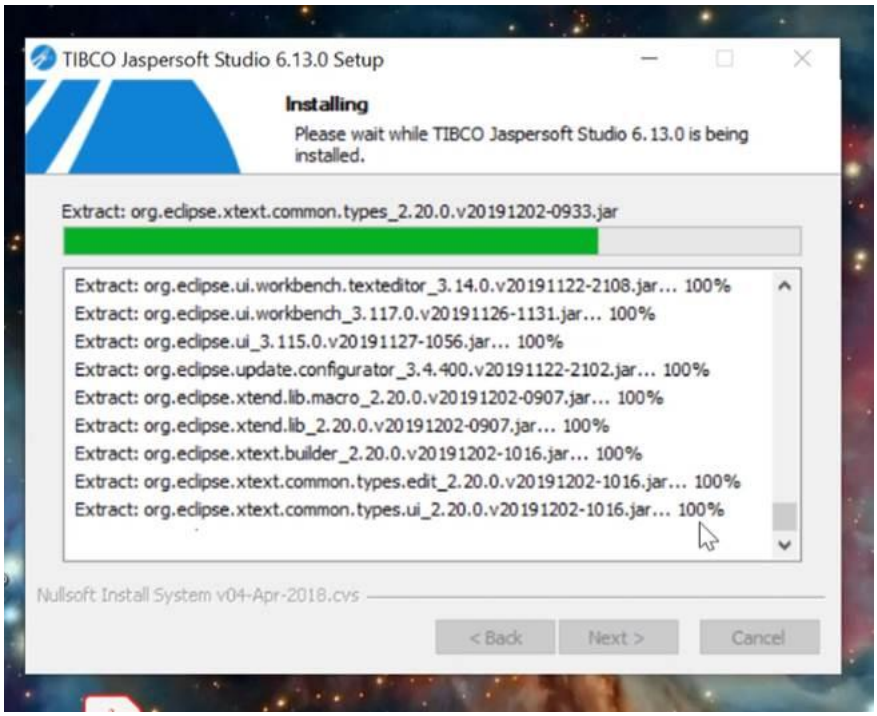


Descarga e instalación de Jaspersoft Studio
Fuente: Elaboración propia



Selección de la versión

Fuente: Elaboración propia



Proceso de instalación

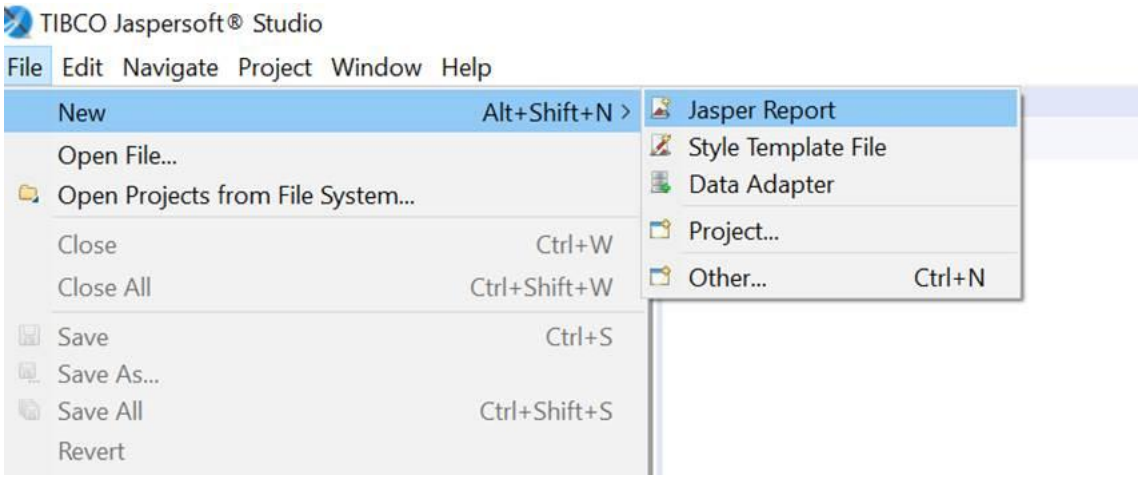
Fuente: Elaboración propia

VÍDEO DE INTERÉS

Visualiza un resumen del paso a paso de la instalación de Jaspersoft Studio para Windows.

2.5.4 Primer report con Jaspersoft Studio

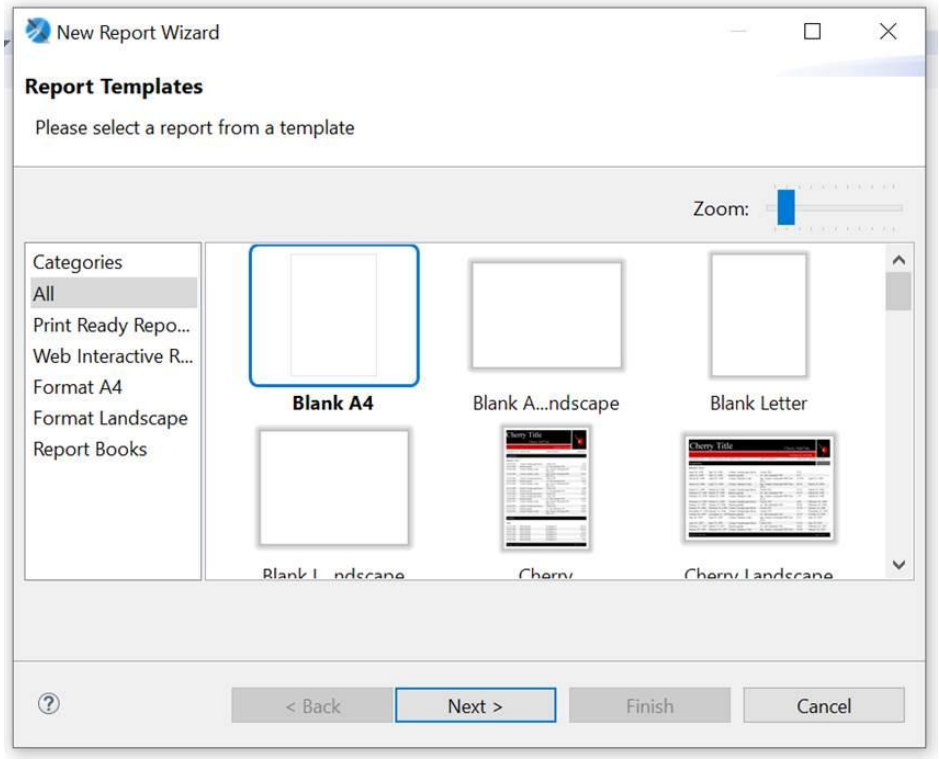
Para realizar un primer report tendremos que, en primer lugar, hacer clic sobre File -> New -> Jasper Report como vemos en la imagen:



Nuevo informe

Fuente: Elaboración propia

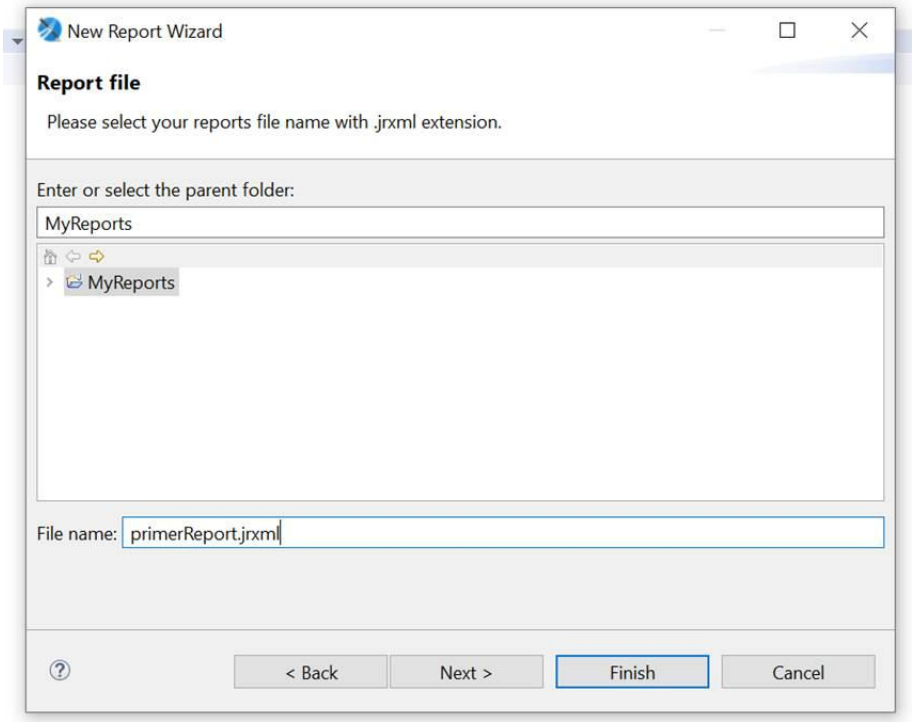
Elegiremos un nuevo report en blanco, tal y como vemos en la imagen:



Informe en blanco

Fuente: Elaboración propia

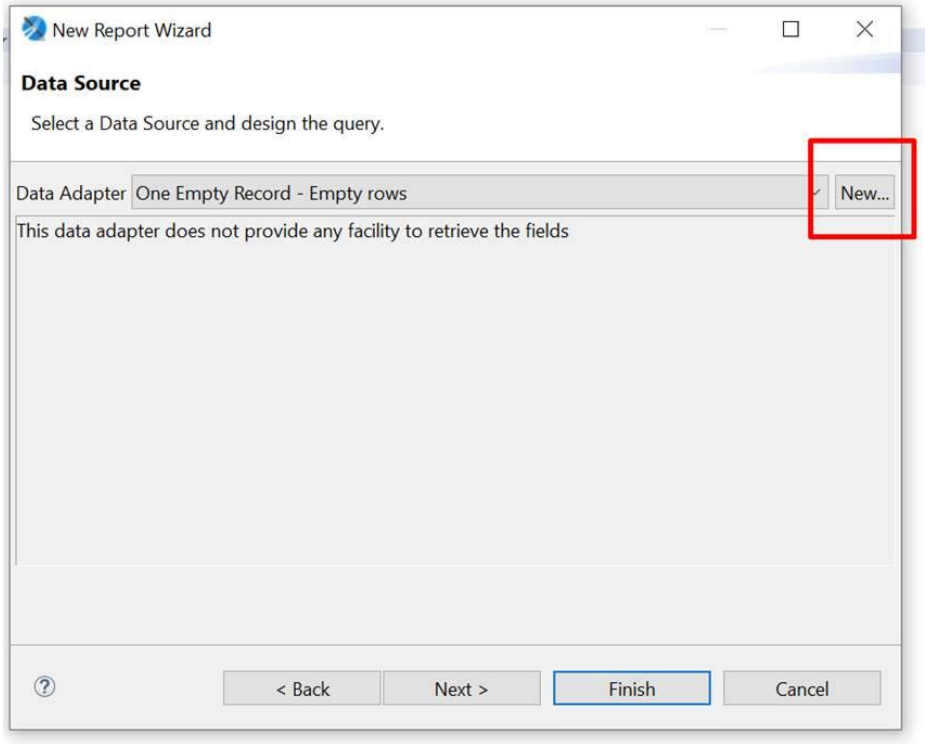
Le daremos un nombre, por ejemplo, “primerReport.jrxml”:



Almacenamiento del informe

Fuente: Elaboración propia

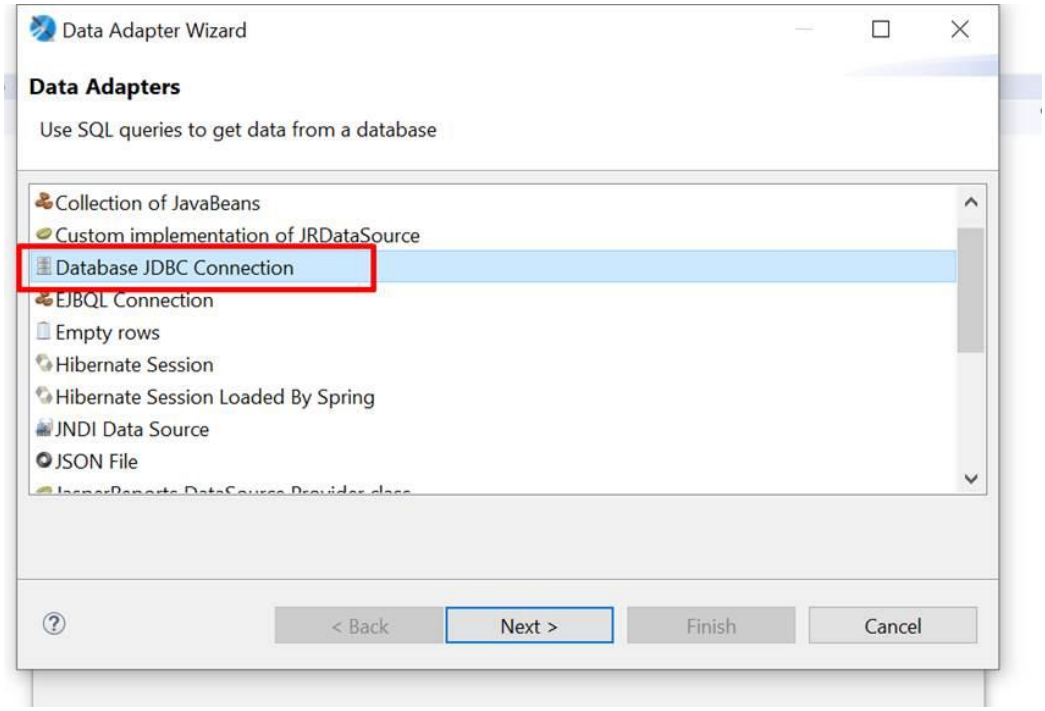
Al ser la primera vez que nos conectamos a la base de datos, tendremos que crear una nueva conexión con la base de datos:



Nuevo conector

Fuente: Elaboración propia

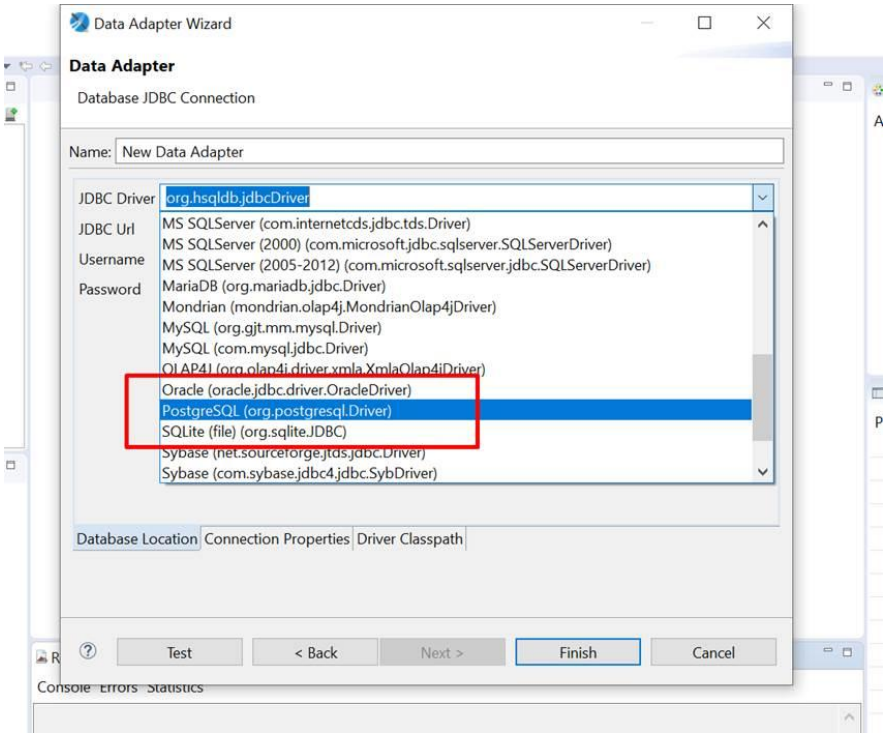
Elegiremos la conexión JDBC:



Selección del tipo de conector

Fuente: Elaboración propia

Y dentro de esta el conector JDBC específico para PostgreSQL:

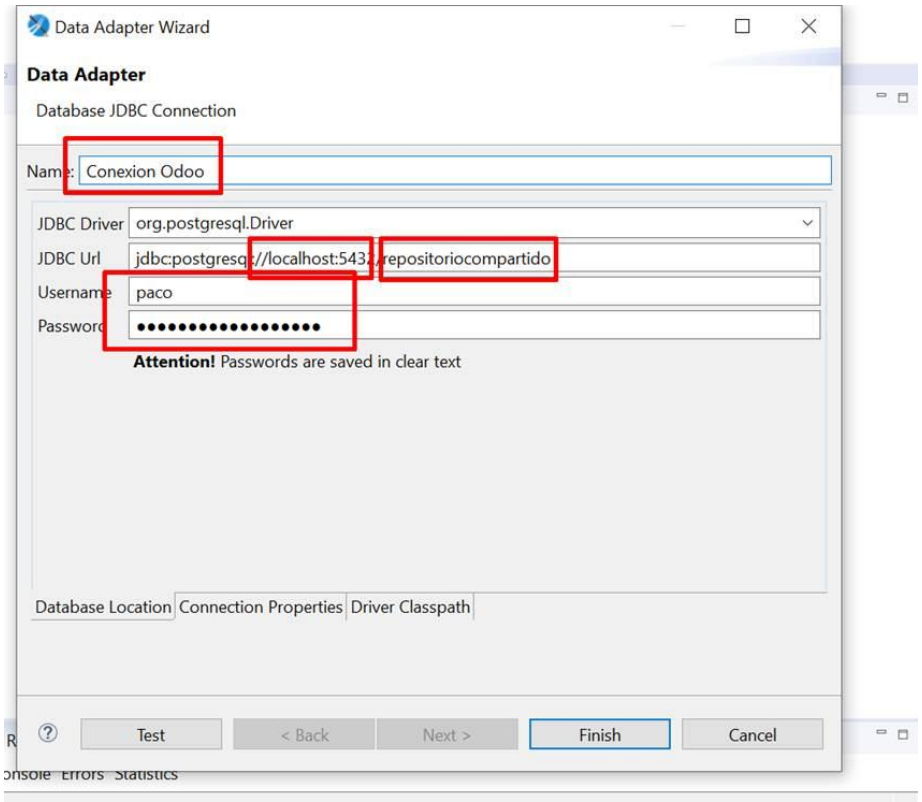


Selección del conector PostgreSQL

Fuente: Elaboración propia

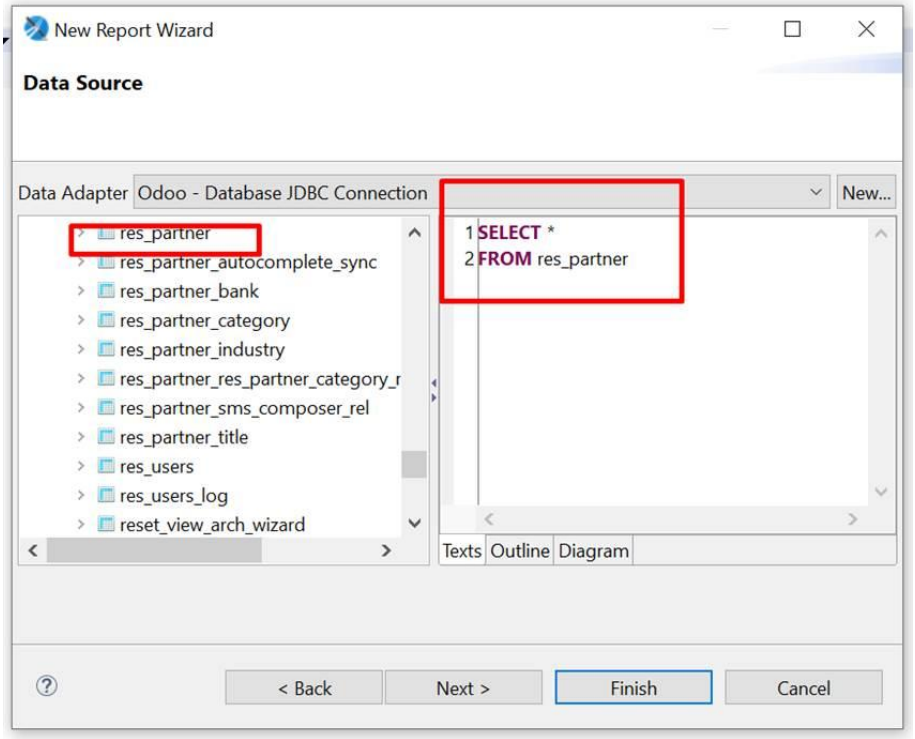
En este punto es importante dar un nombre a la conexión con:

- El servidor. En nuestro caso, utilizamos localhost en la instalación.
- El puerto, por defecto, es 5432.
- La base de datos, en nuestro caso, fue repositorio compartido.
- El usuario y el password.



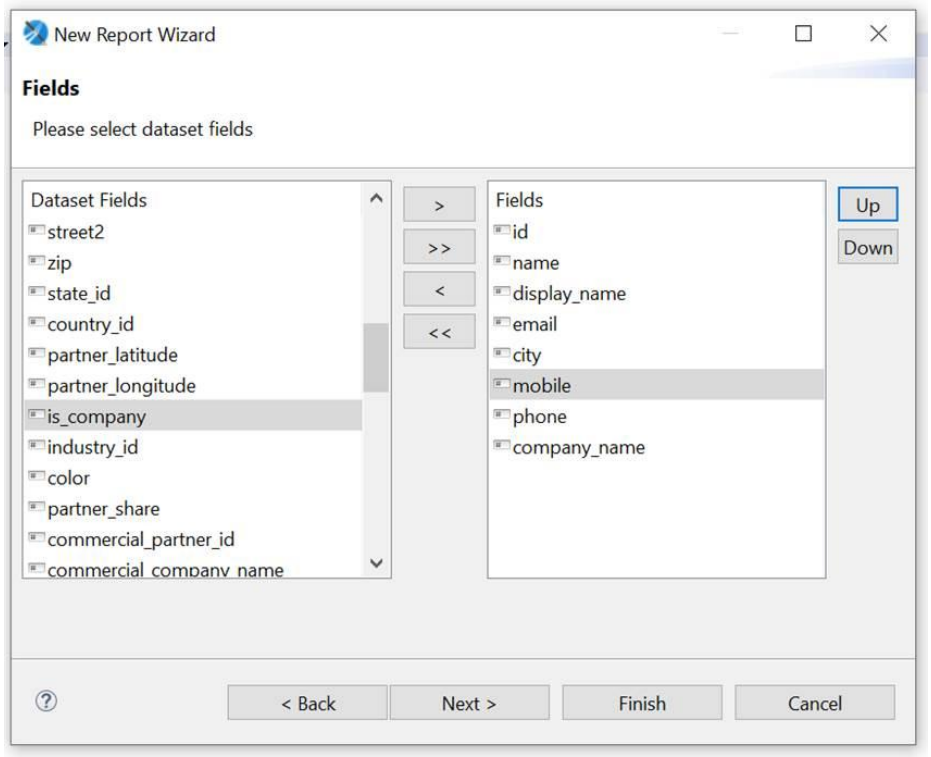
Datos de la conexión
Fuente: Elaboración propia

Escribimos la consulta para tener todos los campos de la tabla res_partner donde tenemos a los clientes.



Datos para el informe
Fuente: Elaboración propia

A partir de esa consulta elegiremos los campos que necesitamos:



Selección de los campos para el informe

Fuente: Elaboración propia

2.5.5 Diseño con Jaspersoft Studio

Una vez que hemos realizado correctamente la conexión y preparación de datos para nuestro primer informe, Jaspersoft nos muestra el entorno de diseño con el report en blanco. En primer lugar, vemos las partes en las que se divide un report:

- **Título**. Se refiere a la primera página de nuestro informe y es aquí donde normalmente se incluyen imágenes y títulos principales.

- **Cabecera de página**. Esta cabecera se repetirá a lo largo de las diferentes páginas del informe.

- **Cabecera de columna.** Esta normalmente se utiliza para presentar unos títulos de la información.

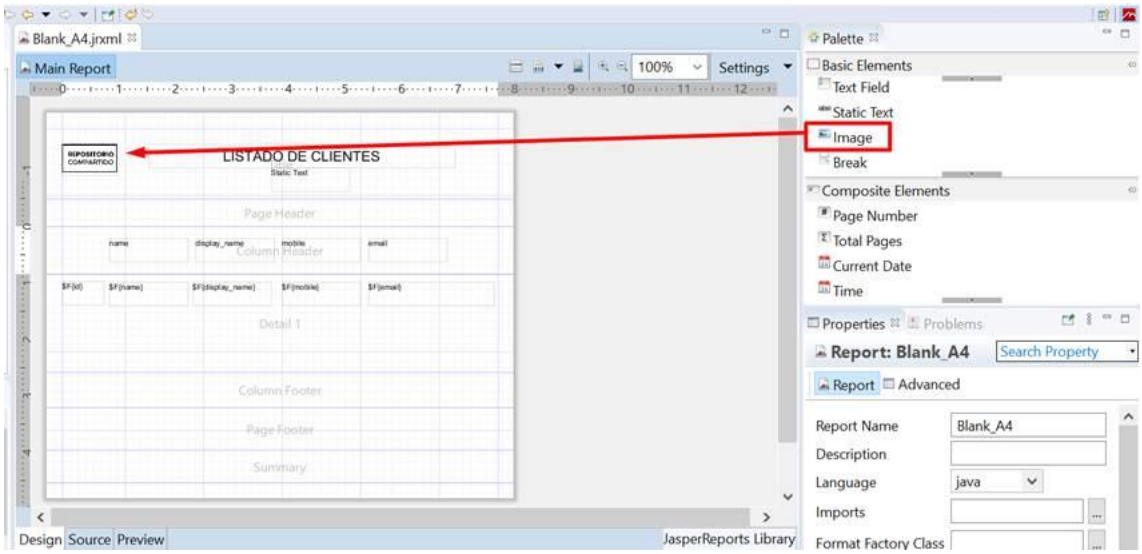
- **Detalle.** En esta zona se incluye la información referida a los datos de la base de datos.

- **Footer.** Al igual que ocurre con la cabecera, tenemos un pie de página que se repetirá a lo largo del informe.

- **Summary.** En este caso, este apartado se mostrará al final del informe.

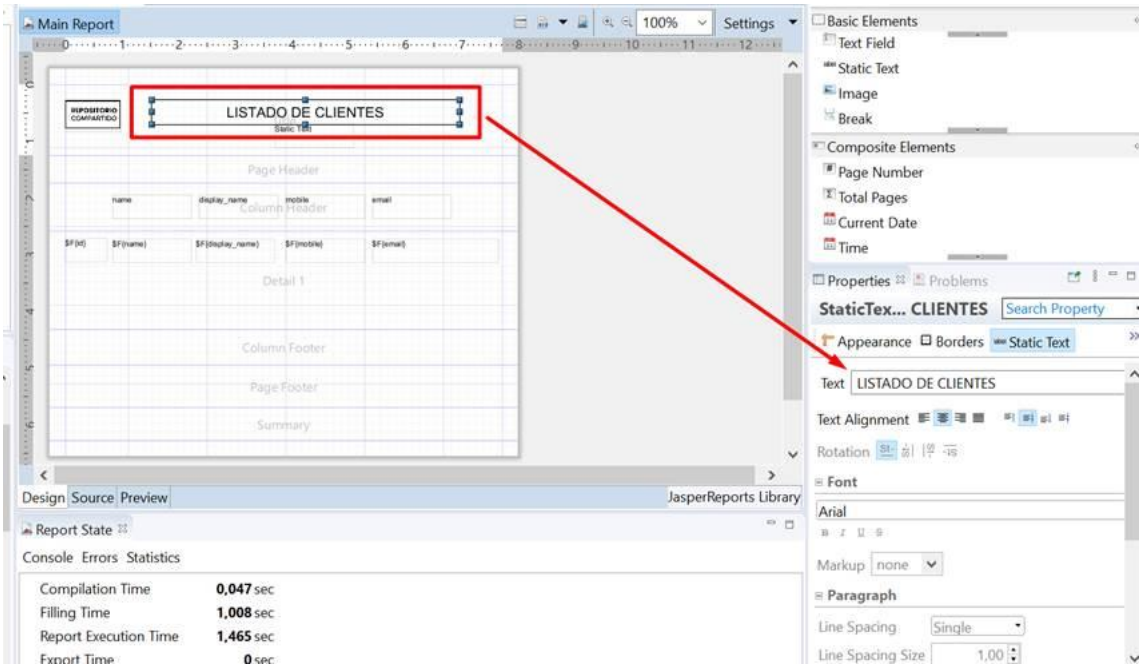
Diseño del informe
Fuente: Elaboración propia

En el diseñador tenemos la paleta de elementos que podremos utilizar para añadir multimedia, textos u otros objetos para hacer más atractivo el informe.



