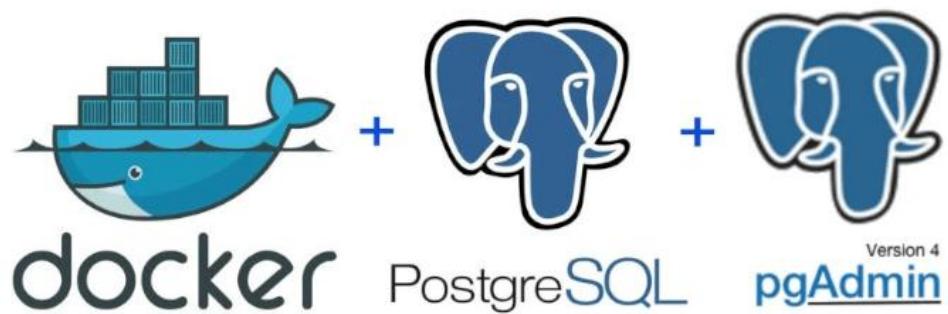


8 DE ENERO DE 2026



TAREA 4.1 - EXPLORACIÓN DE LA BASE DE DATOS DE ODOO CON PGADMIN (V2)

[SUBTÍTULO DEL DOCUMENTO]

Nombre: Daniel García Méndez

Módulo: Sistemas de gestión empresarial

Curso: 2º DAM

CONTENIDO

TAREA 4.1 - Exploración de la base de datos de Odoo con pgAdmin.....	2
1. Puesta en marcha y conexión con la BD	2
1.1. Comprueba que los contenedores están activos.....	2
1.2. Abre pgAdmin y crea una conexión al servidor PostgreSQL de Odoo	2
2. Exploración de la base de datos de Odoo.....	4
2.1. Localiza la base de datos que usas en clase.....	4
2.2. Explora las tablas	6
3. Interpretación de nombres y estructuras.....	8
4. Usuarios de la base de datos	8
5. Reflexión final	10
Webgrafía	10

TAREA 4.1 - EXPLORACIÓN DE LA BASE DE DATOS DE ODOO CON PGADMIN

En esta tarea vas a trabajar directamente con la base de datos de Odoo utilizando **pgAdmin**, un cliente gráfico para PostgreSQL. El objetivo es entender qué datos guarda el ERP, cómo se organizan sus tablas y quién tiene acceso a la información.

La tarea combina pequeñas pruebas prácticas y preguntas breves que te ayudarán a comprender cómo Odoo almacena y gestiona la información internamente.

1. PUESTA EN MARCHA Y CONEXIÓN CON LA BD

1.1. COMPRUEBA QUE LOS CONTENEDORES ESTÁN ACTIVOS

Antes de usar pgAdmin, asegúrate de que los contenedores de Odoo y PostgreSQL están arrancados:

```
docker ps
```

Haz una captura donde se vean los contenedores funcionando.

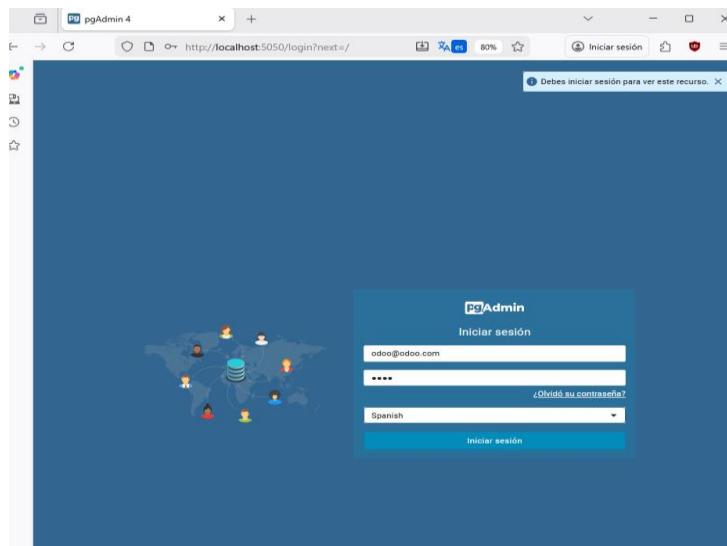
```
C:\odoo_docker>docker ps
CONTAINER ID        IMAGE               COMMAND             CREATED            STATUS              PORTS
9e18dc55acd        odoo:18              "/entrypoint.sh odoo"
2ba4976a2973        dpage/pgadmin4      "/entrypoint.sh"
46e04b9301d8        postgres:15          "docker-entrypoint.s...
1277a510fd9c        portainer/portainer-ce:latest   "/portainer"
                                                               4 weeks ago       Up 4 seconds
                                                               0.0.0.0:9000->9000/tcp, 0.0.0.0:9443->9443/tcp, 0.0.0.0:9443->9443/tcp
                                                               NAMES
                                                               odoo_app
                                                               pgadmin_container
                                                               odoo_db
                                                               portainer
C:\odoo_docker>
```

