

SISTEMAS DE GESTIÓN

CASO PRACTICO I UD3



ALUMNO CESUR

25/26

Alejandro Muñoz de la Sierra

PROFESOR

Óscar González Núñez

INTRODUCCION

En este caso práctico abordamos una situación común en la implantación de un ERP. El departamento de compras ha perdido la confianza en los datos del sistema. El jefe de compras presentó una queja clara. Los listados planos de productos generados por el sistema son confusos. Resultan tediosos de leer y son poco fiables para conocer el catálogo real. Solicitó un informe mensual en el que el catálogo de productos aparezca estructurado, limpio y agrupado por categorías.

Esta petición tiene dos objetivos principales. Primero, auditar visualmente si la información está bien clasificada. Segundo, detectar rápidamente incidencias en las nuevas altas de artículos, como productos huérfanos sin categoría.

Dividimos nuestra actuación en dos frentes para ofrecer una solución útil a la empresa:

A nivel de usuario (**Frontend**): Mejoraremos la interfaz nativa del ERP (**Odoo**). Esto facilitará el trabajo diario del usuario.

A nivel de base de datos (**Backend + BI**): Trabajaremos directamente en **PostgreSQL** para construir una consulta SQL segura. La **conectaremos** con una herramienta de Business Intelligence (**Jaspersoft Studio**). Así generaremos el reporte mensual automatizado, inmutable y listo para imprimir (PDF). La dirección exige este formato para sus reuniones.






PREPARACIÓN DEL ENTORNO DE DESARROLLO (LOCALHOST)

Trabajar con **Odoo** en local consume bastantes recursos de **RAM y CPU**. Detuvimos sus servicios al terminar las unidades anteriores para liberar el ordenador. Acondicionamos nuestro entorno de trabajo antes de iniciar la configuración.

Pensamos en hacer una instalación limpia desde cero. Esta opción suele provocar conflictos de puertos, especialmente con el 5432 de la base de datos PostgreSQL y el 8069 del servidor web de Odoo. **Reactivamos nuestra infraestructura** existente para evitar problemas.

Accedimos al **panel de servicios de Windows (services.msc)**. Localizamos los servicios del motor de base de datos (**postgresql-x64**) y del servidor de aplicaciones (**odoo-server**). Cambiamos su estado a "**Inicio automático**" y los **iniciamos** manualmente.

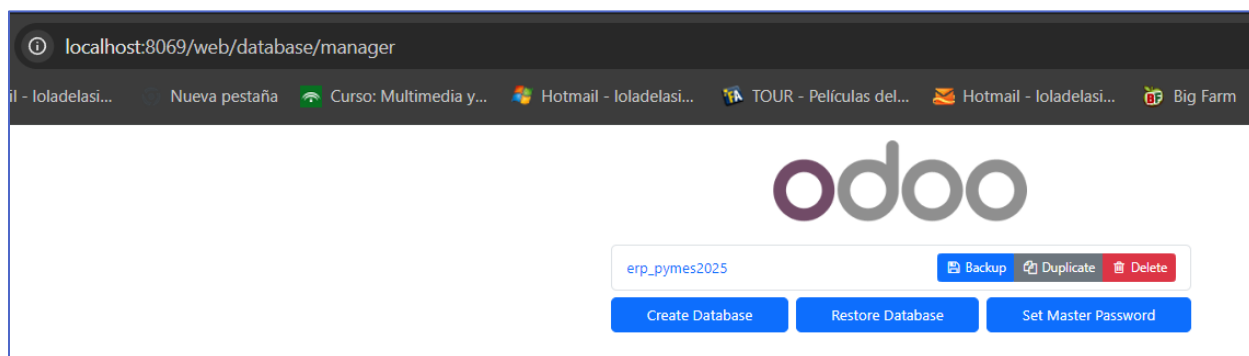
Este mantenimiento nos devolvió el acceso a nuestro localhost:8069. No sobrescribimos las configuraciones globales de prácticas anteriores.

 NVIDIA FrameView SDK service	NVIDIA Fram...		Manual (desen...	Sistema local
 NVIDIA LocalSystem Container	Container se...	En ejecución	Automático	Sistema local
 odoo-server-19.0		En ejecución	Automático	Sistema local
 OneDrive Updater Service	Keeps your ...		Manual (desen...	Sistema local
 OpenSSH Authentication Agent	Agent to hol...		Deshabilitado	Sistema local

CREACIÓN Y DESPLIEGUE DE LA BASE DE DATOS

Estabilizamos el entorno y requerimos un espacio de trabajo aislado (un sandbox) exclusivo para este caso práctico. **No convenía utilizar las bases de datos de las unidades anteriores.** Un informe de agrupaciones necesita un volumen considerable de productos y categorías para tener sentido visual y funcional.

Accedimos al **gestor de bases de datos de Odoo (/web/database/manager)** y creamos una **nueva instancia limpia**. La llamamos **repositorio compartido** según las directrices del enunciado.



Un paso crítico en esta fase fue marcar la casilla de "**Demo data**" (Cargar datos de demostración). Esto ahorra días de trabajo. **Odoo insertó automáticamente un catálogo ficticio completo en el sistema** al crear la base de datos. Incluye **mobiliario de oficina, servicios, consumibles y hardware**. Esto nos ahorró horas de inserción manual de registros como "Silla 1" o "Mesa 2". Pasamos directamente a la fase de análisis. y visualización de la información.

Create Database

Master Password

.....

Database Name

repositoriocompartido

Email

admin

Password

.....

Phone Number

679123456

Language

Spanish / Español

Country

Spain

Demo Data

☒

To enhance your experience, some data may be sent to Odoo online services. See our [Privacy Policy](#).

Continue

localhost:8069/web/database/manager

il - loladelasi...

Nueva pestaña


Curso: Multimedia y...

Hotmail - loladelasi...

TOUR - Películas del...

Hotmail - loladelasi...

Big Farm



erp_pymes2025

Backup

Duplicate

Delete

repositoriocompartido

Backup

Duplicate

Delete

Create Database

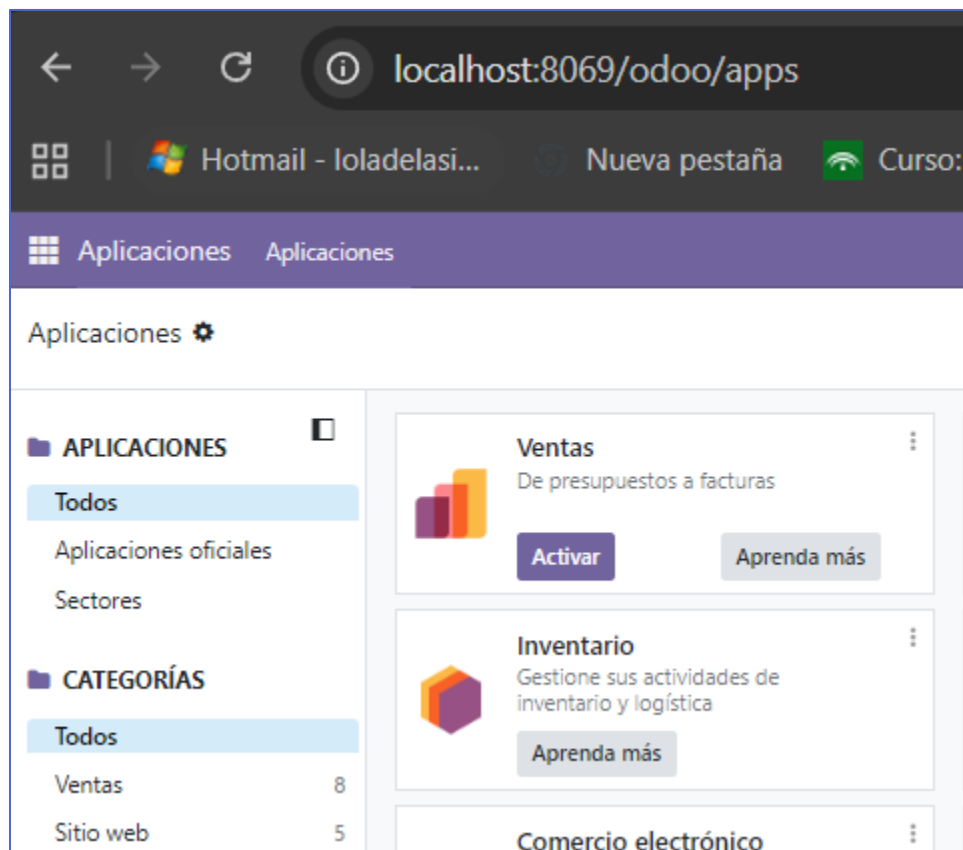
Restore Database

Set Master Password

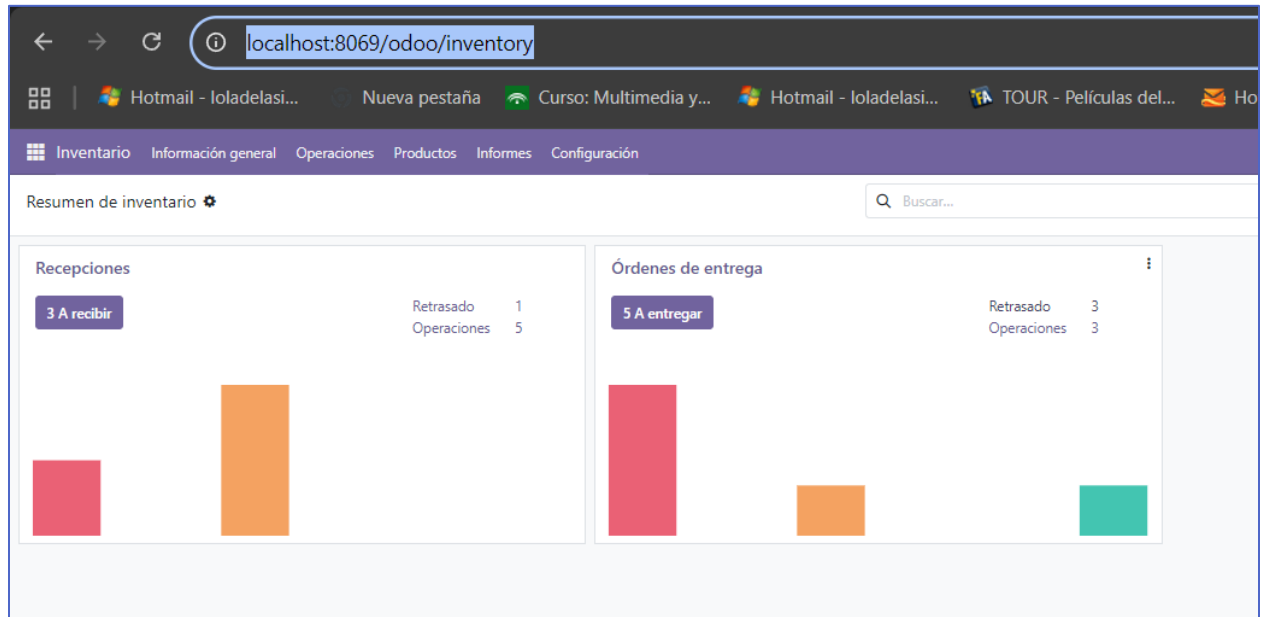
03

FASE 1: CONFIGURACIÓN DE VISTAS NATIVAS EN ODOO

Queríamos dar una respuesta inmediata al jefe de compras. Activamos el **módulo de Inventario** y entrabamos en él. Desarrollábamos el **informe en Jaspersoft** al mismo tiempo. Usamos las capacidades nativas de Odoo. No hace falta programar algo nuevo. Solo hay que configurar lo que ya tenemos.



Módulo inventario



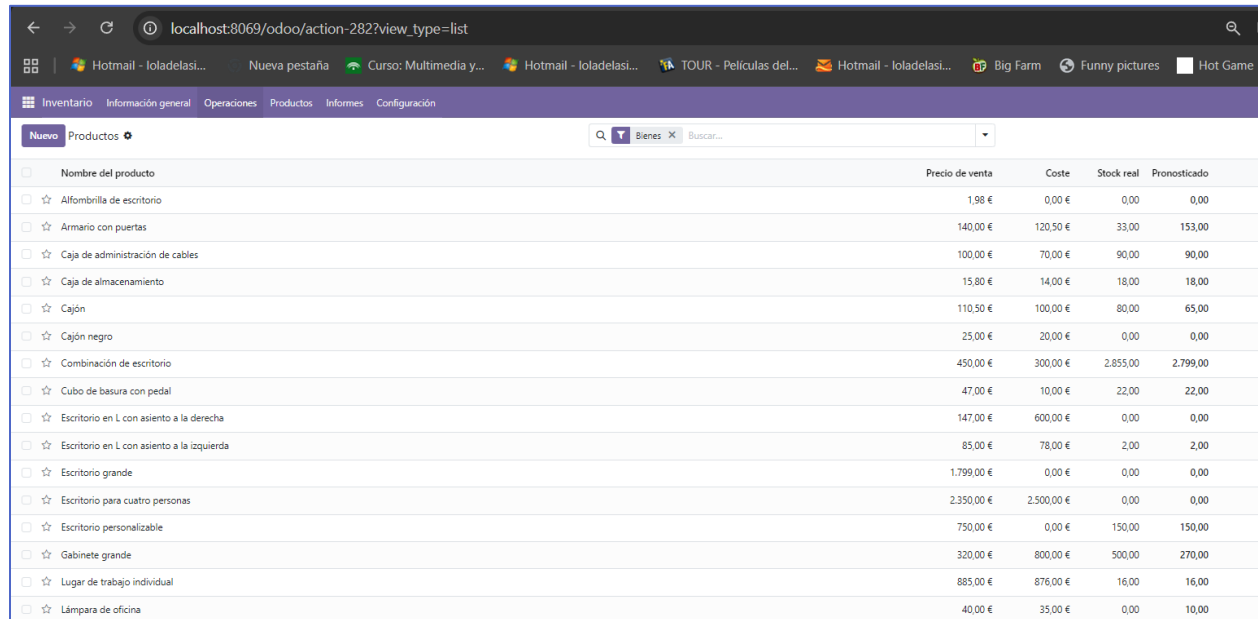
Vista Lista estructurada (El detalle)

Entramos al **catálogo de productos**. El **sistema** nos mostró la **vista Kanban** por defecto. Es el panel de tarjetas con la foto. Esta vista es agradable. Pero no sirve para auditar datos masivos.