Elemento tipo imagen
Fuente: Elaboración propia

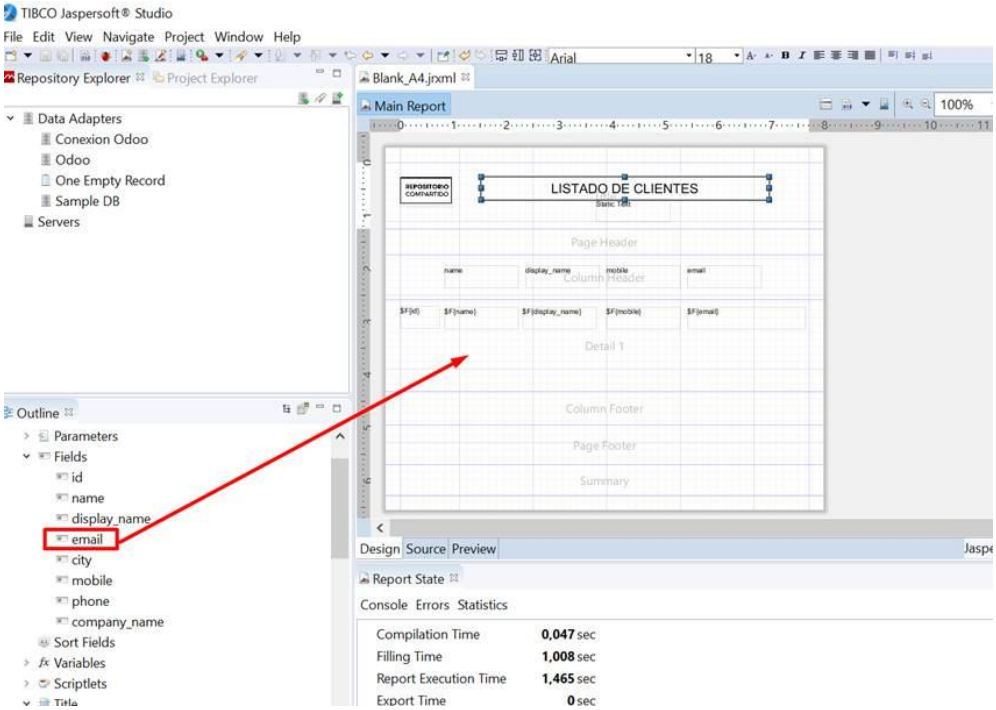
Una vez seleccionado un objeto o incluido un objeto en el diseñador, podremos cambiar las características y propiedades del objeto.



Propiedades de los objetos

Fuente: Elaboración propia

Para poder incluir los datos que hemos seleccionado a través de la consulta programada durante la creación del informe, arrastraremos el campo en la zona de detalle, tal y como muestra la imagen.



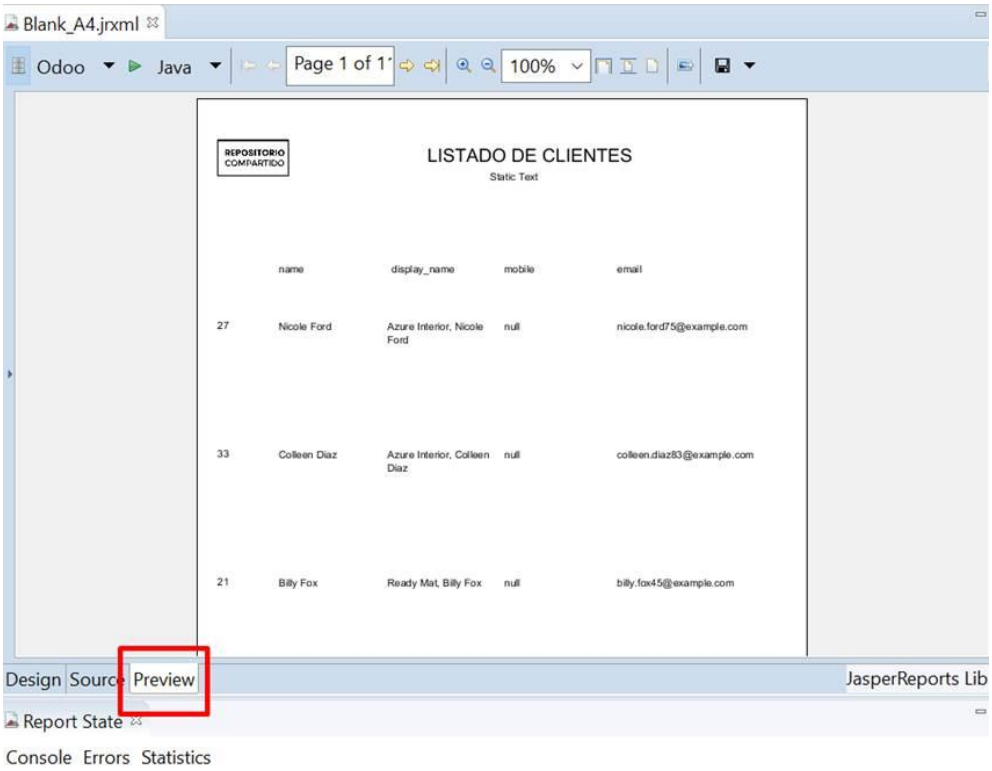
Inserción de campos

Fuente: Elaboración propia

2.5.6 Previsualización con Jaspersoft Studio

Durante el desarrollo del diseño del informe, es importante conocer cómo es el resultado del mismo y si necesitamos realizar modificaciones de diseño, de campos o incluso cambiar la consulta para añadir nuevas tablas o nuevos campos.

En este caso, Jaspersoft Studio nos ofrece la opción de previsualización, como aparece en la imagen.



Previsualización

Fuente: Elaboración propia

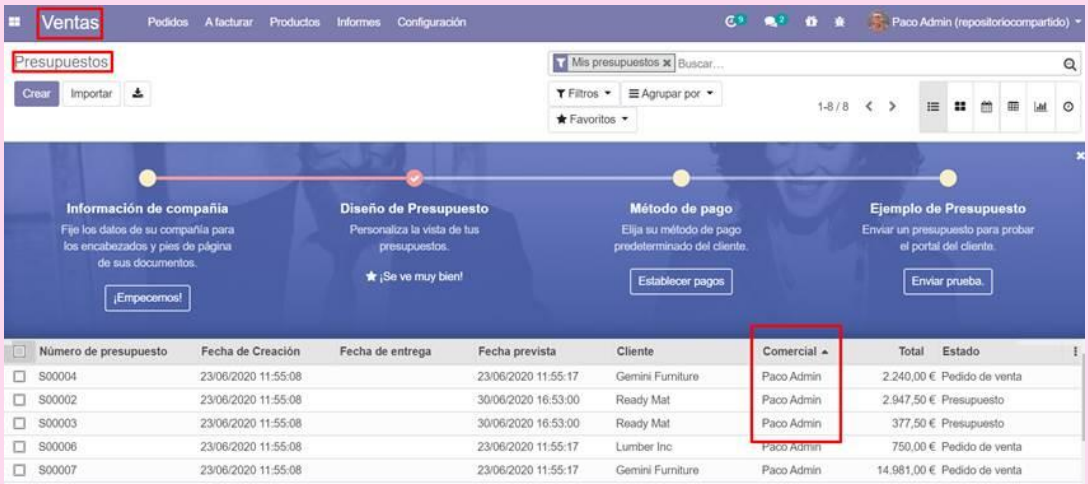
EJEMPLO PRÁCTICO

En el departamento comercial, antes de tener un ERP-CRM, tiene cada comercial el listado de sus propios presupuestos, siendo un inconveniente por tener datos desactualizados, duplicidades, errores de cálculo o, incluso, presupuestos que no coinciden con los que finalmente se registran en administración.

Por ello, como técnico encargado del sistema, te piden investigar cómo obtener esos listados, tanto desde el propio Odoo como a través de herramientas externas como Jaspersoft Studio, asegurando que los responsables comerciales y la dirección puedan tomar decisiones con información coherente y centralizada.

Tendrás dos opciones, una es a través de Odoo y otra es a través de JasperSoft Studio:

1. A través de Odoo tenemos directamente el listado de los presupuestos para un determinado usuario.



Listado de presupuestos por comercial

Fuente: Elaboración propia

A partir de este listado si queremos imprimirlo o exportarlo a Excel simplemente debemos seleccionarlos.

2. Mediante JasperSoft Studio, es importante conocer que intervienen 3 tablas, que las descubrimos cuando estábamos trabajando como desarrolladores:
   a. La tabla sale_order.
   b. La tabla res_users.
   c. La tabla res_partners con los datos para ese user.

A partir de esos datos podremos generar nuestro listado de presupuestos para un comercial.

Datos relacionados en modo desarrollador

Fuente: Elaboración propia

2.6 Exportación de datos e informes

La exportación de información y datos es el proceso en el que extraemos la información a partir de nuestro sistema ERP-CRM y la convertimos en otro formato o almacenamos en otro lugar. Una vez que los datos son exportados de nuestro sistema a otro formato u otro sistema, estos datos quedan desconectados de nuestro ERP-CRM a no ser que se actualice la exportación.