1.2. ABRE PGADMIN Y CREA UNA CONEXIÓN AL SERVIDOR POSTGRESQL DE ODOO

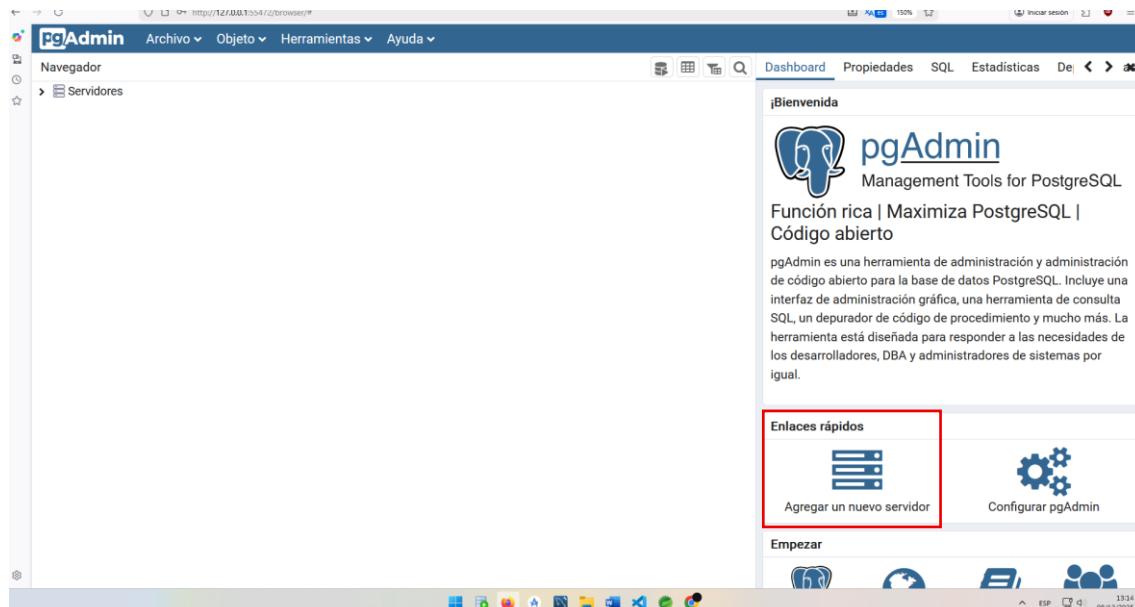
Añade un nuevo servidor con estos datos:

- **Nombre del servidor:** Odoo-Docker
- **Host:** localhost
- **Puerto:** 5432
- **Usuario:** odoo
- **Contraseña:** odoo

Captura obligatoria: pantalla de configuración o pantalla donde se vea el servidor conectado.