The screenshot shows the Odoo Product Catalog interface. The top navigation bar includes 'Inventario', 'Información general', 'Operaciones', 'Productos', 'Informes', and 'Configuración'. The main section is titled 'Nuevos Productos' and features a search bar. Below this, there is a grid of product cards, each displaying a product name, a small image, and pricing information. The products are organized into a Kanban view, with each card representing a product and its details.

Producto	Precio	Stock real
Alfombrilla de escritorio	1,98 €	0,00
Armario con puertas	140,00 €	33,00
Caja de administración de cables	100,00 €	90,00
Caja de almacenamiento	15,80 €	18,00
Cajón	110,50 €	80,00
Cajón negro	25,00 €	0,00
Combinación de escritorio	450,00 €	2.835,00
Cubo de basura con pedal	47,00 €	22,00
Escritorio en L con asiento a la derecha	147,00 €	0,00
Escritorio en L con asiento a la izquierda	85,00 €	2,00
Escritorio grande	1.799,00 €	0,00
Escritorio para cuatro personas	2.350,00 €	0,00
Escritorio personalizable	750,00 €	150,00
Gabinete grande	320,00 €	500,00
Lugar de trabajo individual	885,00 €	16,00
Lámpara de oficina	40,00 €	0,00
Mesa de comedor para exteriores	589,00 €	0,00
Mesa grande para reuniones	4.000,00 €	0,00
Organizador de escritorio	5,10 €	0,00
Pantallas de bloque acústico	295,00 €	16,00
Rotafolio	1.950,00 €	0,00
Silla de oficina	33,00 €	56,00
Silla de oficina negra	120,50 €	10,00
Software de diseño de oficina	280,00 €	0,00
Sofá de dos plazas	1.500,00 €	0,00
Soporte de escritorio con pantalla	2.100,00 €	0,00
Soporte de monitor	3,19 €	0,00

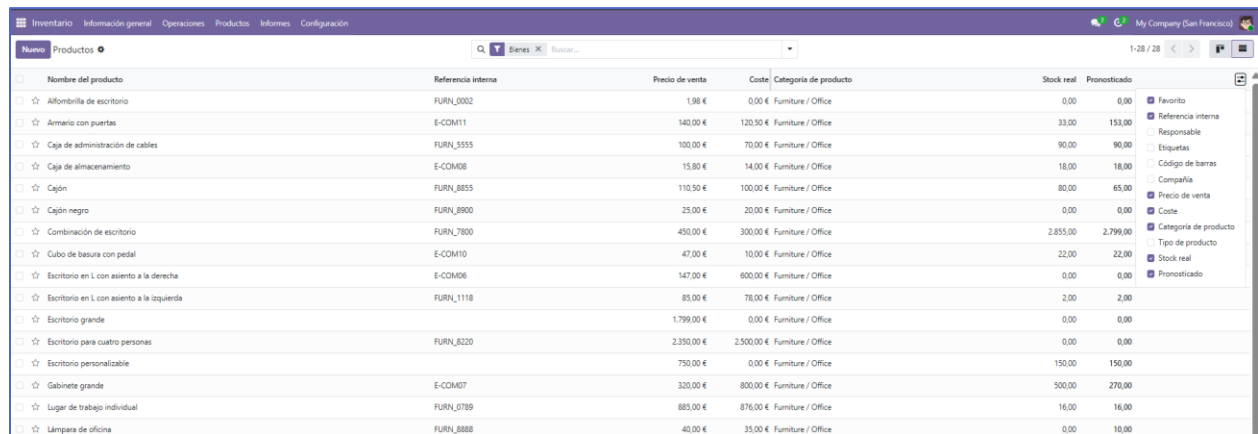
Cambiamos de inmediato a la Vista Lista.



The screenshot shows the Odoo Inventory View List interface. The browser address bar indicates the URL is localhost:8069/odoo/action-282?view_type=list. The top navigation bar includes links for 'Inventario', 'Información general', 'Operaciones', 'Productos', 'Informes', and 'Configuración'. Below this, there's a 'Nuevo' button and a 'Productos' dropdown menu. A search bar is present with the text 'Bienes' and a search icon. The main table displays a list of products with columns for 'Nombre del producto', 'Precio de venta', 'Coste', 'Stock real', and 'Pronosticado'. The products listed include 'Alfombrilla de escritorio', 'Armario con puertas', 'Caja de administración de cables', 'Caja de almacenamiento', 'Cajón', 'Cajón negro', 'Combinación de escritorio', 'Cubo de basura con pedal', 'Escritorio en L con asiento a la derecha', 'Escritorio en L con asiento a la izquierda', 'Escritorio grande', 'Escritorio para cuatro personas', 'Escritorio personalizable', 'Gabinete grande', 'Lugar de trabajo individual', and 'Lámpara de oficina'.

Nombre del producto	Precio de venta	Coste	Stock real	Pronosticado
Alfombrilla de escritorio	1,98 €	0,00 €	0,00	0,00
Armario con puertas	140,00 €	120,50 €	33,00	153,00
Caja de administración de cables	100,00 €	70,00 €	90,00	90,00
Caja de almacenamiento	15,80 €	14,00 €	18,00	18,00
Cajón	110,50 €	100,00 €	80,00	65,00
Cajón negro	25,00 €	20,00 €	0,00	0,00
Combinación de escritorio	450,00 €	300,00 €	2,855,00	2,799,00
Cubo de basura con pedal	47,00 €	10,00 €	22,00	22,00
Escritorio en L con asiento a la derecha	147,00 €	600,00 €	0,00	0,00
Escritorio en L con asiento a la izquierda	85,00 €	78,00 €	2,00	2,00
Escritorio grande	1,799,00 €	0,00 €	0,00	0,00
Escritorio para cuatro personas	2,350,00 €	2,500,00 €	0,00	0,00
Escritorio personalizable	750,00 €	0,00 €	150,00	150,00
Gabinete grande	320,00 €	800,00 €	500,00	270,00
Lugar de trabajo individual	885,00 €	876,00 €	16,00	16,00
Lámpara de oficina	40,00 €	35,00 €	0,00	10,00

Revisamos las columnas por defecto. Faltaba información vital. Abrimos el **configurador de columnas** en la esquina superior derecha. **Añadimos** manualmente "**Referencia Interna**" y "**Categoría del producto**". Estos datos son claves para compras.

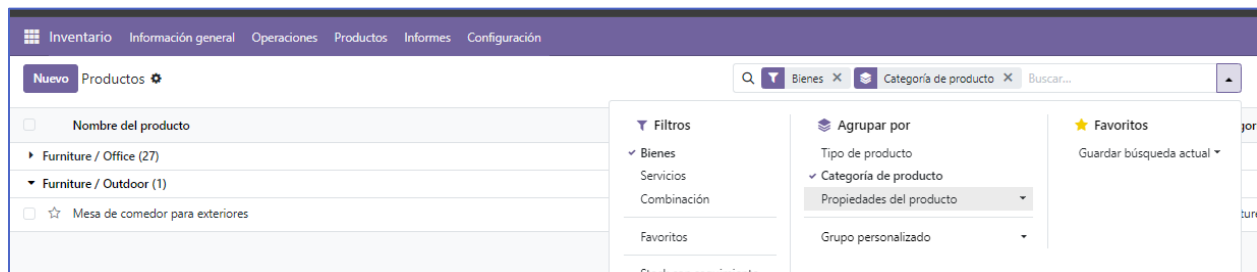


The screenshot shows the Odoo Inventory View List interface with the 'Configuración' (Configuration) button in the top right corner. The table now includes additional columns: 'Referencia interna' and 'Categoría de producto'. The 'Referencia interna' column contains values like 'FURN_0002', 'E-COM11', 'FURN_5555', 'E-COM08', 'FURN_8855', 'FURN_8900', 'FURN_7800', 'E-COM10', 'E-COM06', 'FURN_1118', 'FURN_8320', 'E-COM07', 'FURN_0789', and 'FURN_8888'. The 'Categoría de producto' column contains values like 'Furniture / Office'. The 'Stock real' and 'Pronosticado' columns are also present. The bottom right corner shows a search bar with the text 'Bienes' and a search icon. The bottom right corner also shows a status bar with the text '1-28 / 28' and a refresh button.

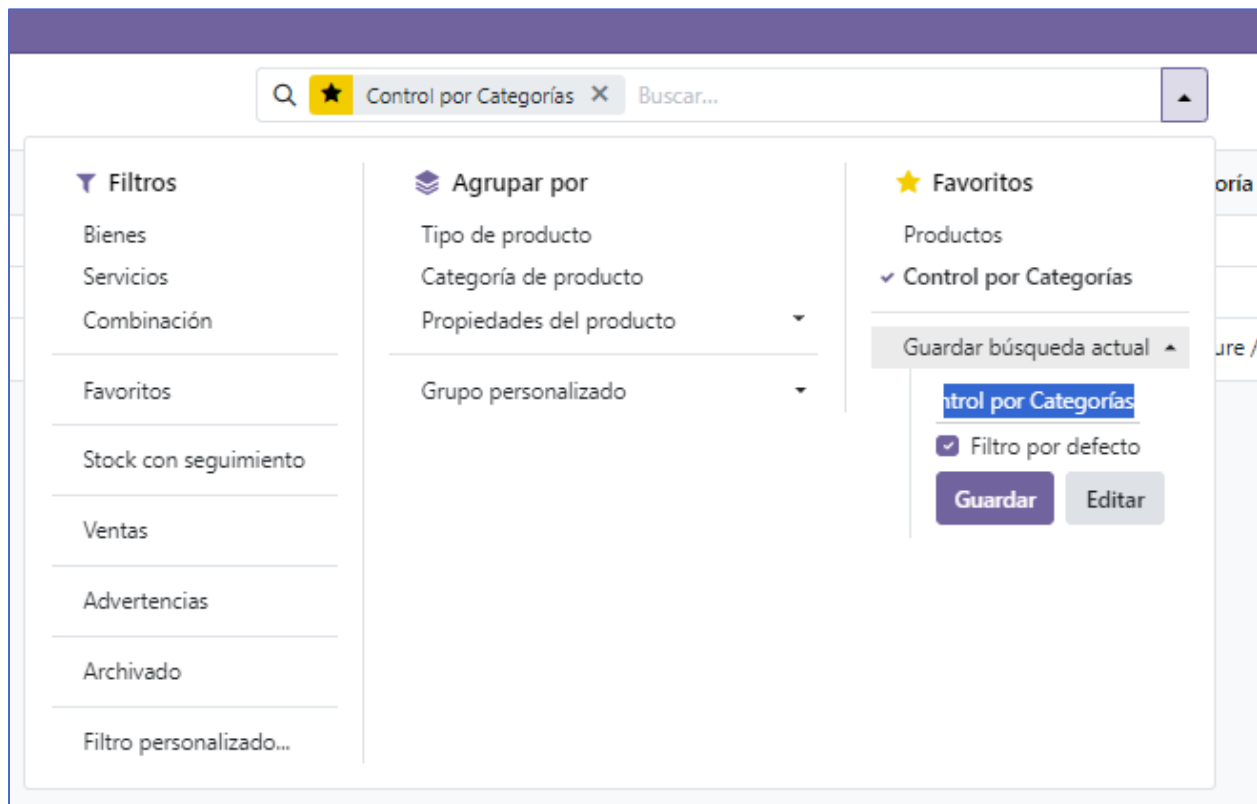
Nombre del producto	Referencia interna	Precio de venta	Coste	Categoría de producto	Stock real	Pronosticado
Alfombrilla de escritorio	FURN_0002	1,98 €	0,00 €	Furniture / Office	0,00	0,00
Armario con puertas	E-COM11	140,00 €	120,50 €	Furniture / Office	33,00	153,00
Caja de administración de cables	FURN_5555	100,00 €	70,00 €	Furniture / Office	90,00	90,00
Caja de almacenamiento	E-COM08	15,80 €	14,00 €	Furniture / Office	18,00	18,00
Cajón	FURN_8855	110,50 €	100,00 €	Furniture / Office	80,00	65,00
Cajón negro	FURN_8900	25,00 €	20,00 €	Furniture / Office	0,00	0,00
Combinación de escritorio	FURN_7800	450,00 €	300,00 €	Furniture / Office	2,855,00	2,799,00
Cubo de basura con pedal	E-COM10	47,00 €	10,00 €	Furniture / Office	22,00	22,00
Escritorio en L con asiento a la derecha	E-COM06	147,00 €	600,00 €	Furniture / Office	0,00	0,00
Escritorio en L con asiento a la izquierda	FURN_1118	85,00 €	78,00 €	Furniture / Office	2,00	2,00
Escritorio grande		1,799,00 €	0,00 €	Furniture / Office	0,00	0,00
Escritorio para cuatro personas	FURN_8320	2,350,00 €	2,500,00 €	Furniture / Office	0,00	0,00
Escritorio personalizable		750,00 €	0,00 €	Furniture / Office	150,00	150,00
Gabinete grande	E-COM07	320,00 €	800,00 €	Furniture / Office	500,00	270,00
Lugar de trabajo individual	FURN_0789	885,00 €	876,00 €	Furniture / Office	16,00	16,00
Lámpara de oficina	FURN_8888	40,00 €	35,00 €	Furniture / Office	0,00	10,00

Usamos la **barra de búsqueda avanzada** para la agrupación.