Como vemos, podemos plantear dos tipos de exportaciones:

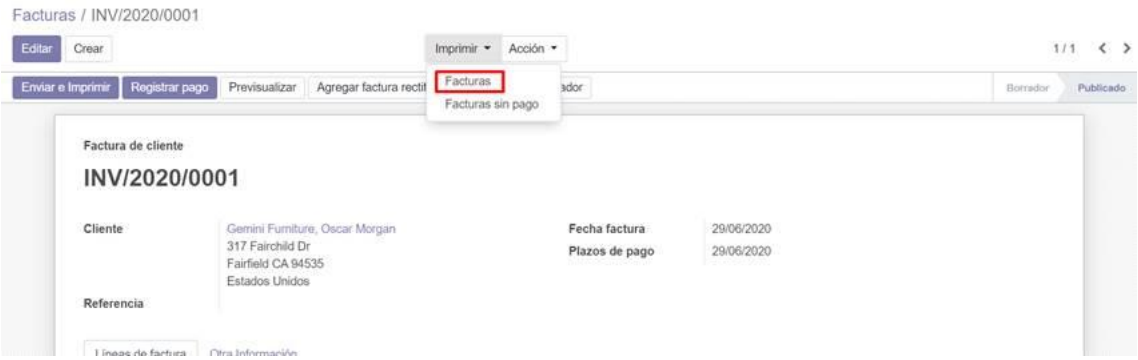
- Exportación a fichero tipo PDF, Excel, Word u otro, legible por una persona y cuya finalidad principalmente es su utilización para la toma de decisiones, control de estado o generación de documentación.

- Exportación a sistemas específicos, la cual consiste en la exportación de la información a una segunda base de datos que está utilizando otro sistema de gestión o de análisis.

2.6.1 Exportar a PDF

Dentro de Odoo encontramos que muchos informes y documentos finales pueden ser exportados a pdf o bien enviados por correo electrónico. Es el caso de las facturas a clientes. Dentro de la factura, y dentro de la acción “Imprimir Factura”, podemos generar el PDF de la factura.

Los PDF usan plantillas QWeb declaradas con `ir.actions.report` en los módulos, personalízalas con herencia de plantillas.

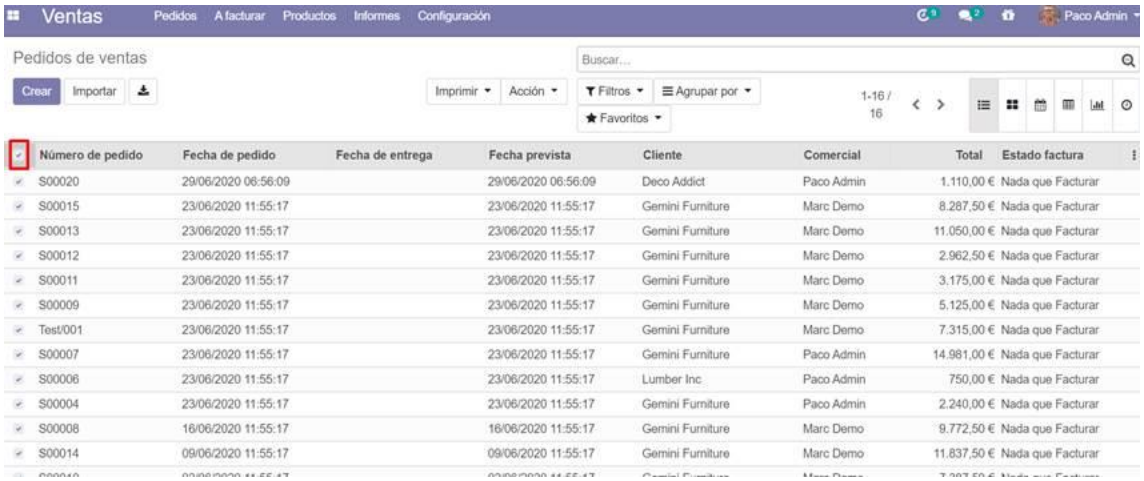


Generación de PDF
Fuente: Elaboración propia

2.6.2 Exportar a Excel/CSV

Muchos de los listados tienen la opción dentro de Odoo de exportar la información a CSV o Excel.

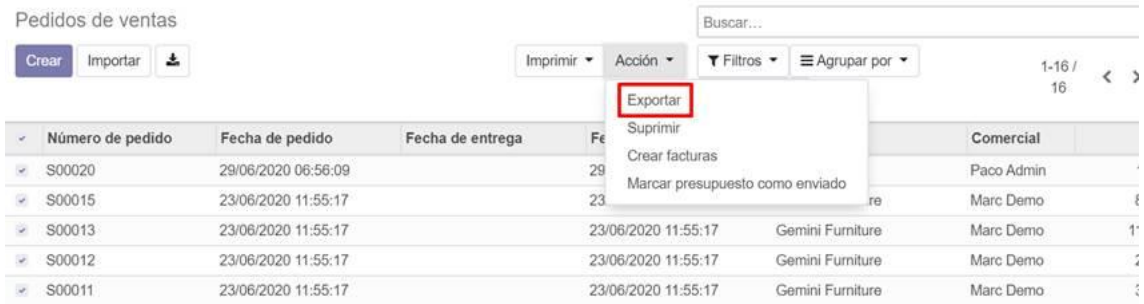
Por ejemplo, tengamos el ejemplo de pedidos de ventas. El primer paso es hacer clic sobre todos los pedidos.



Selección de datos

Fuente: Elaboración propia

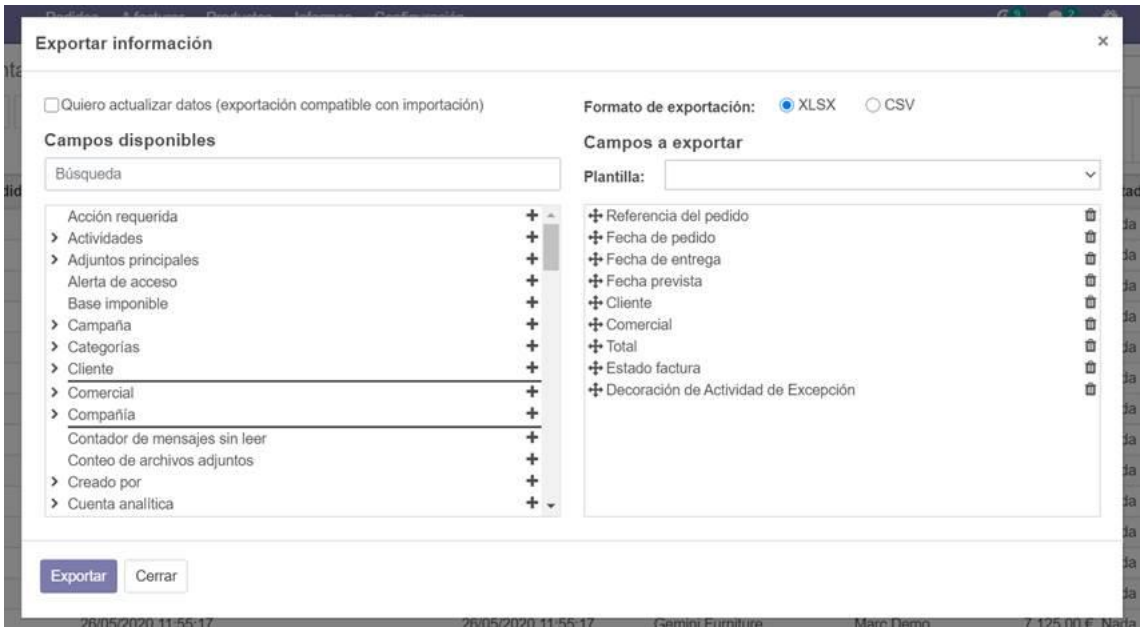
Al seleccionar una, varias o todas las líneas, aparece un nuevo botón de acción y, dentro de las posibles acciones, está la de exportar.



Exportación a Excel

Fuente: Elaboración propia

Al hacer clic sobre exportar nos aparece, a su vez, un nuevo menú donde tenemos, por un lado, los campos posibles y, por otro, los campos seleccionados, así como el formato final.



Configuración de la exportación

Fuente: Elaboración propia

Si apretamos el botón exportar, obtendremos el fichero final Excel.

ENLACE DE INTERÉS

Analiza las posibilidades que nos ofrece dentro de Libre Office, Calc y Libre Office Base.

2.7 Inteligencia de negocio (Business Intelligence)

La inteligencia de negocio (BI) consiste en transformar los datos almacenados en un ERP-CRM en información útil para la toma de decisiones. No se trata únicamente de almacenar datos, sino de analizarlos, representarlos y compartirlos de forma que directivos, mandos intermedios y personal operativo puedan tomar mejores decisiones en tiempo real.

En el caso de Odoo y otros sistemas ERP-CRM, el módulo de BI se integra tanto con las vistas internas (Pivot, Gráfico, Kanban, Spreadsheet) como con herramientas externas de análisis (Power BI, Metabase, Tableau, JasperSoft, etc.)

2.7.1 BI dentro de Odoo

Odoo incluye herramientas integradas que permiten realizar análisis sin salir de la aplicación:

- **Vistas Pivot:** tablas dinámicas que permiten cruzar datos por dimensiones (ejemplo: ventas por mes y por comercial).

- **Vistas Gráfico:** representaciones visuales (barras, líneas, tarta) de datos seleccionados.

- **Odoo Spreadsheet:** permite insertar los resultados de pivot/gráfico en una hoja de cálculo integrada, con KPIs y paneles dinámicos.

- **Cuadros de mando:** cada usuario puede configurar su propia vista de indicadores clave (ventas, compras, incidencias, etc.).

A partir de la versión 16 de Odoo, las hojas de cálculo (Spreadsheet) permiten además realizar simulaciones (qué pasaría si…) y compartir tableros en tiempo real con el equipo.

2.7.2 BI con herramientas externas

En algunos casos, la empresa necesita análisis más avanzados, históricos o que integren varias fuentes (ERP, CRM, web, redes sociales, e-commerce). Para ello, se utilizan plataformas de BI externas conectadas a la base de datos de Odoo (PostgreSQL).

Ejemplos:

- **Power BI (Microsoft):** conexión a PostgreSQL mediante el conector Npgsql. Permite crear paneles interactivos accesibles desde web o móvil.