Iremos a la opción señalada para agregar un servidor y pondremos los datos anteriormente vistos en el apartado anterior:



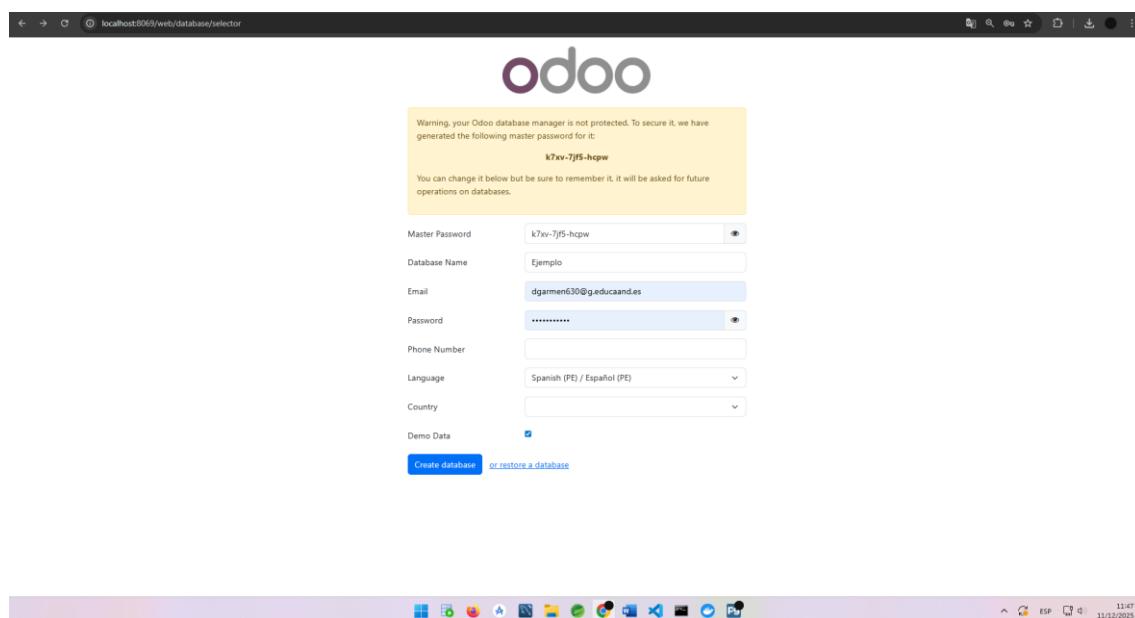
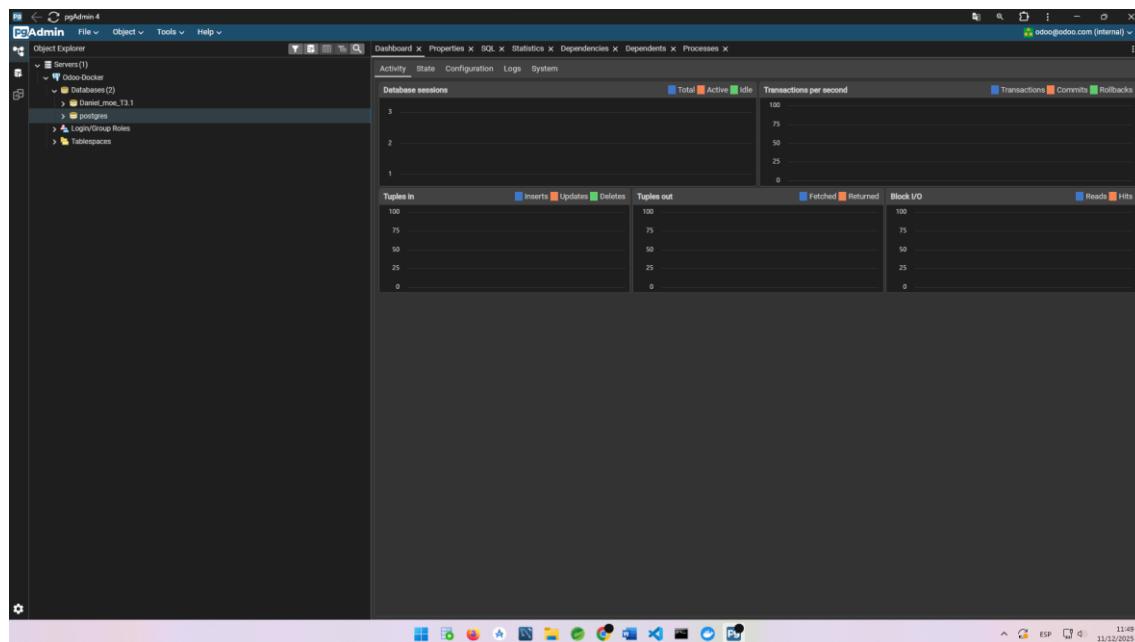
The screenshot shows the pgAdmin interface. In the top right, there's a welcome message: "¡Bienvenida" with the pgAdmin logo, followed by "Management Tools for PostgreSQL" and "Función rica | Maximiza PostgreSQL | Código abierto". Below this, a text block describes pgAdmin as a graphical administration tool for PostgreSQL. On the right, there's a section titled "Enlaces rápidos" with two buttons: "Agregar un nuevo servidor" (highlighted with a red box) and "Configurar pgAdmin". At the bottom, there's a "Empezar" button and some icons. The bottom part of the screenshot shows two overlapping dialog boxes: "Crear - Servidor" and "Crear - Conexión". The "Crear - Servidor" dialog has tabs for General, Conexión, SSL, Túnel SSH, and Avanzado. The "Nombre" field is filled with "Odoo-Dockef". The "Crear - Conexión" dialog also has tabs for General, Conexión, SSL, Túnel SSH, and Avanzado. The "Nombre/dirección del host" field is set to "localhost". Both dialogs have standard save and cancel buttons at the bottom.