Desplegamos el menú de filtros y **aplicamos "Agrupar por"**. Seleccionamos **"Categoría de producto"**. El resultado fue excelente. Vimos un listado colapsado y ordenado por familias. Era limpio y fácil de auditar.



Aplicamos una regla. Pensar en el usuario. El jefe de compras no debe repetir pasos cada mañana. **Guardamos esta configuración en "Favoritos"** con el nombre **"Control por Categorías"**. La marcamos como **vista por defecto**.



Vista Pivot para análisis de volumen (La foto global)

La **Vista Lista** resuelve el **detalle**. No ofrece la comparativa global. **Configuramos una Vista Pivot** mediante el icono de **tabla dinámica**.

Estructuramos la tabla para responder a la gerencia:

Añadimos el campo "**Categoría de producto**" en el eje vertical.

Seleccionamos el "**Numero**" de artículos y la "**Cantidad**" en las medidas.

The screenshot shows the Odoo Pivot View interface for 'Historial de movimientos'. The main table displays data for 'Total' and 'Furniture / Office' categories, with columns for 'Cantidad en la unidad de medida del producto' and 'Número'. The interface includes a search bar, a filter panel on the right, and a favorites panel on the far right.

	Cantidad en la unidad de medida del producto	Número
Total	3.848,00	17
Furniture / Office	3.848,00	17

Esta vista funciona como un **reemplazo de Excel** dentro de Odoo. Da al responsable de compras una herramienta potente. Detecta anomalías de un vistazo. Quizá la categoría "Gastos misceláneos" tiene 500 productos. Sabe que hay un problema de clasificación. Puede desplegar la celda e investigar.

Guardamos la vista en favoritos y ya estamos listos para continuar.

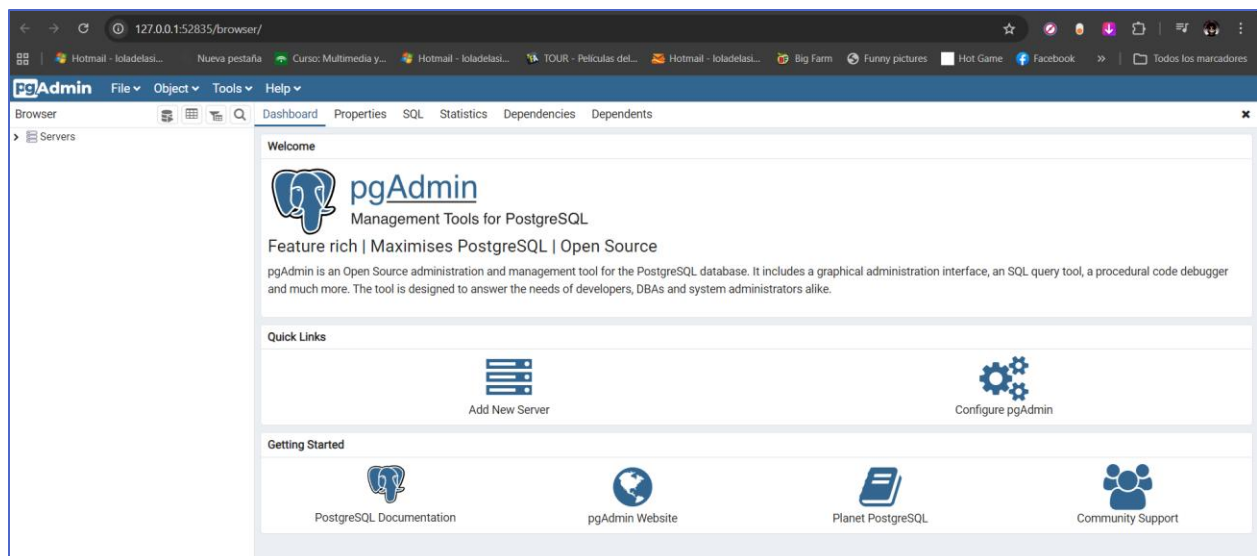
04

FASE 2: ANÁLISIS DE BASE DE DATOS Y EXTRACCIÓN SEGURA DE INFORMACIÓN (POSTGRESQL)

Terminamos la parte visual en el ERP para el jefe de compras. Luego abordamos el núcleo del caso práctico. Preparamos los datos para el informe externo en Jaspersoft Studio. Dejamos la interfaz web de Odoo y **conectaremos** directamente al motor de base de datos con **pgAdmin 4**.

1. Vamos a abrir pgAdmin 4 (Nuestra herramienta principal)

Buscaremos "**pgAdmin 4**" en el menú de **inicio de Windows** para explorar los datos. Durante el curso vimos las consultas por consola.

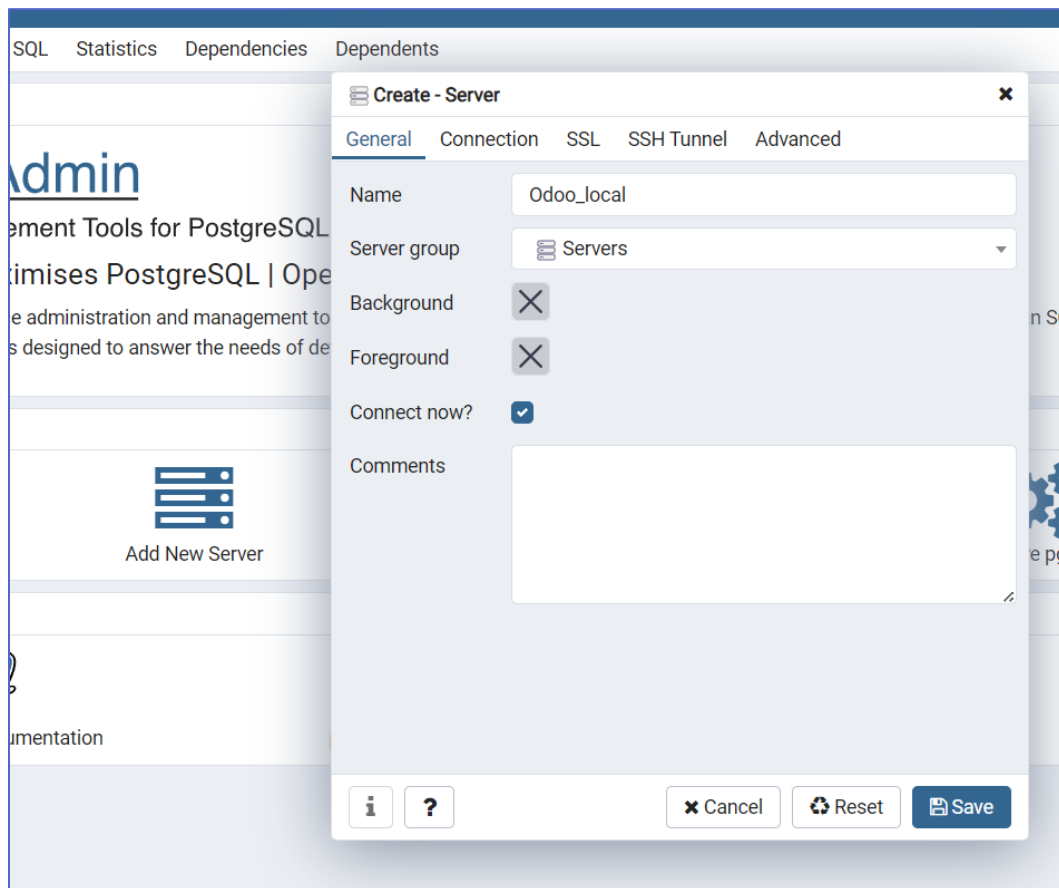


La aplicación pedirá probablemente una contraseña maestra (**Master Password**) al abrirse. Esta medida de seguridad cifra las contraseñas guardadas de los servidores. Si no recordamos la clave, podemos intentar recordar la original de la instalación. También podemos resetear el entorno si es necesario. Superamos este control y estaremos listos para conectar con Odoo.

2. Conectamos nuestro servidor (Acceso directo a Odoo)

El objetivo es evitar la capa web de Odoo (el ORM) y **conectar directamente al sistema interno**. Miramos la columna izquierda (Browser) en la interfaz principal de **pgAdmin** para crear la conexión.

Hacemos clic derecho en el nodo principal **Servers > Create > Server...**



Asignamos un nombre descriptivo en la pestaña **General** para identificarlo luego (por ejemplo, **Odoo_Local**). Este nombre sirve solo como referencia visual.

Vamos a la pestaña **Connection** e introducimos los parámetros técnicos siguientes:

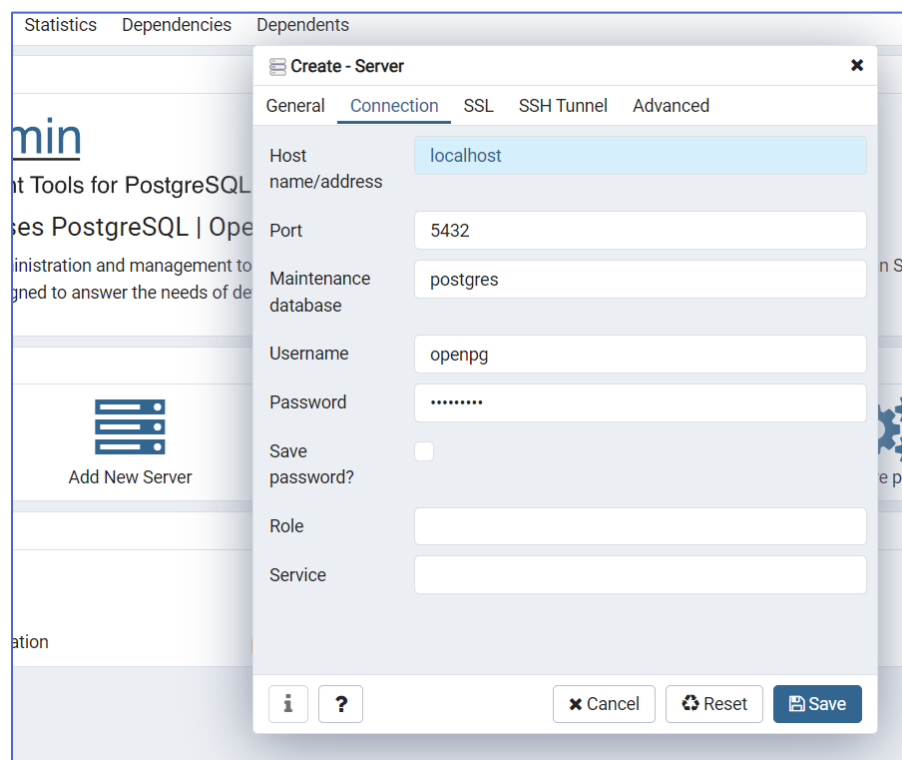
Host name/address: localhost (el motor de PostgreSQL se ejecuta localmente en el PC).

Port: 5432 (este puerto predeterminado de PostgreSQL se verifica en los servicios de Windows).

Maintenance database: postgres (la base de datos administrativa predeterminada).

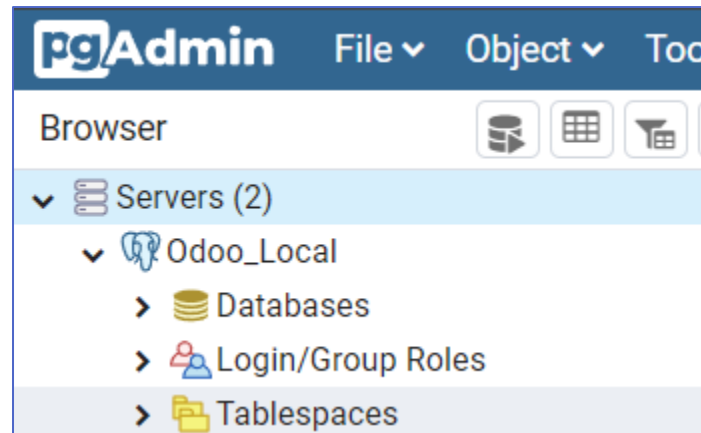
Username: openpg. Prestad atención a este error común. El instalador "todo en uno" de Odoo en Windows crea el usuario openpg por seguridad. Odoo no usa el usuario postgres en este caso. Probad con postgres solo si la instalación fue manual e independiente.