- **Metabase (software libre):** interfaz sencilla, con generación de dashboards y preguntas dinámicas sobre la base de datos.

- **Apache Superset (software libre):** plataforma de BI avanzada, orientada a grandes volúmenes de datos.

- **JasperSoft Studio**: útil para diseñar informes personalizados y listados detallados, como ya vimos en apartados anteriores.

Lo correcto es crear un usuario de solo lectura en PostgreSQL para BI externo, limitando el acceso a vistas o esquemas preparados para reporting.

2.7.3 Arquitectura de BI

- **Acceso directo (Live):** conexión de la herramienta BI directamente a PostgreSQL de Odoo. Útil para consultas rápidas, pero puede impactar en el rendimiento.

- **ETL y Data Mart:** extracción, transformación y carga de datos hacia un pequeño repositorio agregado (ventas mensuales, inventario histórico).

- **Data Warehouse:** integración de múltiples sistemas (ERP, CRM, e-commerce, marketing digital) en un repositorio central para análisis global.

3. HERRAMIENTAS DE MONITORIZACIÓN Y DE EVALUACIÓN DEL RENDIMIENTO

Una vez instalado el ERP y en funcionamiento, debes garantizar su estabilidad, seleccionando qué herramientas de monitorización y auditoría implementarías para controlar servidores, aplicaciones y accesos.

Para un técnico es muy importante conocer en cualquier momento en qué estado se encuentran los sistemas y las aplicaciones que tiene su empresa o a los que está a cargo. En el caso de los sistemas de gestión empresarial debemos tener en cuenta planificar tres sistemas:

- **Las infraestructuras.** Es importante tener herramientas que analicen y nos permitan actuar ante errores sobre los servidores donde tenemos instaladas las aplicaciones. Tanto si nuestras aplicaciones utilizan sistemas en la nube como si utilizan sistemas locales, las medidas más comunes que podremos y deberemos realizar son las del análisis del espacio, la conectividad y el funcionamiento del hardware en general. En el caso de sistemas en la nube es especialmente sencillo tener estos valores y poder actuar en consecuencia, ya que todos los proveedores de sistemas ya incluyen dentro del sistema herramientas de control y notificaciones personalizadas.

- **El software de gestión empresarial.** Este es más complejo de mantener, ya que son sistemas normalmente cerrados y propietarios que, o bien el fabricante nos proporciona las herramientas necesarias para su mantenimiento, o bien el mantenimiento es totalmente dependiente del fabricante.

- **Los datos relacionados.** Dentro de un software de gestión empresarial, la información es el bien más preciado. De hecho, todo el proceso, desde el análisis de una implantación hasta el mantenimiento, gira alrededor de tener esa información centralizada y accesible de forma coherente.

Además de este enfoque clásico de infraestructura-software-datos, hoy es habitual desplegar monitorización por capas:

1. Métrica (CPU, RAM, tiempos de respuesta)
2. Logs centralizados
3. Trazas de aplicaciones
4. Experiencia de usuario (synthetics/real-user monitoring).

Una pila de referencia muy usada es Prometheus (métricas) + Grafana (dashboards) para cuadros de mando en tiempo real, o Nagios/Zabbix si prefieres chequeos activos tradicionales.

Para el caso de sistemas que tengamos instalados en servidores en la empresa, tenemos herramientas como Nagios que es una herramienta de código abierto para la monitorización de redes.

Entre sus múltiples funciones destacaríamos que permite vigilar los equipos (hardware) y servicios (software) que se especifiquen, alertando cuando el comportamiento de estos no es el deseado.

Entre sus características principales caben destacar que:

- Permite la monitorización de servicios de red (SMTP, POP3, HTTP, SNMP...).
- Permite la monitorización de los recursos de sistemas hardware (carga del procesador, uso de los discos, memoria, estado de los puertos...).
- Garantiza la independencia de sistemas operativos.
- Posibilita la monitorización remota mediante túneles SSL cifrados o SSH.
- Posibilita la programación de plugins específicos para nuevos sistemas.

Es un sistema ampliamente utilizado por su gran versatilidad, puesto que permite analizar casi cualquier parámetro además de generar alertas que pueden ser recibidas por el administrador de la red a través del correo electrónico.

Como alternativas y complementos más destacados encontramos:

- Prometheus, para recopilar series temporales (con exporters para PostgreSQL, Nginx, sistemas, etc.) y alertas, se consulta con PromQL.
- Grafana, para dashboards (integra Prometheus, PostgreSQL, Loki, etc.)
- Centralización de logs con Elastic Stack / OpenSearch o Grafana Loki (logs + etiquetas, muy ligero).

ENLACE DE INTERÉS

En este enlace tienes más información y la referencia al fabricante sobre Nagios.

3.1 Trazas del sistema (logs)

En informática denominamos "log" al historial o registro de las acciones, eventos, errores y otros comportamientos que se den dentro de la aplicación. Normalmente, los logs son ficheros planos almacenados en el servidor, aunque nos encontramos sistemas más complejos que pueden estructurar la información de almacenaje en bases de datos.

En el caso de una instalación Odoo como la que estamos trabajando, tendremos principalmente dos tipos de registros o logs:

- La del servidor de aplicaciones, en la que tendremos información sobre todo lo que tenga que ver con las peticiones, los usuarios y dicho servidor web.
- La del servidor de datos o base de datos, en la que tenemos información detallada del funcionamiento del servidor, pero también de las peticiones de datos a la base de datos.

En Odoo 18, lo habitual en producción es desplegar en Linux (paquete `odoo` y servicio `systemd`) y escribir el log en `/var/log/odoo/odoo-server.log` (configurable en `odoo.conf`: `logfile=`, `log_level=`, `logrotate=`). En Windows o entornos de pruebas, la ruta dependerá de la instalación.

El código sería para unidad de systemd (/etc/systemd/system/odoo.service)

```
[Unit]
Description=Odoo 18 ERP
After=network.target postgresql.service

[Service]
Type=simple
User=odoo
Group=odoo
# Ruta al ejecutable de Odoo (ajústala según tu instalación)
ExecStart=/usr/bin/odoo -c /etc/odoo/odoo.conf
# Reinicio ante fallo
Restart=on-failure
RestartSec=5s
# Entorno (opcional)
Environment=PYTHONUNBUFFERED=1

# Seguridad recomendada (opcional, ajusta a tu entorno)
NoNewPrivileges=true
PrivateTmp=true
ProtectSystem=full
ProtectHome=true

[Install]
WantedBy=multi-user.target
```

Para PostgreSQL, la configuración de logging se realiza en `postgresql.conf` con parámetros como `log_destination` (stderr, csvlog, jsonlog, syslog) y `log_line_prefix` (prefijo con PID, usuario, DB, etc.)

El código para abrir el colector y elegir destino, sería:

```
# Requiere reinicio de servicio si cambias logging_collector
logging_collector = on

# Destinos disponibles: stderr, csvlog, jsonlog, syslog
# (csvlog y jsonlog necesitan logging_collector=on)
log_destination = 'jsonlog'     # o 'csvlog', 'stderr', 'syslog', o combinaciones
```

Como muchos logs, la información que encontramos se reparte en campos:

- Fecha y hora del registro.
- Proceso que provoca el registro. Normalmente, siempre es el mismo proceso y viene representado por un identificador numérico.
- Tipo o categoría del registro. Dependiendo de la aplicación, nos encontramos con registros de tipo INFO, WARN, ERR…
- Detalle del registro.

Como observamos en nuestro extracto del fichero de eventos, la configuración del servidor está en un fichero denominado "odoo.conf", en el cual podemos encontrar información sobre la configuración de los logs y otros parámetros como podemos ver en el siguiente código.

```
[options]
; --- conexión a PostgreSQL ---
db_host = 127.0.0.1
db_port = 5432
db_user = odoo
db_password = TU_PASSWORD

; --- rutas ---
data_dir = /var/lib/odoo
addons_path = /usr/lib/python3/dist-packages/odoo/addons,/opt/odoo/custom-addons

; --- logging ---
logfile = /var/log/odoo/odoo-server.log
log_level = info
; rota el log internamente (diaria, mantiene ~30 por defecto)
logrotate = true
; ejemplos de handlers si quieres granularidad:
```

```
; log_handler = werkzeug:WARNING,odoo.sql_db:ERROR,:INFO
```

Por lo tanto, en este fichero podemos configurar el directorio y fichero donde se almacenarán los logs, el nivel hasta el cual queremos almacenar o si queremos almacenarlo en base de datos o no.

Para los permisos y directorios:

```
sudo useradd -r -s /usr/sbin/nologin -d /var/lib/odoo -m odoo
sudo mkdir -p /var/log/odoo
sudo chown -R odoo:odoo /var/log/odoo /var/lib/odoo /opt/odoo
```

Para el caso de la base de datos, en la instalación actual realizada sobre Linux, el fichero de log de PostgreSQL se encuentra en la carpeta configurada en el parámetro `log_directory` del archivo `postgresql.conf` (por defecto, dentro de `…/data/log/`). Si tenemos activado `logging_collector = on` y hemos definido un nombre de fichero con `log_filename`, encontraremos allí los registros generados por el motor de base de datos.

Un ejemplo típico de los registros que nos encontraremos sería:

```
2025-09-11 09:12:31 CEST [1234] user=odoo db=miempresa LOG:  sentencia: BEGIN ISOLATION LEVEL REPEATABLE READ
2025-09-11 09:12:31 CEST [1234] user=odoo db=miempresa LOG:  sentencia: SELECT "res_users".id FROM "res_users" WHERE ("res_users"."active" = true) ORDER BY "res_users"."id"
2025-09-11 09:12:31 CEST [1234] user=odoo db=miempresa LOG:  sentencia: COMMIT
2025-09-11 09:12:32 CEST [1234] user=odoo db=miempresa LOG:  sentencia: DISCARD ALL
```

Como vemos, en este log pueden reflejarse las consultas SQL que se están ejecutando contra nuestra base de datos, así como los eventos internos de PostgreSQL (transacciones, conexiones, checkpoints, etc.).