2. EXPLORACIÓN DE LA BASE DE DATOS DE ODOO

2.1. LOCALIZA LA BASE DE DATOS QUE USAS EN CLASE

Despliega:

Servers → Odoo-Docker → Databases



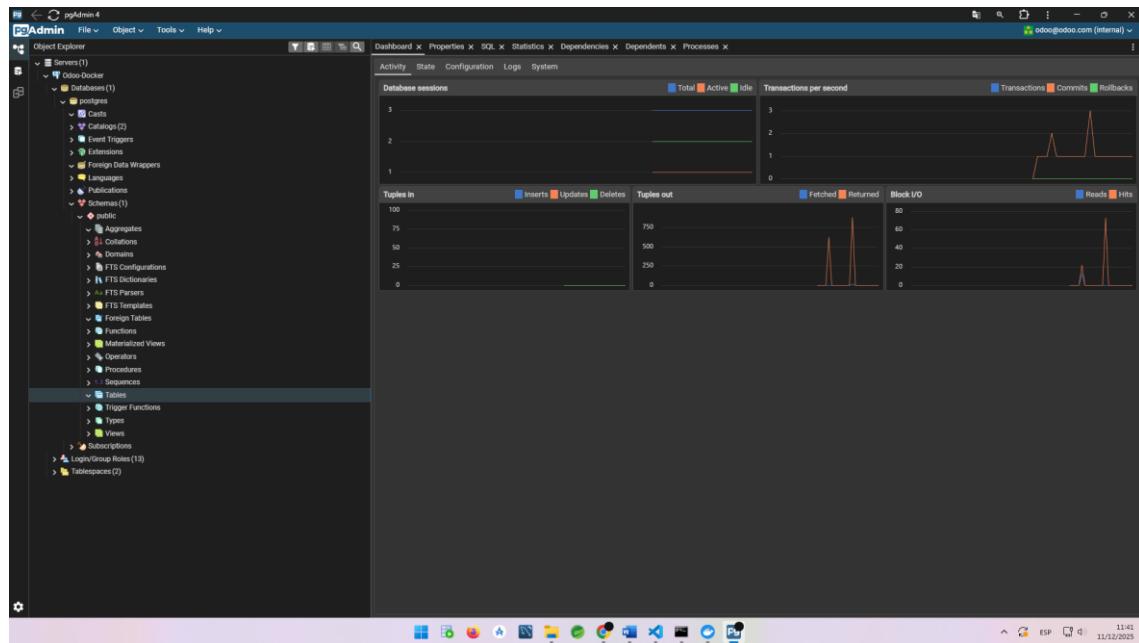
1. ¿Qué bases de datos aparecen?