Password: openpgpwd. Esta es la contraseña cifrada predeterminada del instalador de Odoo. Usad la contraseña propia si la cambiasteis durante la instalación.



3. Localizamos nuestra base de datos en el esquema

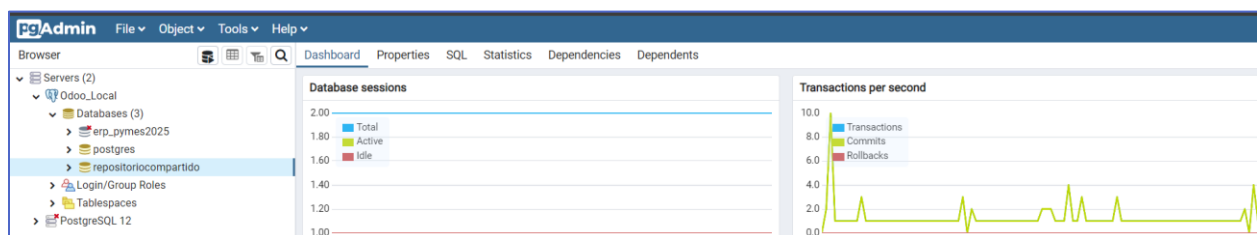
El icono del servidor se vuelve azul si los datos son correctos. El menú izquierdo se despliega y da acceso total al motor relacional.



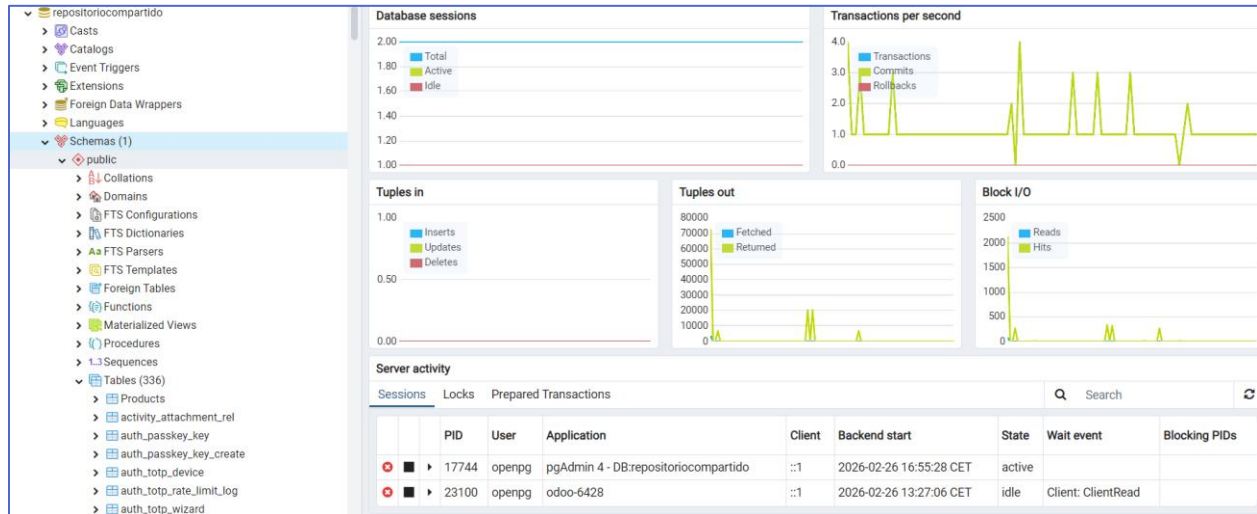
Debemos navegar por la jerarquía de PostgreSQL para encontrar los datos de prueba:

Desplegamos la **ruta Servers > Odoo_Local > Databases**.

Veremos allí varias bases de datos. Buscamos la instancia creada anteriormente con los datos de demostración: **repositoriocompartido**.



Desplegamos esa base y seguimos por la arquitectura: **Schemas > public > Tables**.



Nota de DAM (El impacto visual del ORM): Encontraremos más de **800 o 900 tablas al hacer clic en Tables**. Esta cantidad puede intimidar al principio. Odoo es un sistema muy modular. Su ORM genera tablas automáticamente para todo (mensajes, adjuntos, configuraciones web).

No debemos preocuparnos. **Ignoraremos los datos irrelevantes** para montar el informe del departamento de compras. **Usaremos el buscador del panel superior para filtrar**. Nos centraremos en las tablas con el **prefijo product_ (como product_template o product_category)**. Estas contienen la información del inventario.

Search Objects - repositoriocompartido/openpg@Odoo_Local

product_ All types Search

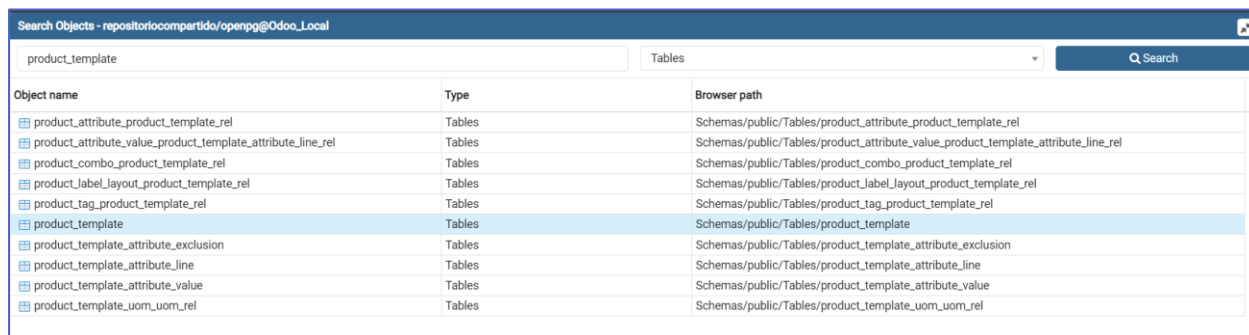
Object name	Type	Browser path
✓ product_attribute_check_multi_chec...	Check Constraints	Schemas/public/Tables/product_attribute/Constraints/product_att
auto_print_product_labels	Columns	Schemas/public/Tables/stock_picking_type/Columns/auto_print_p
custom_product_template_attribute...	Columns	Schemas/public/Tables/product_attribute_custom_value/Columns
group_product_pricelist	Columns	Schemas/public/Tables/res_config_settings/Columns/group_prod.
group_product_variant	Columns	Schemas/public/Tables/res_config_settings/Columns/group_prod.
group_stock_production_lot	Columns	Schemas/public/Tables/res_config_settings/Columns/group_stoc.
module_product_expiry	Columns	Schemas/public/Tables/res_config_settings/Columns/module_pro
product_attribute_id	Columns	Schemas/public/Tables/product_attribute_product_template_rel/C
product_attribute_value_id	Columns	Schemas/public/Tables/product_attribute_value_product_template
product_attribute_value_id	Columns	Schemas/public/Tables/product_template_attribute_value/Column
product_categ_selectable	Columns	Schemas/public/Tables/stock_route/Columns/product_categ_sele
product_code	Columns	Schemas/public/Tables/product_supplierinfo/Columns/product_c.
product_combo_id	Columns	Schemas/public/Tables/product_combo_product_template_rel/Col

370 matches found.

Close

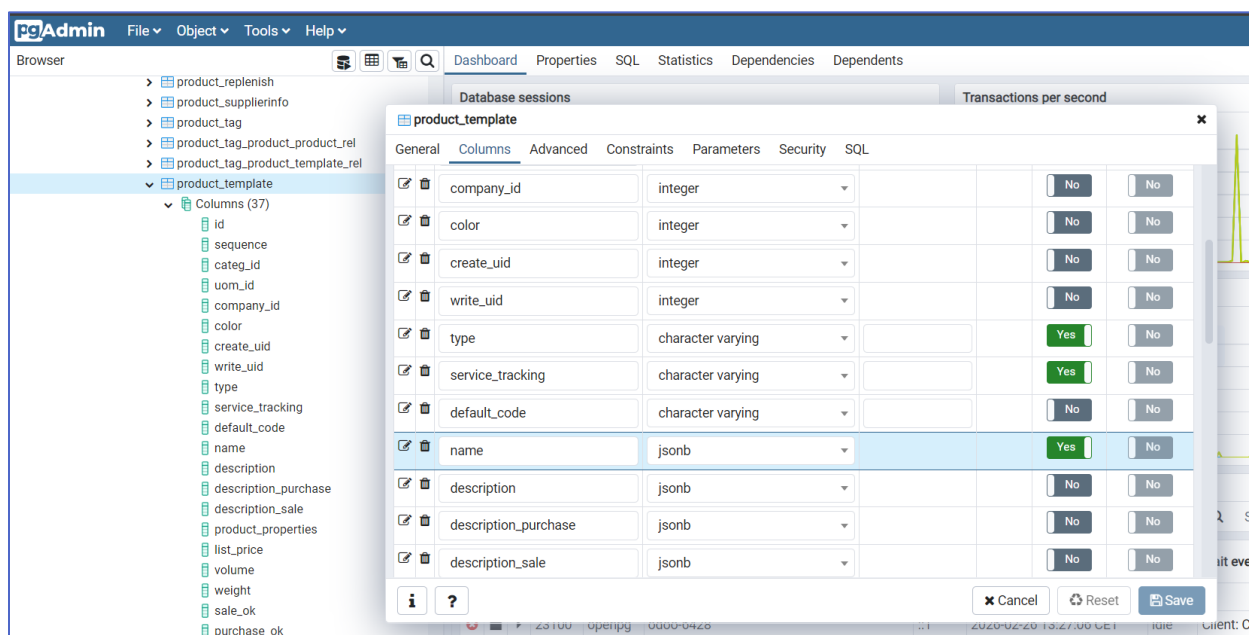
Localizamos tres tablas fundamentales para el informe:

product_template: Esta es la tabla maestra. Normalmente se suele buscar los productos en **product_product** por error, pero esa tabla gestiona variantes. La **información general reside en product_template**.



Object name	Type	Browser path
product_attribute_product_template_rel	Tables	Schemas/public/Tables/product_attribute_product_template_rel
product_attribute_value_product_template_attribute_line_rel	Tables	Schemas/public/Tables/product_attribute_value_product_template_attribute_line_rel
product_combo_product_template_rel	Tables	Schemas/public/Tables/product_combo_product_template_rel
product_label_layout_product_template_rel	Tables	Schemas/public/Tables/product_label_layout_product_template_rel
product_tag_product_template_rel	Tables	Schemas/public/Tables/product_tag_product_template_rel
product_template	Tables	Schemas/public/Tables/product_template
product_template_attribute_exclusion	Tables	Schemas/public/Tables/product_template_attribute_exclusion
product_template_attribute_line	Tables	Schemas/public/Tables/product_template_attribute_line
product_template_attribute_value	Tables	Schemas/public/Tables/product_template_attribute_value
product_template_uom_uom_rel	Tables	Schemas/public/Tables/product_template_uom_uom_rel

Revisamos sus columnas. Identificamos los campos exigidos: **nombre (name)**, **referencia interna (default_code)** y **fecha de alta (create_date)**. También localizamos las **claves foráneas** para cruzar datos: **categ_id** y **create_uid**.



Column	Type	Primary	Not Null
company_id	integer	No	No
color	integer	No	No
create_uid	integer	No	No
write_uid	integer	No	No
type	character varying	Yes	No
service_tracking	character varying	Yes	No
default_code	character varying	No	No
name	jsonb	Yes	No
description	jsonb	No	No
description_purchase	jsonb	No	No
description_sale	jsonb	No	No

product_category: Esta tabla guarda el árbol de categorías. La tabla principal solo contiene el número de identificación. Cruzaremos los datos con su clave primaria (id) contra el categ_id de los productos. Así obtendremos el nombre en texto legible.

res_users: Necesitamos esta tabla para identificar al empleado responsable. La enlazaremos mediante el campo create_uid de la tabla de productos. Extraeremos el campo login. Este dato suele ser el correo corporativo o el nombre de usuario.