El fichero que permite configurar todos estos parámetros del servidor y, en particular, la parte relativa al registro de logs es el `postgresql.conf`, normalmente ubicado en el directorio de datos de PostgreSQL (por ejemplo, `/etc/postgresql/15/main/postgresql.conf` en Debian/Ubuntu, o `/var/lib/pgsql/data/postgresql.conf` en distribuciones basadas en Red Hat).

Un ejemplo de configuración mínima de postgresql.conf adaptado al caso de uso Odoo en Linux que hemos ido viendo, sería:

```
# Activar el colector de logs (necesario para volcar a ficheros)
logging_collector = on

# Carpeta donde se almacenarán los logs (relativa a data_directory)
log_directory = 'log'

# Nombre de los ficheros de log
log_filename = 'postgresql-%Y-%m-%d.log'

# Rotación de logs: diario
log_rotation_age = 1d
# Tamaño máximo antes de rotar (0 = ilimitado)
log_rotation_size = 0

# Destino del log: stderr, csvlog, jsonlog, syslog o combinación
log_destination = 'stderr'

# Prefijo de cada línea de log (incluye fecha/hora, PID, usuario, base de datos y host
cliente)
log_line_prefix = '%m [%p] user=%u db=%d host=%h '

# Nivel mínimo de mensajes que se registran
log_min_messages = warning
# Registrar también conexiones y desconexiones
log_connections = on
log_disconnections = on
# Registrar esperas largas por bloqueos
log_lock_waits = on
# Registrar consultas lentas (>500 ms en este ejemplo)
log_min_duration_statement = 500
```

En este ejemplo tenemos como parámetros más importantes:

- `logging_collector`: activa el proceso que captura la salida de logs.
- `log_directory` y `log_filename`: controlan la ubicación y el nombre de los archivos de log.
- `log_line_prefix`: define qué información aparece al inicio de cada línea (muy útil para auditoría y depuración).

- `log_min_duration_statement`: registra consultas que superen un tiempo de ejecución determinado (en milisegundos).
- `log_connections` y `log_disconnections`: dejan constancia de quién entra y sale del sistema.

ARTÍCULO DE INTERÉS

La lectura y manejo de logs no siempre es sencillo y, sobre todo, cuando tenemos que analizar varios sistemas con cientos de miles de entradas por día. En este artículo se analizan posibles herramientas que nos ayudan en esta tarea.

3.2 Auditorías de control de acceso a los datos

La configuración del control de acceso a nuestra aplicación es esencial y la realizamos a dos niveles:

1. **Control de acceso al servidor de base de datos.**

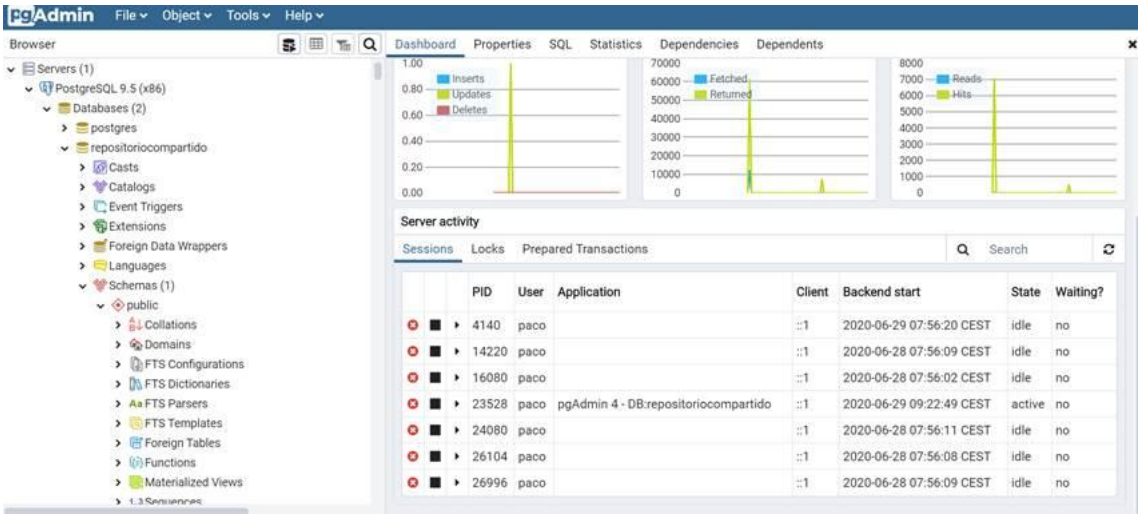
El control de acceso al servidor de base de datos es sencillo de configurar y controlar, ya que deberíamos configurar los siguientes usuarios y permisos de accesos:

- Usuario/s de administración de la base de datos. Se encargará de tareas únicamente relacionadas con la base de datos, tales como programar backups, o tareas de mantenimiento, así como la creación de nuevos usuarios y bases de datos.

- Usuario de accesos a la base de datos del sistema de gestión empresarial. Será un usuario que utilice la aplicación para el acceso a la base de datos y que normalmente será utilizado por la aplicación para realizar todas las consultas a la misma.

- Usuarios de uso y lectura para aplicaciones de exportación, BI y generación de informes.

A través de herramientas como pgAdmin, se puede realizar un seguimiento y auditoría de tanto los usuarios que están realizando un acceso como de qué están realizando. A través del panel de control tenemos un seguimiento y auditoría básica (sesiones, bloqueos, consultas activas):



Herramienta monitorización en pgAdmin

Fuente: Elaboración propia

En estos casos, es recomendable reforzar `pg_hba.conf`, TLS si procede, roles mínimos y password policies. Para BI externo, usa rol RO y, mejor aún, vistas dedicadas.

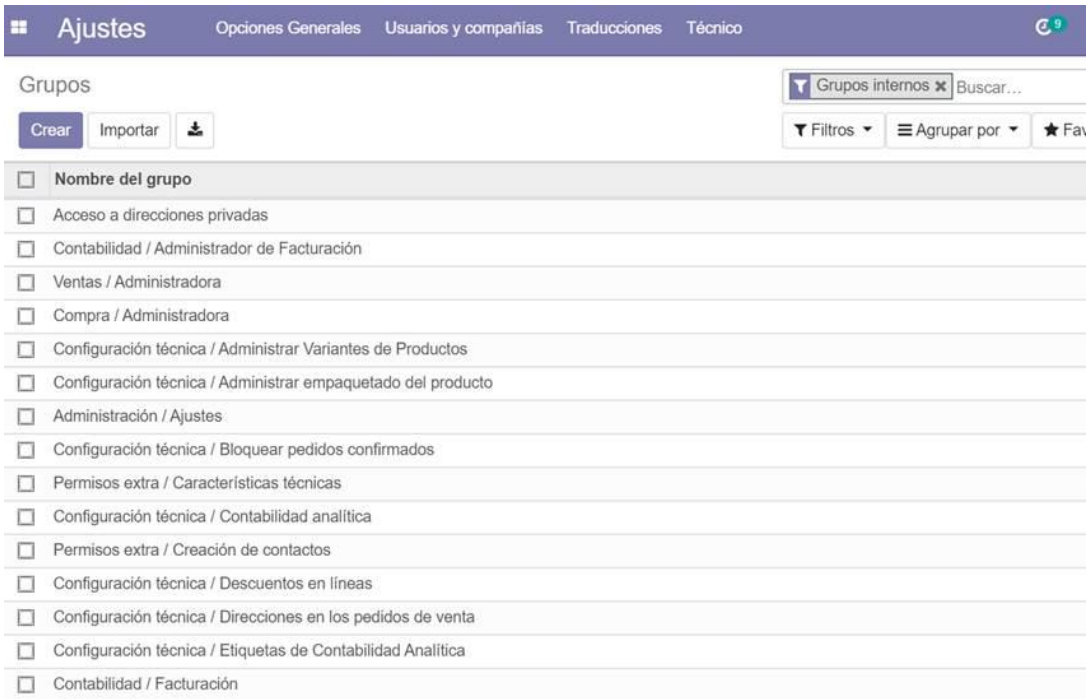
2. Control de acceso a la aplicación.

Para el caso del control de acceso a nivel de aplicación, contamos con la herramienta "usuarios" y los grupos de Odoo, siempre que tengamos un usuario de administración para la gestión de los mismos.

Ajustes de usuarios en Odoo

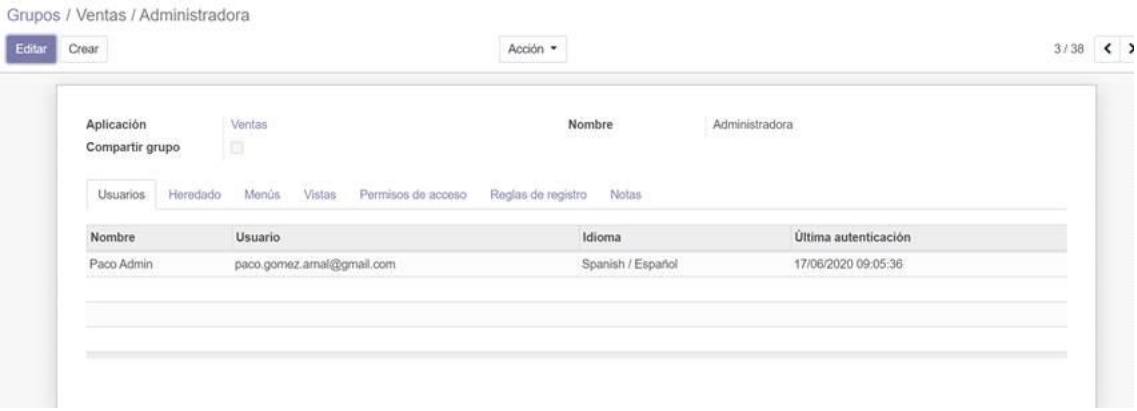
Fuente: Elaboración propia

Mediante el uso del menú “Grupos” podemos crear y configurar grupos de permisos bien para determinadas acciones, o bien para determinados módulos, tal y como vemos en la imagen.