Aparece la base de datos de postgres, esta es la base de datos en mantenimiento por defecto y la base de datos de Daniel_moe_T3.1 de la tarea anterior.

2. ¿Cuál corresponde a tu instalación de Odoo?

La base de Datos Daniel_moe_T3.1 y la de prueba.

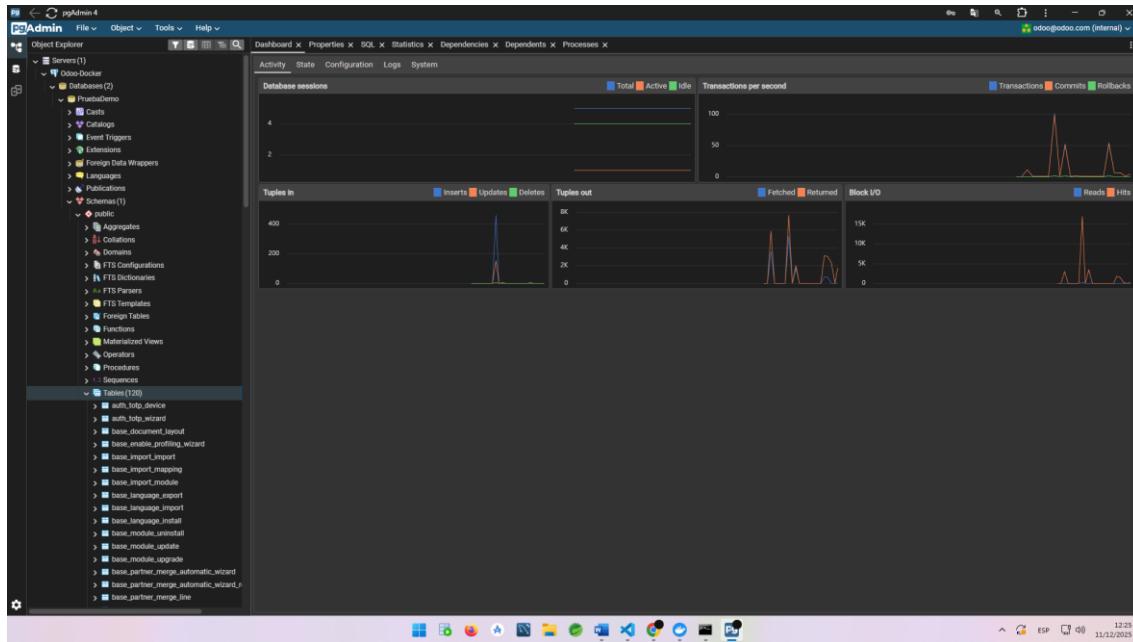
Incluye una captura.



2.2. EXPLORA LAS TABLAS

Ve a:

Schemas → public → Tables



Responde:

1. ¿Cuántas tablas aparecen en la base de datos?

Aparecen 120 tablas.

2. Elige **dos módulos de Odoo** (por ejemplo, Ventas y Contactos) y localiza **una tabla de cada uno**.

Explica por qué sabes que esas tablas pertenecen al módulo elegido (usa pistas como el prefijo: *sale_*, *res_*...).