Securización del entorno: Creación de un usuario de solo lectura:

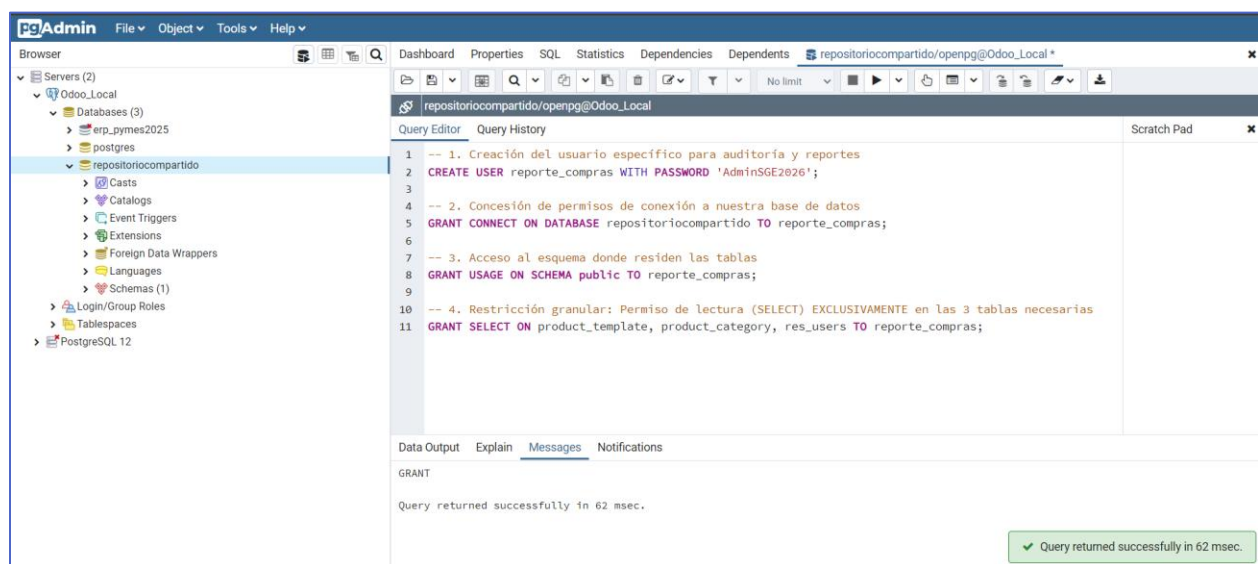
Abordamos la seguridad antes de escribir código para extraer información.

Usar credenciales de **superusuario** para herramientas externas de Business Intelligence es una **mala práctica**. Podría ocurrir un fallo de configuración o una inyección SQL. Esto borraría o dañaría toda la base de datos del ERP.

Abrimos la **herramienta de consultas de pgAdmin** para proteger los datos.

Diseñamos un **script SQL** para crear un **perfil restringido**. El informe externo **solo debe leer datos**. Jamás tendrá permisos para escribir (**INSERT**), modificar (**UPDATE**) o borrar (**DELETE**):

Abrimos la **Query Tool** y ejecutamos el script de SQL



Esta configuración asegura que el jefe de compras reciba sus datos cada mes. La integridad y seguridad del sistema Odoo permanecen intactas.

Construcción de la consulta SQL relacional

El entorno ya era seguro. El último paso de esta. La fase fue **diseñar la consulta SQL para el Dataset de Jaspersoft**. El objetivo era obtener una **tabla plana y limpia**. Esta tabla debía ser comprensible **para el usuario final**.

Tomamos decisiones de diseño importantes sobre los **JOINS**:

Usamos un INNER JOIN para las categorías. Odoo obliga a que todo producto tenga una categoría por regla de negocio. El sistema asigna la categoría "All" por defecto si falta una. No existe riesgo de perder registros con este cruce.

Implementamos un LEFT JOIN para cruzar con los usuarios (**res_users**). Esta decisión previene problemas futuros. La empresa podría importar un catálogo antiguo mediante Excel sin el creador del registro. Un **INNER JOIN** ocultaría esos productos del informe. Eso falsearía el stock real ante el jefe de compras. **El LEFT JOIN evita este riesgo**. El producto aparecerá en el listado. La columna del responsable quedará en blanco si hace falta.

Añadimos una cláusula **de ordenación (ORDER BY)**. Priorizamos la **categoría** y luego el **producto** alfabéticamente. Esto **facilita el trabajo posterior en Jaspersoft Studio**. El orden es necesario y no estético. El motor de bandas de Jaspersoft fallará si los datos no vienen agrupados desde la base de datos.

La consulta final fue ejecutada y validada en pgAdmin:

Ejecutamos la prueba en el Query Tool de pgAdmin.

repositoriocompartido/openpg@Odoo_Local

Query Editor

Query History

Scratch Pad

1

SELECT

2

pc.name AS categoria,

3

pt.name AS producto,

4

pt.default_code AS referencia,

5

pt.create_date AS fecha_alta,

6

ru.login AS responsable

7

FROM

8

product_template pt

9

INNER JOIN

10

product_category pc ON pt.categ_id = pc.id

11

LEFT JOIN

12

res_users ru ON pt.create_uid = ru.id

13

ORDER BY

14

pc.name ASC,

15

pt.name ASC;

Data Output

Explain

Messages

Notifications

	categoria	producto	referencia	fecha_alta	responsable
	character varying	jsonb	character varying	timestamp without time zone	character varying
1	Expenses	{'en_US': 'Hotel Accommodation', 'es_ES': 'Alojamiento d...	[null]	2026-02-26 13:01:44.639988	__system__
2	Expenses	{'en_US': 'Restaurant Expenses', 'es_ES': 'Gastos de rest...	[null]	2026-02-26 13:01:44.639988	__system__
3	Home Construction	{'en_US': 'Furniture Assembly', 'es_ES': 'Montaje de mue...	[null]	2026-02-26 13:01:44.639988	__system__
4	Office	{'en_US': 'Acoustic Bloc Screens', 'es_ES': 'Pantallas de ...	[null]	2026-02-26 13:01:44.639988	__system__
5	Office	{'en_US': 'Cabinet with Doors', 'es_ES': 'Armario con pue...	E-COM11	2026-02-26 13:01:44.639988	__system__

La consola devolvió un listado correcto de los productos de demostración de Odoo.

Tradujimos las IDs internas de la base de datos.

Obtuvimos nombres reales de categorías y correos de usuarios.

Los datos quedaron extraídos y refinados.

Teníamos la base lista para ir a Jaspersoft.

El siguiente paso era maquetar el PDF definitivo.

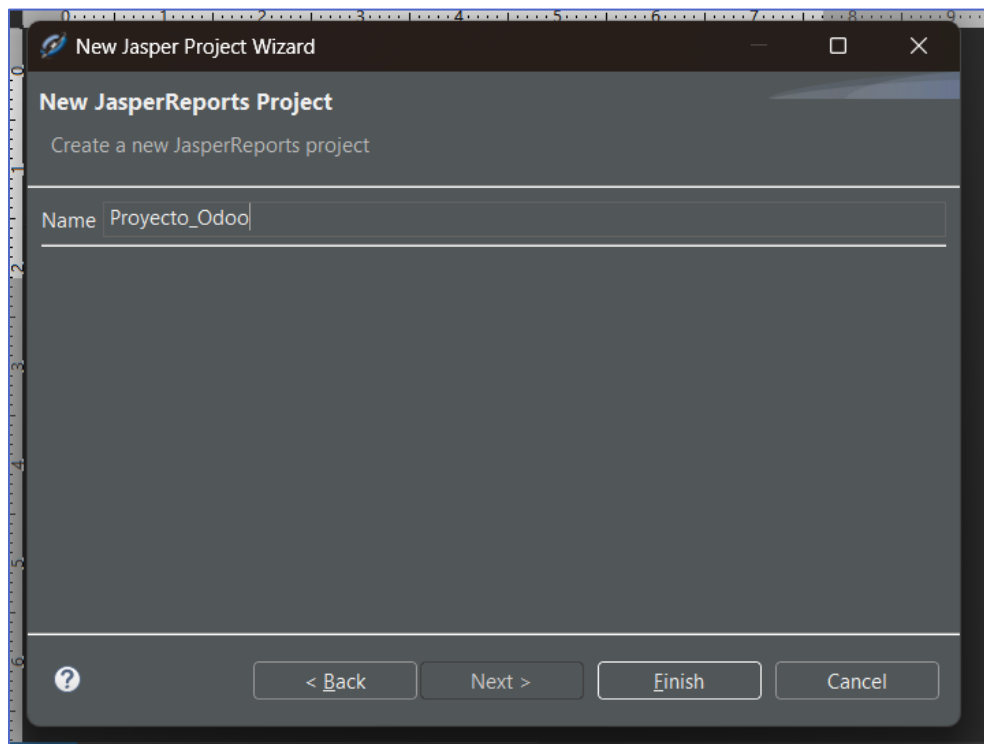
05

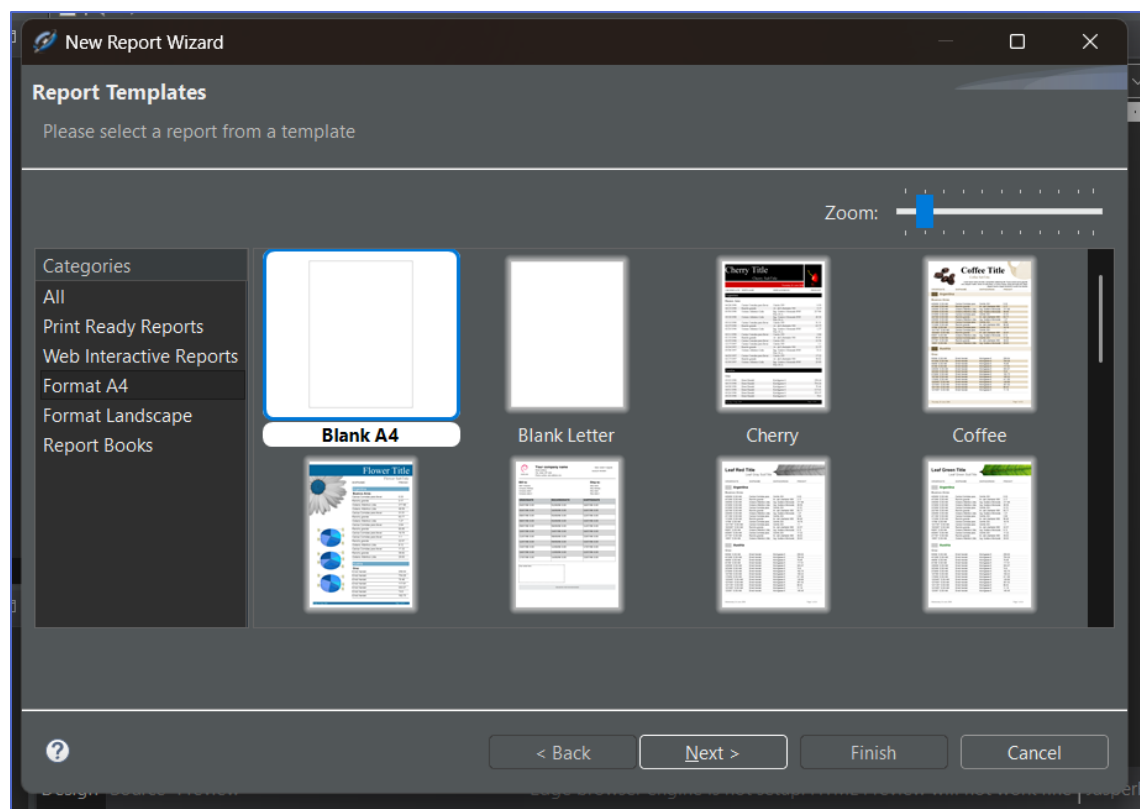
FASE 3: DISEÑO Y AUTOMATIZACIÓN DEL REPORTE EN JASPERSTUDIO

La base de datos estaba protegida y la consulta SQL funcionaba en pgAdmin. El último reto consistía en **conectar Odoo con** una herramienta de Business Intelligence externa. El objetivo era diseñar el PDF final para la gerencia. Usamos **Jaspersoft Studio** para esta tarea.

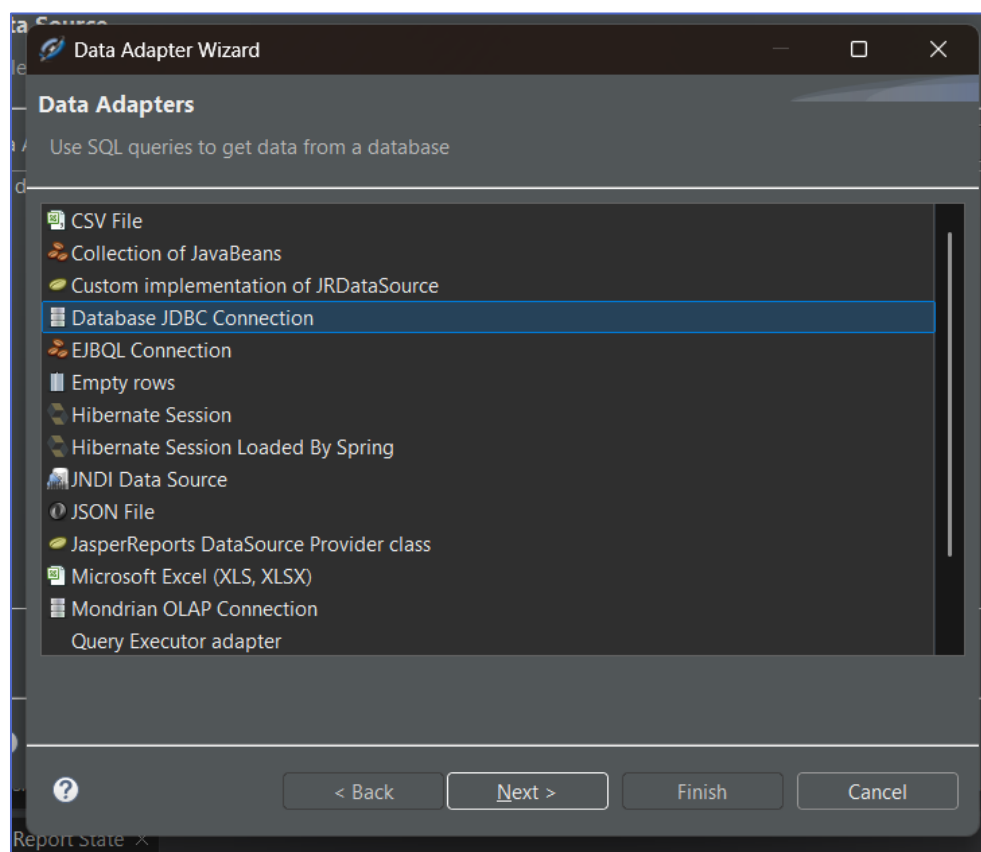
Configuración del Data Adapter (La conexión segura)

Abrimos **Jaspersoft** y creamos un informe nuevo con la plantilla Blank A4.