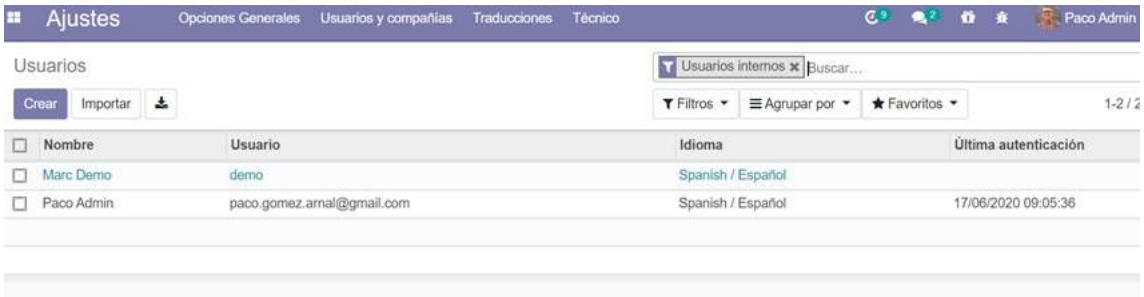
Permisos de usuarios en Odoo
Fuente: Elaboración propia

Además, si entramos en uno de los grupos configurados, por ejemplo, el de ventas, podemos realizar la auditoría de control a un determinado módulo.



Permisos de usuarios por módulo
Fuente: Elaboración propia

Mediante el menú de usuarios podemos crear y modificar usuarios, así como los permisos que estos van a tener, tal y como muestra la imagen.



Listado de usuarios

Fuente: Elaboración propia

En esta última imagen, podemos ver cómo podemos dar accesos y quitarlos a partes del software una vez que tenemos activado el sistema como desarrollador.

Permisos del usuario administrador

Fuente: Elaboración propia

EJEMPLO PRÁCTICO

Durante la evaluación del sistema Odoo y su implementación en la empresa de alimentación donde trabajas como técnico de desarrollo, te plantean también la necesidad de generar el plan de usuarios que va a utilizar la aplicación.

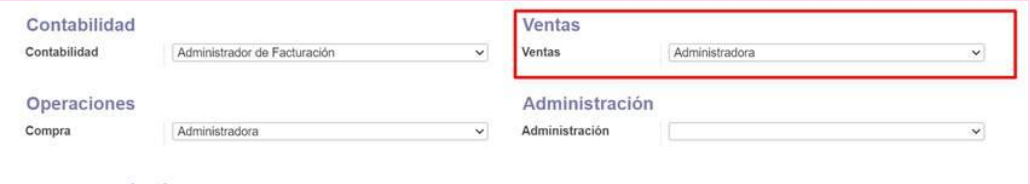
Teniendo en cuenta la siguiente configuración de empresa:

- 1 jefe de almacén por cada uno de los 3 almacenes y 5 operarios.
- 1 jefe de compras.
- 1 director comercial y 5 comerciales.
- 1 director financiero y 2 administrativos.
- 1 gerente.
- 1 director informático y 1 técnico.

¿Cuál sería la mejor configuración de usuarios?

Para un correcto funcionamiento y tener un histórico correcto de todo el trabajo realizado por todos los usuarios necesitamos que cada uno de los mismos tengan un usuario diferente.

A partir de ahí, cada uno de los directores serán administradores de sus respectivos módulos y, el resto, solo usuarios. Por ejemplo, al director comercial se le dará permisos como administrador de ventas y al resto de comerciales únicamente como usuarios, tal y como se ve en la imagen.



Permisos del usuario administrador

Fuente: Elaboración propia

3.3 Incidencias: Identificación y resolución

Una incidencia se puede definir como todo proceso que no transcurre dentro de la normalidad programada o configurada. Las incidencias en los sistemas de gestión empresarial son muy diversas tanto en el origen como en su contenido:

- **A nivel de plataforma.** Puede deberse a un fallo en el servidor hardware, del tipo fallo en el disco duro o bien a nivel del sistema operativo.

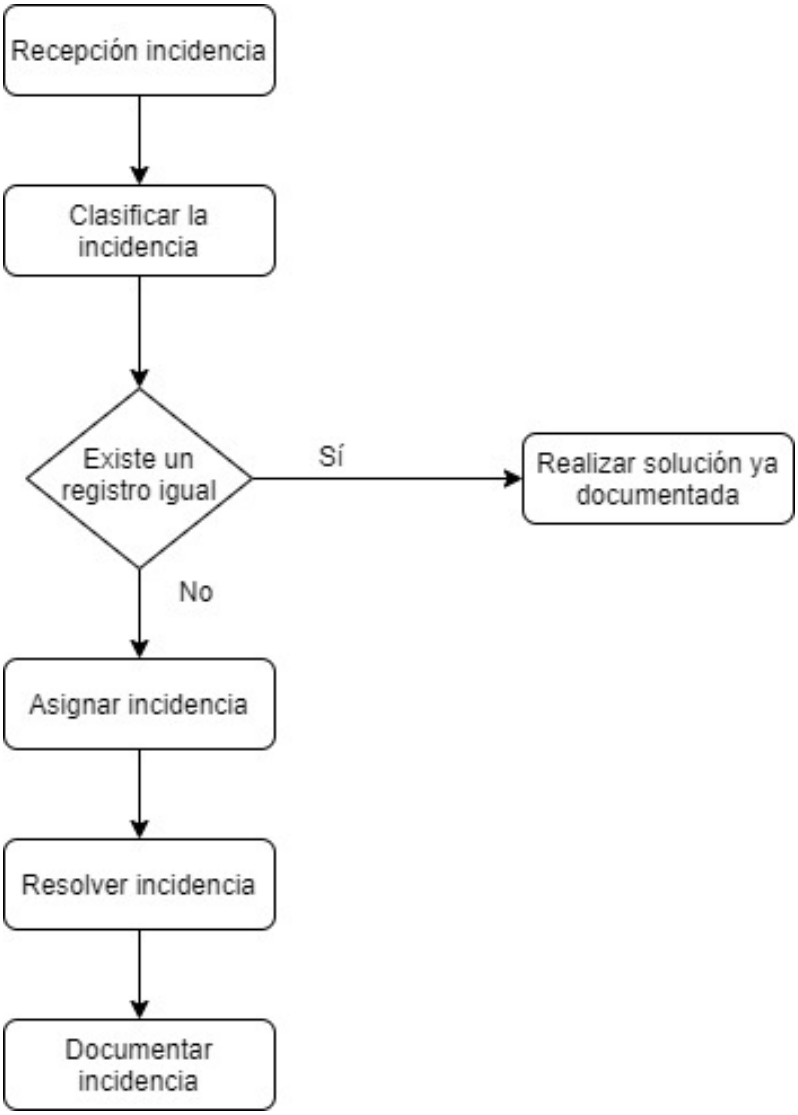
- **A nivel de aplicación.** En ese caso las incidencias son muy variadas, porque pueden deberse a causas técnicas del propio software o bien a nivel de negocio, mucho más complejas de analizar y solucionar, ya que en ese caso puede tratarse de una incorrecta implantación o parametrización.

- **A nivel de información.** No nos referimos a incidencias a nivel del gestor de bases de datos, que puede englobarse dentro del apartado anterior, más bien nos referimos a una mala calidad del almacenamiento de la información, ya que nos podemos encontrar con entradas duplicadas, mal escritas o incluso no enlazadas correctamente.

Sea cual sea la incidencia y el problema, es muy importante que el usuario tenga un sistema de helpdesk para poder recoger la incidencia, evaluarla, asignarla, resolverla y documentarla. La documentación en este proceso es imprescindible ya que nos permite, como técnicos, poder tener una base de información que se pueda utilizar en un futuro.

Si trabajas con Odoo, tienes la opción de activar el módulo Helpdesk (gestión de tickets, equipos, SLA, base de conocimiento, portal de cliente). Permite pipelines por estados, artículos de Knowledge y Help Center para autoservicio.

Un posible diagrama de flujo de trabajo de helpdesk podría ser el siguiente:



Flujo de trabajo para helpdesk

Fuente: Elaboración propia

RESUMEN FINAL

En esta unidad hemos tratado el concepto de base de datos como núcleo de cualquier sistema de gestión empresarial, destacando PostgreSQL como motor principal en Odoo, pudiendo, desde ella organizar tablas, vistas, modelos y campos que permiten almacenar y estructurar la información de clientes, proveedores, pedidos o facturas. También se ha visto cómo acceder y consultar esos datos mediante SQL y cómo aprovechar procedimientos y cálculos que automatizan procesos habituales en la empresa.

A continuación, se han estudiado las interfaces que permiten a los usuarios trabajar con la información: formularios, listados, informes y gráficos. Además, de hacer un recorrido por las acciones de extraer datos mediante Odoo o con herramientas externas como Jaspersoft Studio, así como su exportación a diferentes formatos. Cerramos esta parte con la introducción a la inteligencia de negocio, tanto con funciones integradas de Odoo como con soluciones externas como Power BI o Metabase.

Para finalizar, hemos hecho un recorrido por las herramientas de monitorización y evaluación del rendimiento, analizando logs del sistema, auditorías de accesos y gestión de incidencias, como cierre de una visión integral sobre cómo organizar, consultar y mantener la información dentro de un ERP-CRM.