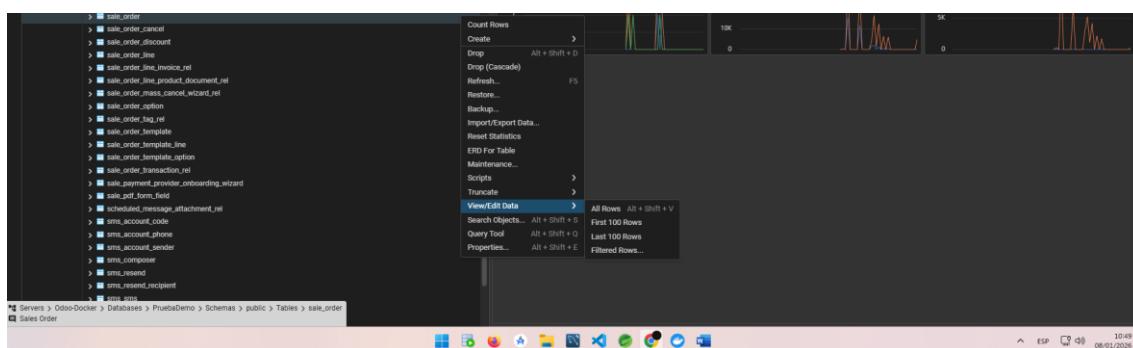
Incluye una captura de la lista de tablas.

2.3. Consulta datos desde pgAdmin

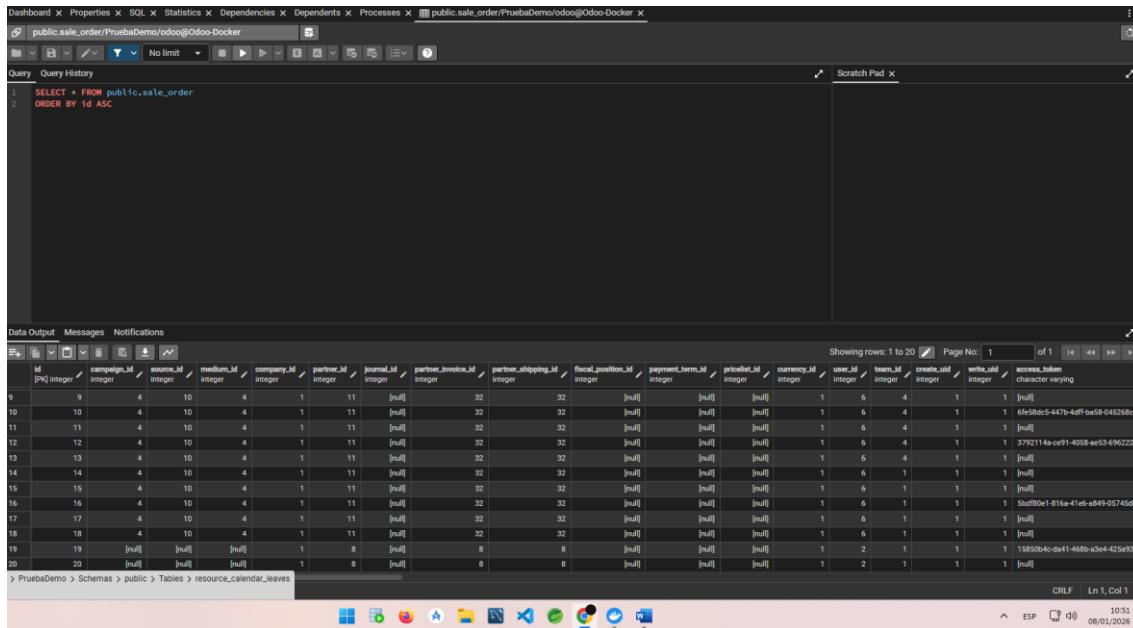
Elige **una tabla que contenga datos** (por ejemplo *res_partner* o *product_template*).

•

Abre **View/Edit Data** → *All Rows*.



Captura de registros de la tabla SALE_ORDER



The screenshot shows the Odoo SQL Client interface with the following details:

- Query:** SELECT * FROM public.sale_order ORDER BY id ASC
- Data Output:** Shows 20 rows of data from the sale_order table.
- Columns:** The columns listed are: id, campaign_id, source_id, medium_id, company_id, partner_id, journal_id, partner_invoice_id, partner_shipping_id, fiscal_position_id, payment_term_id, pricelist_id, currency_id, user_id, team_id, create_uid, write_uid, and access_token.
- Sample Data:** The first few rows show data such as:
 - Row 1: id=9, campaign_id=4, source_id=10, medium_id=4, company_id=1, partner_id=11, journal_id=null, partner_invoice_id=32, partner_shipping_id=32, fiscal_position_id=null, payment_term_id=null, pricelist_id=null, currency_id=null, user_id=1, team_id=6, create_uid=4, write_uid=1, access_token=[null]
 - Row 2: id=10, campaign_id=4, source_id=10, medium_id=4, company_id=1, partner_id=11, journal_id=null, partner_invoice_id=32, partner_shipping_id=32, fiscal_position_id=null, payment_term_id=null, pricelist_id=null, currency_id=null, user_id=1, team_id=6, create_uid=4, write_uid=1, access_token=6fe58dc5-447b-4dff-ba58-04526fc0
 - Row 3: id=11, campaign_id=4, source_id=10, medium_id=4, company_id=1, partner_id=11, journal_id=null, partner_invoice_id=32, partner_shipping_id=32, fiscal_position_id=null, payment_term_id=null, pricelist_id=null, currency_id=null, user_id=1, team_id=6, create_uid=4, write_uid=1, access_token=[null]
 - Row 4: id=12, campaign_id=4, source_id=10, medium_id=4, company_id=1, partner_id=11, journal_id=null, partner_invoice_id=32, partner_shipping_id=32, fiscal_position_id=null, payment_term_id=null, pricelist_id=null, currency_id=null, user_id=1, team_id=6, create_uid=4, write_uid=1, access_token=3792114a-ce91-4058-ae53-696222
 - Row 5: id=13, campaign_id=4, source_id=10, medium_id=4, company_id=1, partner_id=11, journal_id=null, partner_invoice_id=32, partner_shipping_id=32, fiscal_position_id=null, payment_term_id=null, pricelist_id=null, currency_id=null, user_id=1, team_id=6, create_uid=4, write_uid=1, access_token=[null]

- ¿Cuántos registros aparecen?
En total salen 20 registros.
- Describe tres columnas que te llamen la atención:
Id, campaign_id y company_id son identificadores que se usan para relacionar tablas.

1. id

- **Tipo de dato:** integer (clave primaria)
- **Qué representa:** Es el identificador único de cada registro en la tabla resource_calendar_leaves. En el contexto de Odoo, esta tabla gestiona las ausencias o días no laborables en los calendarios de recursos (como empleados o máquinas). El id permite distinguir cada ausencia registrada.

2. campaign_id

- **Tipo de dato:** integer
- **Qué representa:** Referencia a una campaña de marketing (utm.campaign). Se usa para rastrear el origen de una acción comercial o venta. Aunque esta tabla está relacionada con calendarios, Odoo permite asociar ausencias a campañas si están vinculadas a procesos como vacaciones por incentivos o planificación comercial.

3. company_id

- **Tipo de dato:** integer
- **Qué representa:** Identifica a qué empresa pertenece el recurso o la ausencia registrada. En entornos multiempresa, esta columna es clave para separar la información entre empresas dentro del mismo sistema Odoo. Apunta a la tabla res.company.
- nombre de la columna
- tipo de dato
- qué información representa dentro del ERP

Incluye una captura de la tabla con algunos registros.

3. INTERPRETACIÓN DE NOMBRES Y ESTRUCTURAS

Los nombres de las tablas en PostgreSQL provienen de las clases Python de Odoo.

Responde:

1. ¿Cómo se transforma el nombre de una clase como `sale.order` para generar la tabla correspondiente?

Se quita el punto y se pone guion bajo: `sale.order` → `sale_order`

2. Pon dos ejemplos más a partir de nombres de módulos que ya hayas explorado.

- `res.partner` → `res_partner` → tabla de contactos (clientes, proveedores, etc.)
- `sale.order.line` → `sale_order_line` → líneas de cada pedido (productos, precios, cantidades)

3. Explica por qué este sistema de nombres facilita identificar los datos del ERP.

4. USUARIOS DE LA BASE DE DATOS

Ve a:

Login/Group

Roles

Object Explorer

- > Foreign Data Wrappers
- > Languages
- > Publications
- > Schemas (1)
 - public
 - > Aggregates
 - > Collations
 - > Domains
 - > FTS Configurations
 - > FTS Dictionaries
 - > FTS Parsers
 - > FTS Templates
 - > Foreign Tables
 - > Functions
 - > Materialized Views
 - > Operators
 - > Procedures
 - > Sequences
 - > Tables (394)
 - > Trigger Functions
 - > Types
 - > Views
 - > postres
- > Login/Group Roles (13)
 - odoor
 - pg_checkpoint
 - pg_database_owner
 - pg_execute_server_program
 - pg_monitor
 - pg_read_all_data
 - pg_read_all_settings
 - pg_read_all_stats
 - pg_read_server_files
 - pg_signal_backend
 - pg_stat_scan_tables
 - pg_write_all_data
 - pg_write_server_files

File ▾ Object ▾ Tools ▾ Help ▾

Object Explorer

Extensions

Foreign Data Wrappers

Languages

Publications

Schemas (1)

public

Aggregates

Collations

Domains

FTS Configurations

FTS Dictionaries

FTS Parsers

FTS Templates

Foreign Tables

Functions

Materialized Views

Operators

Procedures

Sequences

Tables (394)

Trigger Functions

Types

Views

Subscriptions

postres

Login/Group Roles (13)

odoor

pg_checkpoint

pg_database_owner

pg_execute_server_program

pg_monitor

pg_read_all_data

pg_read_all_settings

pg_read_all_stats

pg_read_server_files

pg_signal_backend

pg_stat_scan_tables

pg_write_all_data

pg_write_server_files

Tables

Tablespaces

Responde:

1. ¿Qué usuarios existen en el servidor PostgreSQL?

Solo existe uno llamado odoo, lo demás son roles.

2. ¿Cuál es el usuario que utiliza Odoo para conectarse?

3. Abre las propiedades del usuario **odoo**:

The screenshot shows the pgAdmin interface for managing PostgreSQL users. At the top, there's a tree view of roles under the 'odoo' schema. A context menu is open over the 'odoo' role, with the 'Properties...' option highlighted. Below this, a detailed properties dialog for the 'odoo' role is displayed. The 'Privileges' tab is selected, showing several checkboxes for various administrative permissions, all of which are checked (indicated by blue toggles). At the bottom of the dialog are buttons for 'Close', 'Reset', and 'Save'.

- ¿Puede crear bases de datos? ¿Tiene permisos de login?

Como podemos ver en la pantalla, este usuario puede crear las bases de datos y también permite hacer login.

4. Explica por qué un ERP necesita un usuario propio para la base de datos.

Porque así el ERP trabaja con su propio usuario y no toca nada que no debe. Ese usuario solo tiene los permisos justos para crear tablas, leer datos y guardar cambios. De esta forma:

- **se evita romper la base por accidente**
- **se controla mejor la seguridad**
- **y todo lo que hace el ERP queda separado del resto de cosas del servidor.**

Incluye una captura donde se vea la lista de roles.

5. REFLEXIÓN FINAL

Responde con tus palabras:

1. ¿Qué parte te ha servido más para entender cómo funciona un ERP por dentro?

Ver las tablas. Ahí es donde ves realmente qué guarda el sistema y cómo todo lo que haces acaba ahí.

2. ¿Te ha sorprendido la cantidad de tablas que utiliza Odoo? ¿Por qué?

Sí, porque desde fuera Odoo parece más sencillo, pero por dentro tiene un montón de cosas y cada módulo mete las suyas.

3. ¿Qué relación ves entre lo que ocurre en la interfaz de Odoo y los datos que has visto en las tablas?

Muy directa, ya que lo que haces en la pantalla se convierte en datos en esas tablas. La interfaz es la parte bonita, lo importante está guardado ahí.

WEBGRAFÍA

<https://experts-deny-b9a.craft.me/xOJQn4khXm30Ob>

<https://openwebinars.net/blog/pgadmin-para-novatos-como-gestionar-bases-de-datos-sin-complicaciones/>