El archivo vacío necesitaba datos. Fuimos a la pestaña **Data Adapters** para establecer la **comunicación con PostgreSQL**. Adaptador tipo **JDBC**



Seguridad previa fue útil en este paso.

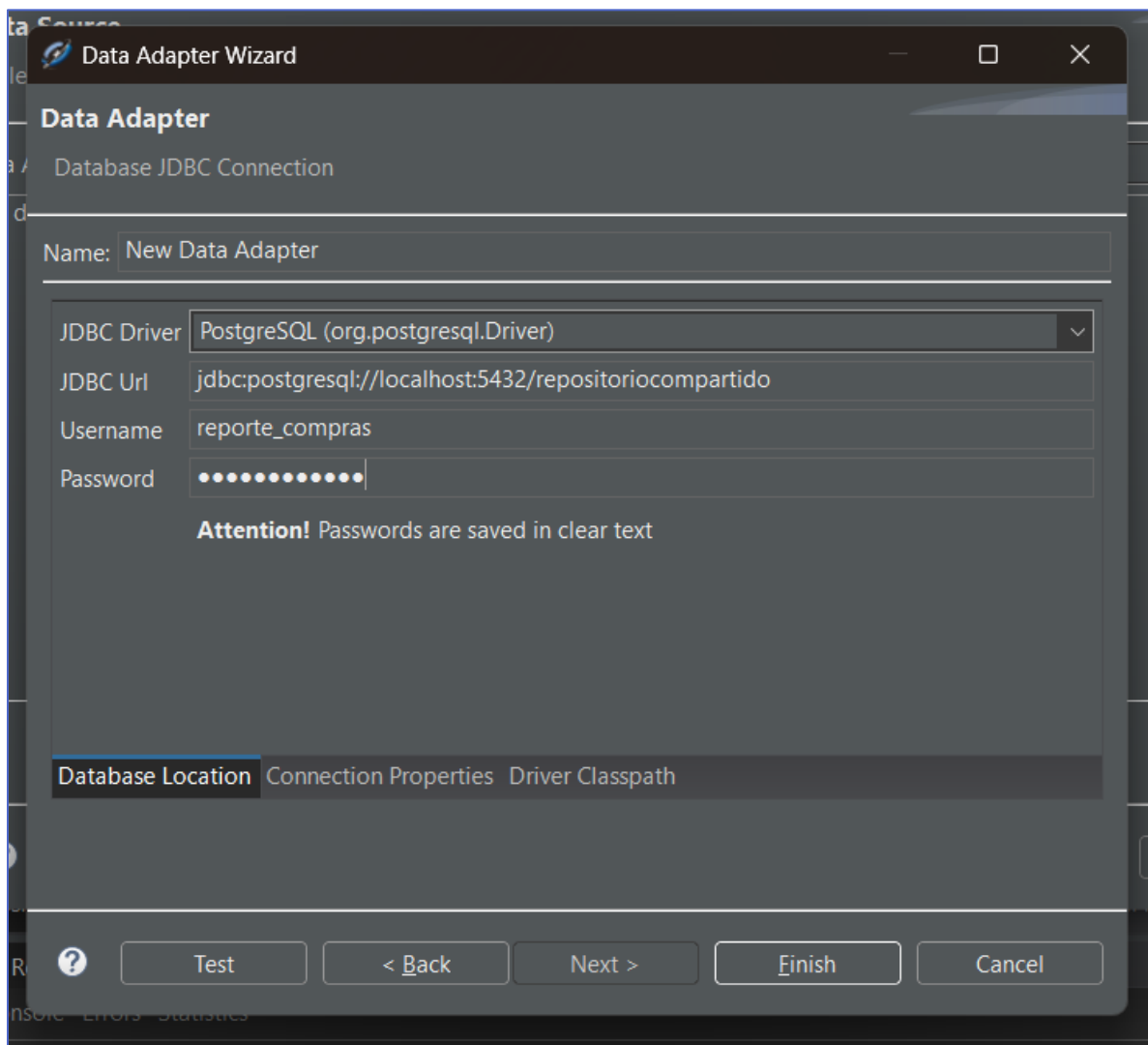
Seleccionamos el **controlador de PostgreSQL** en el asistente de conexión.

Introducimos la cadena **JDBC**:

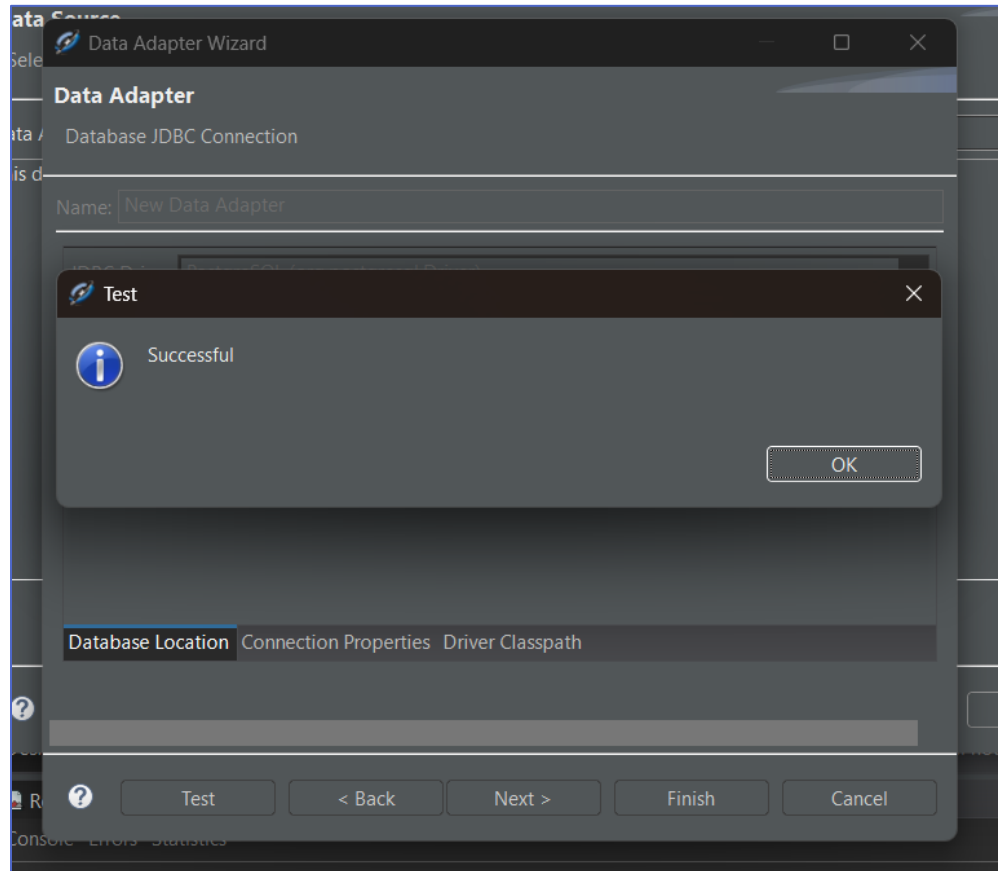
`jdbc:postgresql://localhost:5432/repositoriocompartido`.

No usamos el **usuario** administrador. Usamos el que creamos con el **script de SQL**

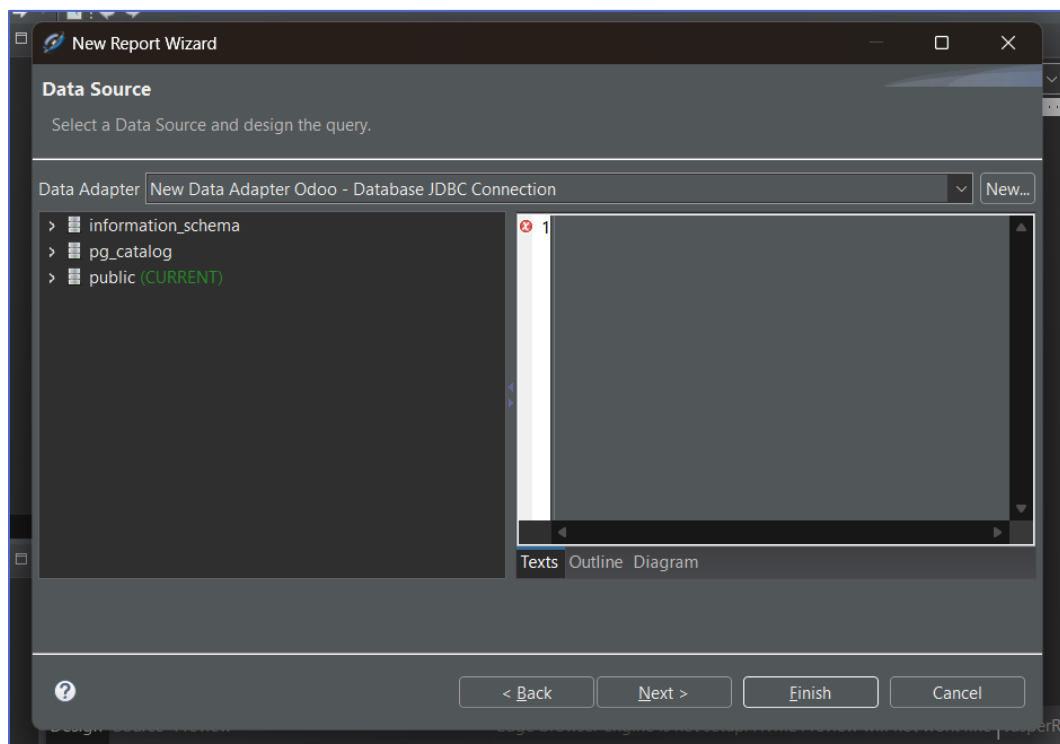
Configuramos la conexión con las **credenciales del usuario de solo lectura**, **'reporte_compras'**.



Probamos la conexión con el botón **"Test"** y funcionó.

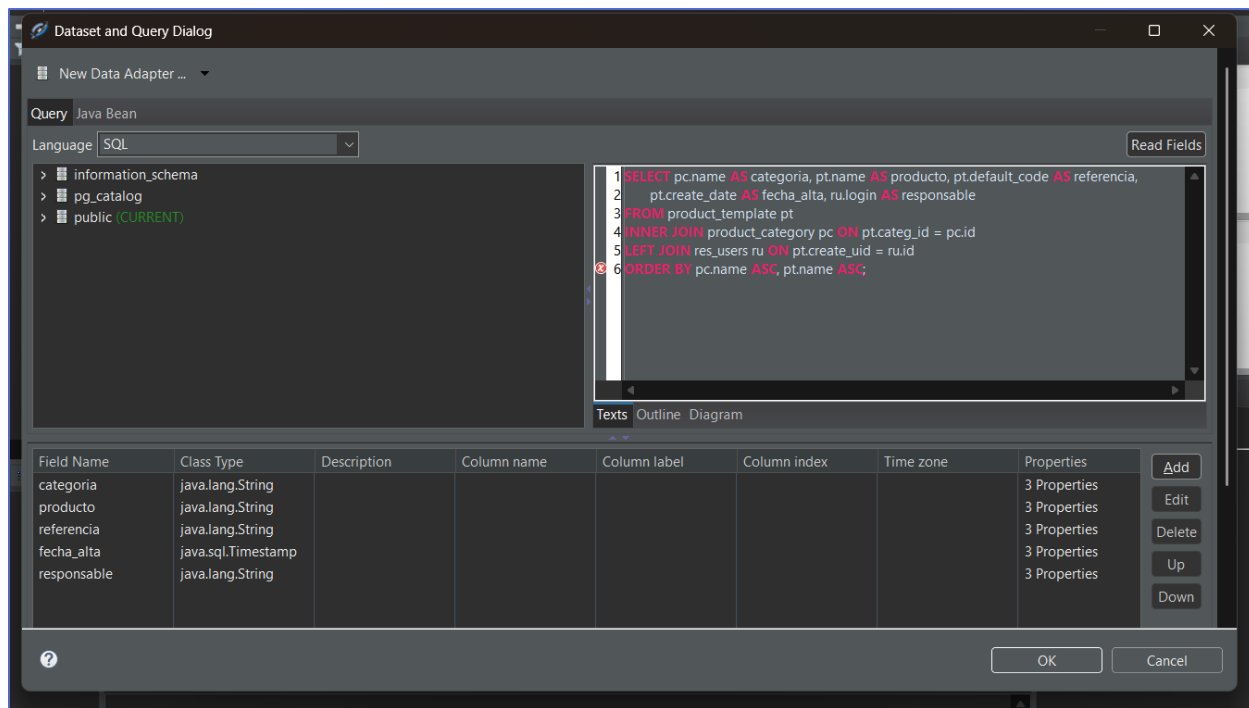


Esto nos dio tranquilidad. Es imposible dañar el inventario de Odoo durante el diseño.



Inyección de la consulta y mapeo de campos

Abrimos el diálogo **Dataset and Query** del informe. Seleccionamos el lenguaje **SQL** y pegamos la consulta con los JOINS probados en pgAdmin.



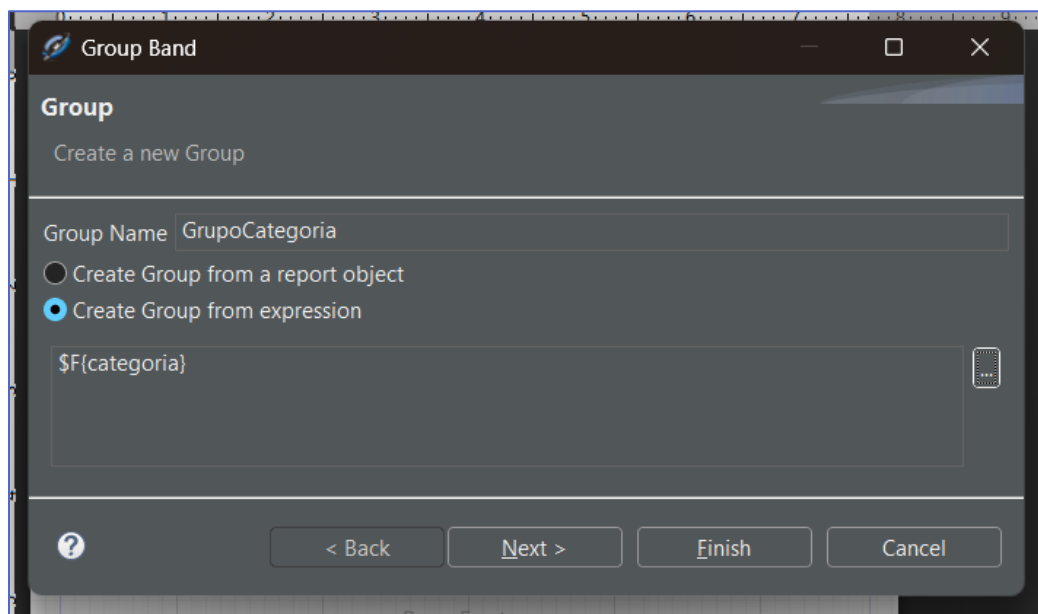
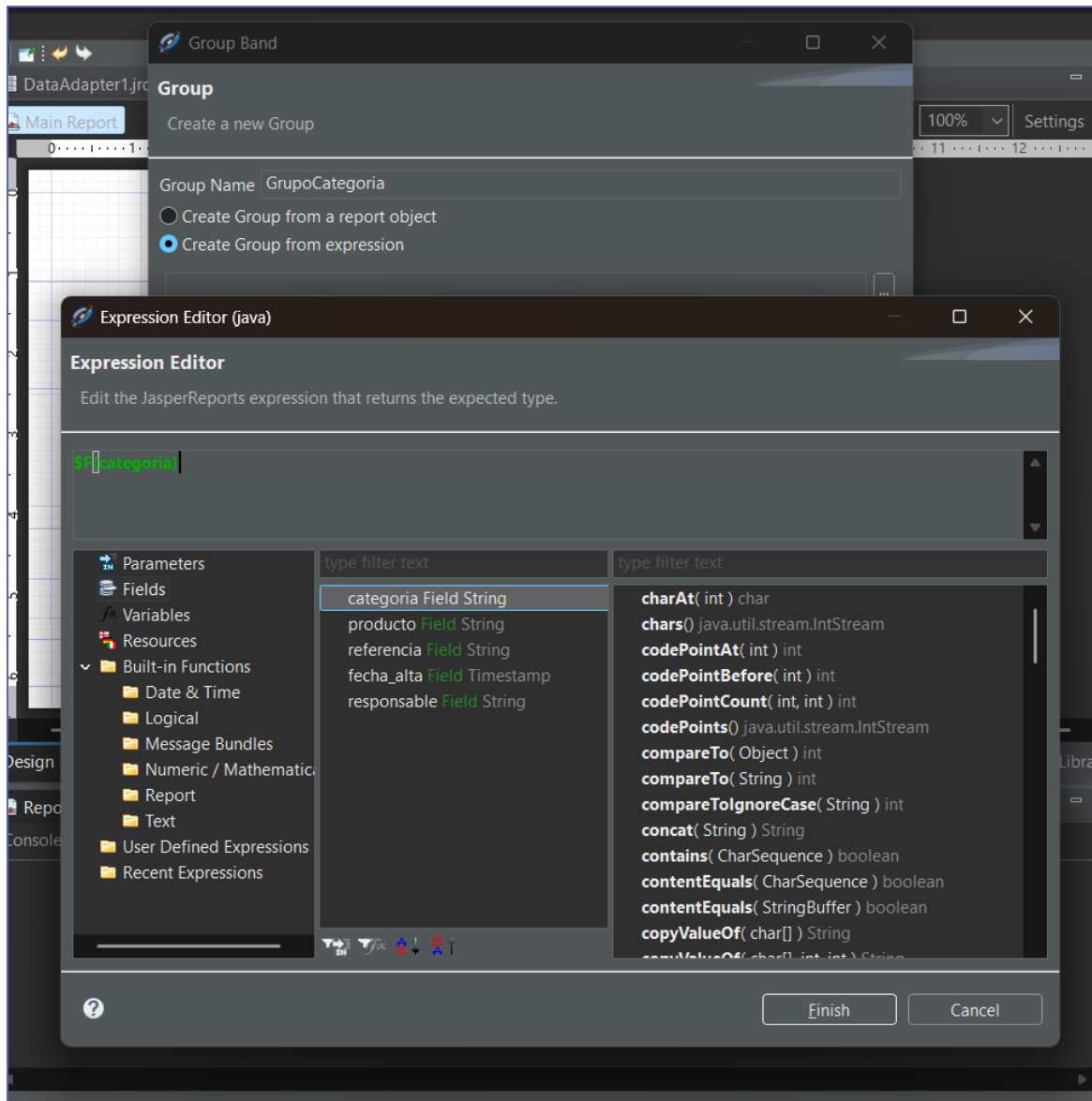
Jaspersoft se conectó y leyó los metadatos de Odoo.

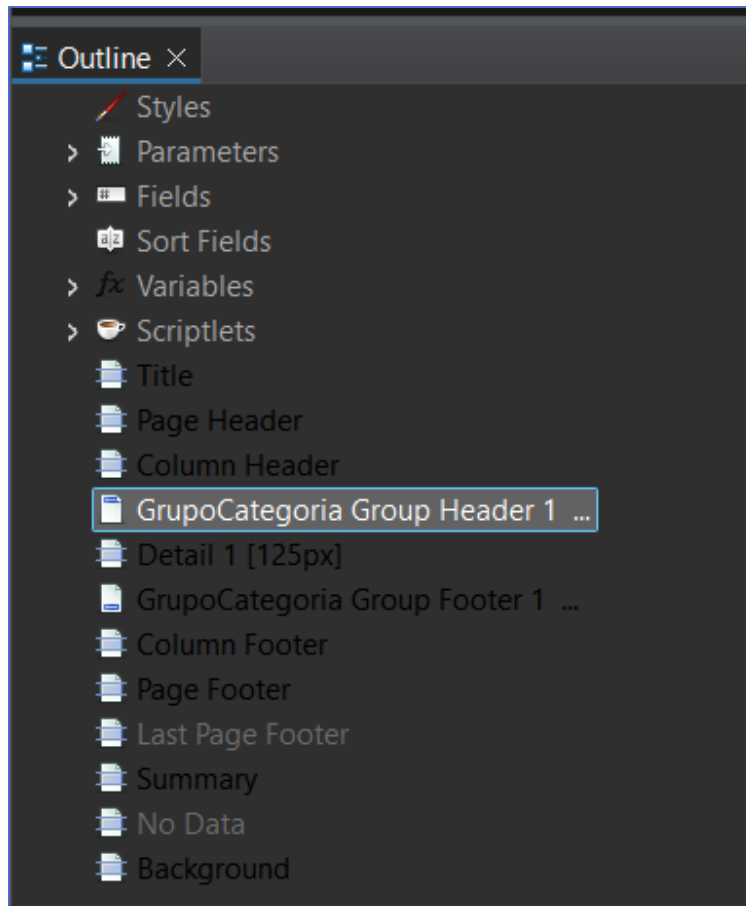
La pestaña inferior mostró los campos disponibles: **\$F{categoria}**, **\$F{producto}**, **\$F{referencia}**, **\$F{fecha_alta}** y **\$F{responsable}**. Los datos estaban listos para el lienzo.

El truco de la agrupación (Grupos en Jaspersoft)

El jefe de compras exigió que el informe se **agrupe por categorías**. Una lista plana no era aceptable. Jaspersoft requiere una configuración técnica específica para esto. Ordenar la consulta SQL no es suficiente. La interfaz debe saber cuándo dividir la lista y crear una nueva cabecera.

Fuimos a la **vista Outline** a la izquierda de la pantalla. Hicimos clic derecho sobre el informe y elegimos **"Create Group"**. Nombramos al grupo **GrupoCategoria**. Indicamos que el **criterio de ruptura fuera el campo \$F{categoria}**. Esta acción añadió dos bandas al diseño: **Group Header** y **Group Footer**.

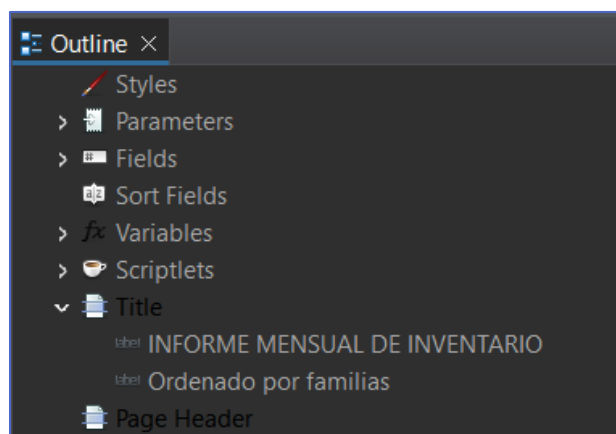




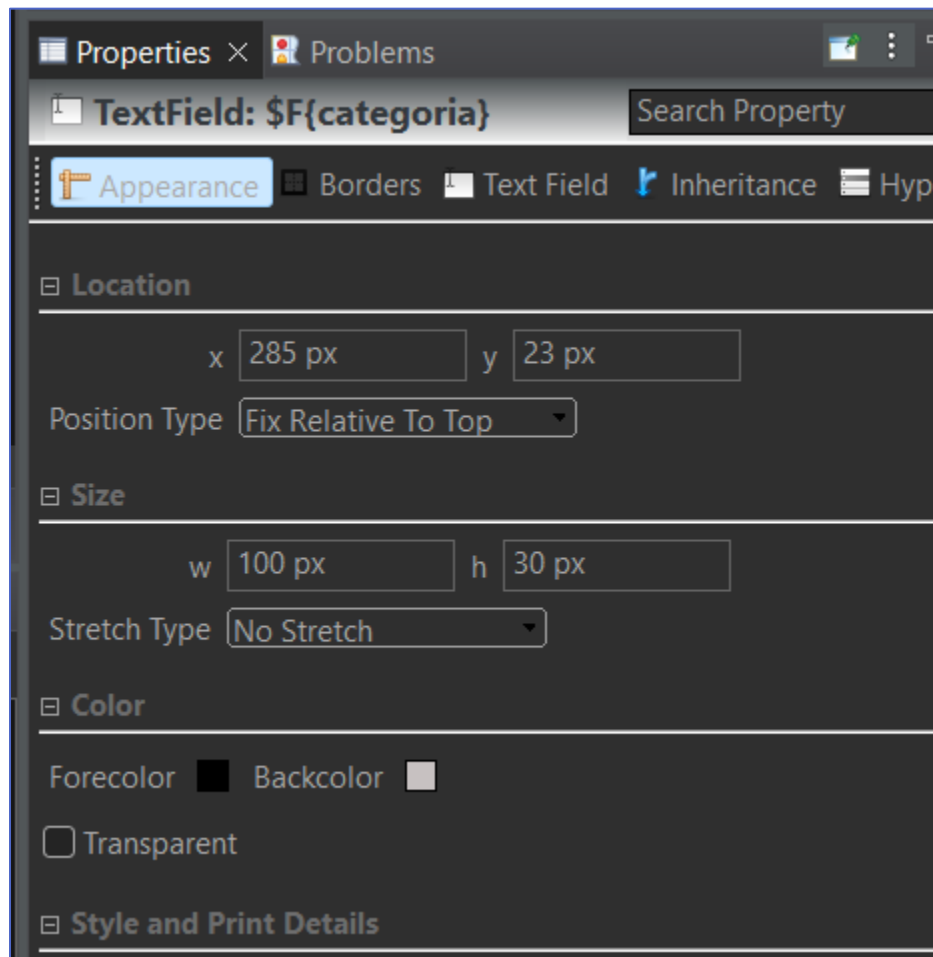
Maquetación y diseño visual (Evitando errores de novato)

Organizamos los elementos para lograr una apariencia profesional. Tomamos decisiones de diseño concretas:

Banda Title: Insertamos un texto estático con el título "**INFORME MENSUAL DE INVENTARIO**". Usamos una fuente grande. Añadimos un subtítulo que indicaba el orden del catálogo por familias.



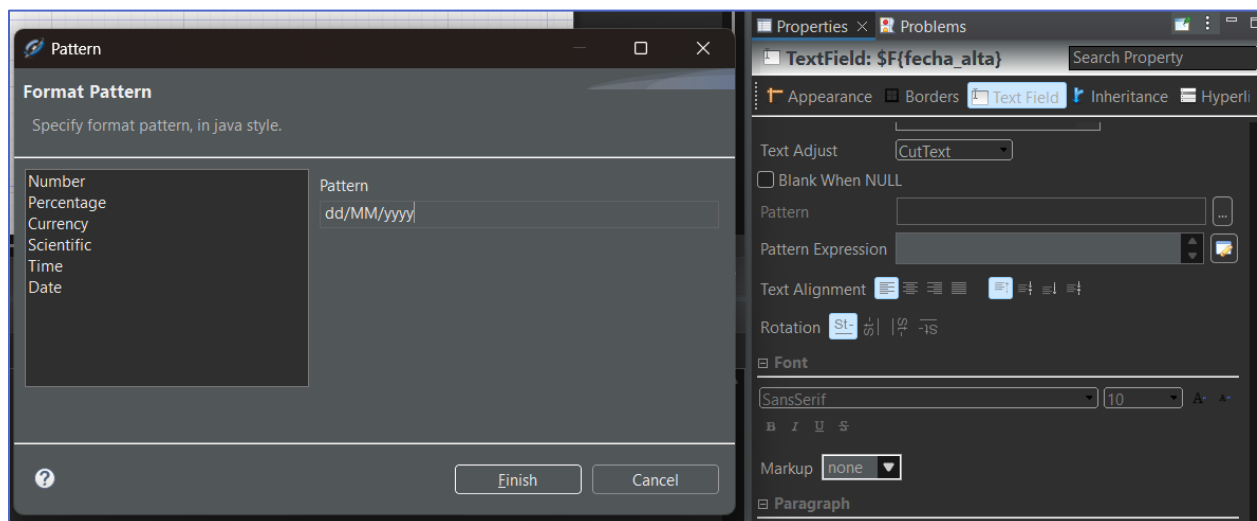
Banda Group Header (La clave visual): Arrastramos el campo **`$F{categoria}`** aquí. Aplicamos un fondo **gris claro**, **texto en negrita y un borde inferior**. Esto aclara visualmente el cambio de familia.



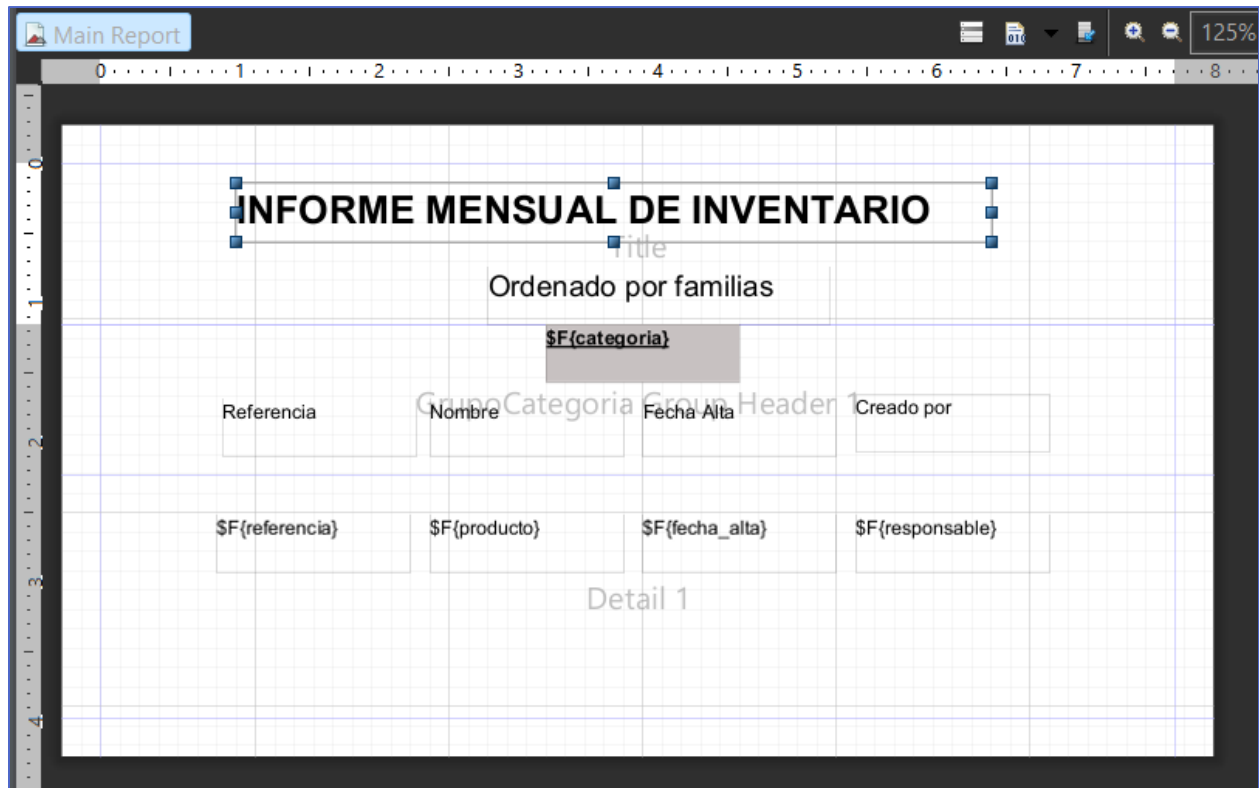
Nota de diseño: Colocamos las etiquetas de las columnas debajo del nombre de la **categoría** en esta banda. Incluimos **"Referencia", "Nombre", "Fecha Alta" y "Creado por"**. Ubicarlas aquí evita errores de visualización. Esto evita que las cabeceras se repitan en cada fila. Es un error común. También impide que queden descolgadas si una categoría nueva empieza a mitad de página.



Banda Detalle 1: Esta banda itera por cada registro de la base de datos. Alineamos los campos de texto dinámicos de izquierda a derecha. Ajustamos el formato del campo fecha (**\$F{fecha_alta}**). Eliminamos la marca de tiempo técnica con horas y milisegundos (2026-02-26 15:30:00.000). Usamos un formato legible (**dd/MM/yyyy**).

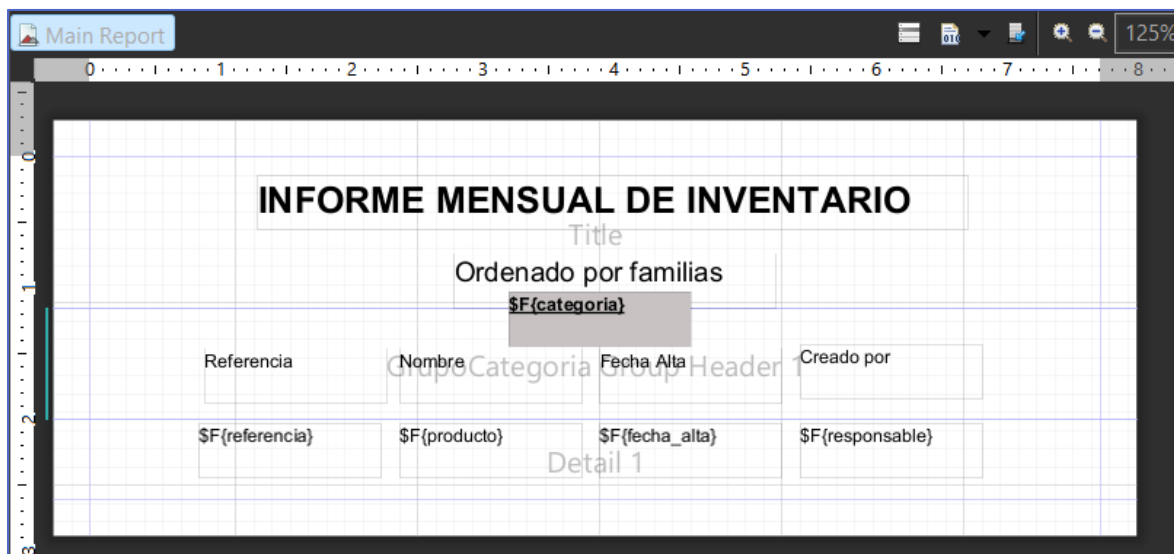


Limpieza final: Buscamos aprovechar el espacio. Queríamos evitar errores de paginación y hojas en blanco al final del PDF. **Eliminamos** todas las bandas sin uso. Esto incluyó **Page Header, Column Header, Column Footer, Page Footer, Summary y Background**



Compilación y resultado final

Ajustamos las distancias para aprovechar el espacio



Terminamos el diseño y pulsamos la pestaña "**Preview**". Jaspersoft compiló el archivo de diseño XML (.jrxml) a un binario ejecutable (.jasper). El sistema lanzó nuestra consulta con el usuario restringido contra el puerto 5432 de PostgreSQL. Procesó los datos. Nos devolvió un documento **PDF correcto**.

INFORME MENSUAL DE INVENTARIO

Ordenado por familias

Expenses

Referencia	Nombre	Fecha Alta	Creado por
null	{"en_US": "Hotel Accommodation",	26/02/2026	__system__
null	{"en_US": "Restaurant Expenses", "es_ES":	26/02/2026	__system__

Home Construction

Referencia	Nombre	Fecha Alta	Creado por
null	{"en_US": "Furniture Assembly", "es_ES":	26/02/2026	__system__

Office

Referencia	Nombre	Fecha Alta	Creado por
null	{"en_US": "Acoustic Bloc Screens",	26/02/2026	__system__
E-COM11	{"en_US": "Cabinet with Doors", "es_ES":	26/02/2026	__system__

Vimos el resultado. Aparecía una **franja gris** destacada, por ejemplo "Mobiliario de Oficina". Justo debajo estaban todos sus **productos tabulados**. El sistema detectaba el **cambio de valor** al terminar estos. Imprimía la siguiente franja gris, como "Consumibles Informáticos". La lista continuaba. Cumplimos el objetivo visual.

CONCLUSIONES

Este desarrollo dio una solución completa. Resolvimos un problema frecuente de los departamentos de compras en dos niveles.

A nivel Frontend, dimos agilidad al responsable en su día a día. Configuramos favoritos en las vistas Lista y Pivot de Odoo. El usuario maneja los datos. Ve el volumen de stock. Detecta anomalías al instante sin salir del navegador.

A nivel Backend/BI, aseguramos la fiabilidad de los datos a largo plazo para las reuniones. Construimos un informe mensual automatizado en Jaspersoft Studio. Un perfil de base de datos seguro respalda este reporte.

Este caso práctico demuestra un punto importante. Implantar un sistema de gestión empresarial (SGE) es más que instalar software y crear usuarios. El valor de un desarrollador está en entender el modelo relacional. Debe extraer la información de forma segura y transformarla. Gerencia se quejaba de "datos caóticos". Convertimos eso en un sistema de información sólido. Ahora está ordenado y listo para decidir.

REFERENCIAS

<https://www.odoo.com/documentation/19.0/es/applications.html>

<https://www.odoo.com/documentation/19.0/es/applications/productivity.html>

<https://www.postgresql.org/docs/18/index.html>

<https://www.postgresql.org/docs/current/sql-grant.html>

<https://neon.com/postgresql/tutorial>

<https://neon.com/postgresql/postgresql-tutorial/postgresql-joins>

Odoo 18 Tutorial: Create & Customize List, Form, Kanban & Calendar Views – Complete Guide

https://www.youtube.com/watch?v=aEhVCoF_S8E

Mini Curso Jasper Studio 02 Conexión a PostgreSQL y Ejemplo

<https://www.youtube.com/watch?v=yNL-B4p9XAE>

Configurar Conexión Data Adapter JasperSoft Studio - JasperReports

<https://www.youtube.com/watch?v=Z4ld7XpUn6Q>

<https://codigoxules.org/creando-informes-java-jasperreports-desde-jaspersoft-studio/>