



TAREA 4.1 - Exploración de la base de datos de Odoo con pgAdmin



Nombre y apellidos: Cristina Sandoval Laborde

Curso: 2ºDAM

Asignatura: Sistemas de gestión

Índice

1. Puesta en marcha y conexión con la BD	1
1.1. Comprueba que los contenedores están activos	1
1.2. Abre pgAdmin y crea una conexión al servidor PostgreSQL de Odoo	1
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.....	5
2. Elige dos módulos de Odoo (por ejemplo Ventas y Contactos) y localiza una tabla de cada uno.	6
2.3. Consulta datos desde pgAdmin	9
3. Interpretación de nombres y estructuras.....	11
4. Usuarios de la base de datos	12
5. Reflexión final	15
Webgrafía.....	15

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.

```
Windows PowerShell
Copyright (C) Microsoft Corporation. Todos los derechos reservados.

Instale la versión más reciente de PowerShell para obtener nuevas características y mejoras. https://aka.ms/PSWindows

PS C:\Users\alumno> cd "C:\Users\alumno\Desktop\docker_test"
PS C:\Users\alumno\Desktop\docker_test> docker ps
error during connect: Get "http://%2F%2Fpipe%2FdockerDesktopLinuxEngine/v1.51/containers/json": open //./pipe/docker
DesktopLinuxEngine: The system cannot find the file specified.
PS C:\Users\alumno\Desktop\docker_test> docker ps
CONTAINER ID IMAGE COMMAND CREATED STATUS PORTS
NAMES
9f4452f1f590 odoo:18 "/entrypoint.sh odoo" 44 hours ago Up 13 seconds 0.0.0.0:8069->806
9/tcp, [::]:8069->8069/tcp
f0969edec3ad dpage/pgadmin4:latest "/entrypoint.sh" 44 hours ago Up 13 seconds 0.0.0.0:5050->80/
tcp, [::]:5050->80/tcp
33607a86187d postgres:15 "docker-entrypoint.s..." 44 hours ago Up 13 seconds 0.0.0.0:5432->543
2/tcp, [::]:5432->5432/tcp
b33c9f71e65e portainer/portainer-ce:latest "/portainer" 4 weeks ago Up 13 seconds 0.0.0.0:9000->900
0/tcp, [::]:9000->9000/tcp, 0.0.0.0:9443->9443/tcp, [::]:9443->9443/tcp portainer
PS C:\Users\alumno\Desktop\docker_test> |
```

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.

Crear - Servidor

x

General Conexión SSL Túnel SSH Avanzado

Nombre Odoo-Docker

Grupo de servidores Servidores

Fondo X

Primer plano X

¿Conectarse ahora?

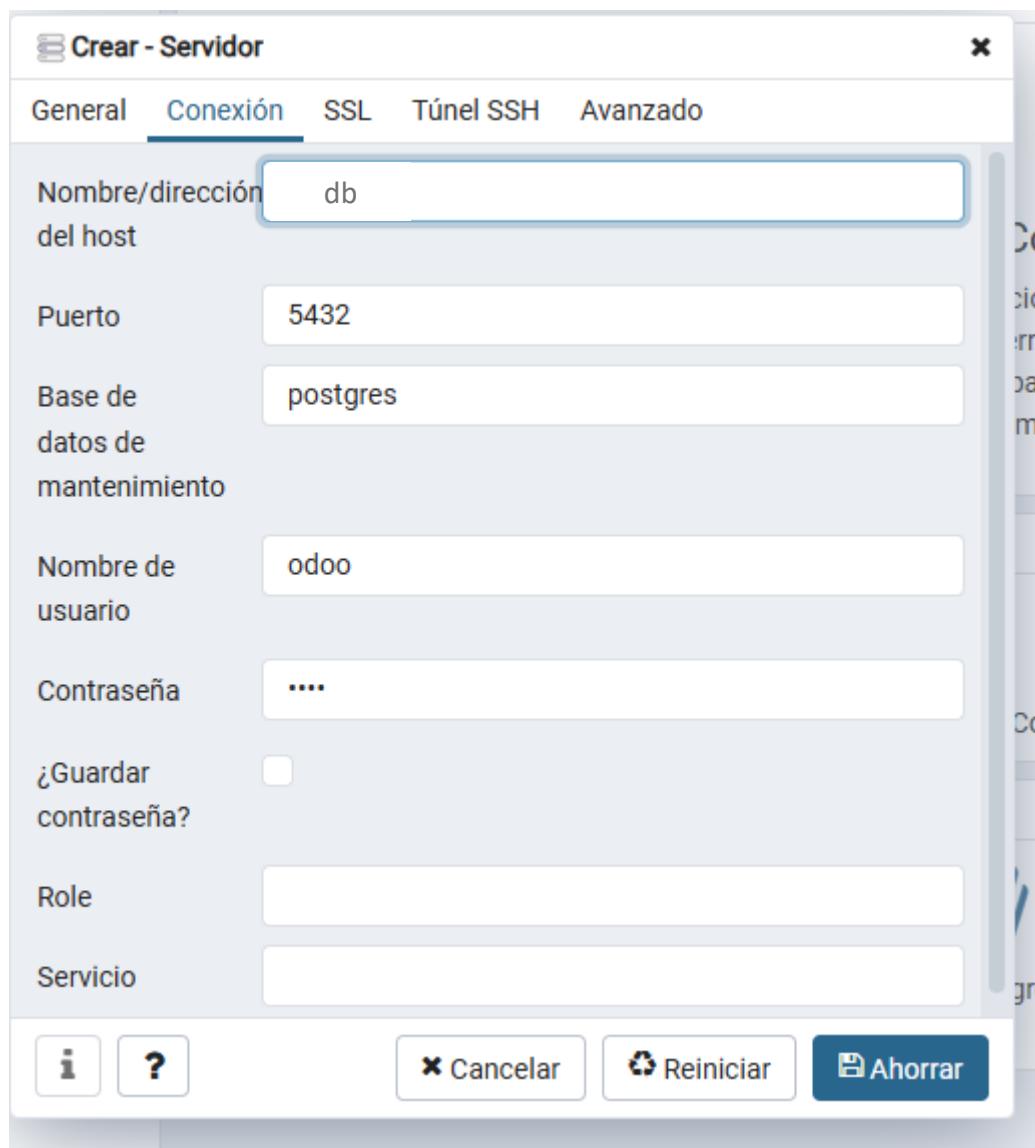
Comentarios



X Cancelar

⟳ Reiniciar

💾 Ahorrar



Tuve que cambiar el docker-compose.yml para añadirle el puerto y algunas modificaciones que nos dijo Eva para que funcionara, sino no se podía crear el servidor.

En local de poner localhost como decían las instrucciones tuve que poner db, como se ve observa en la imagen de arriba.

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

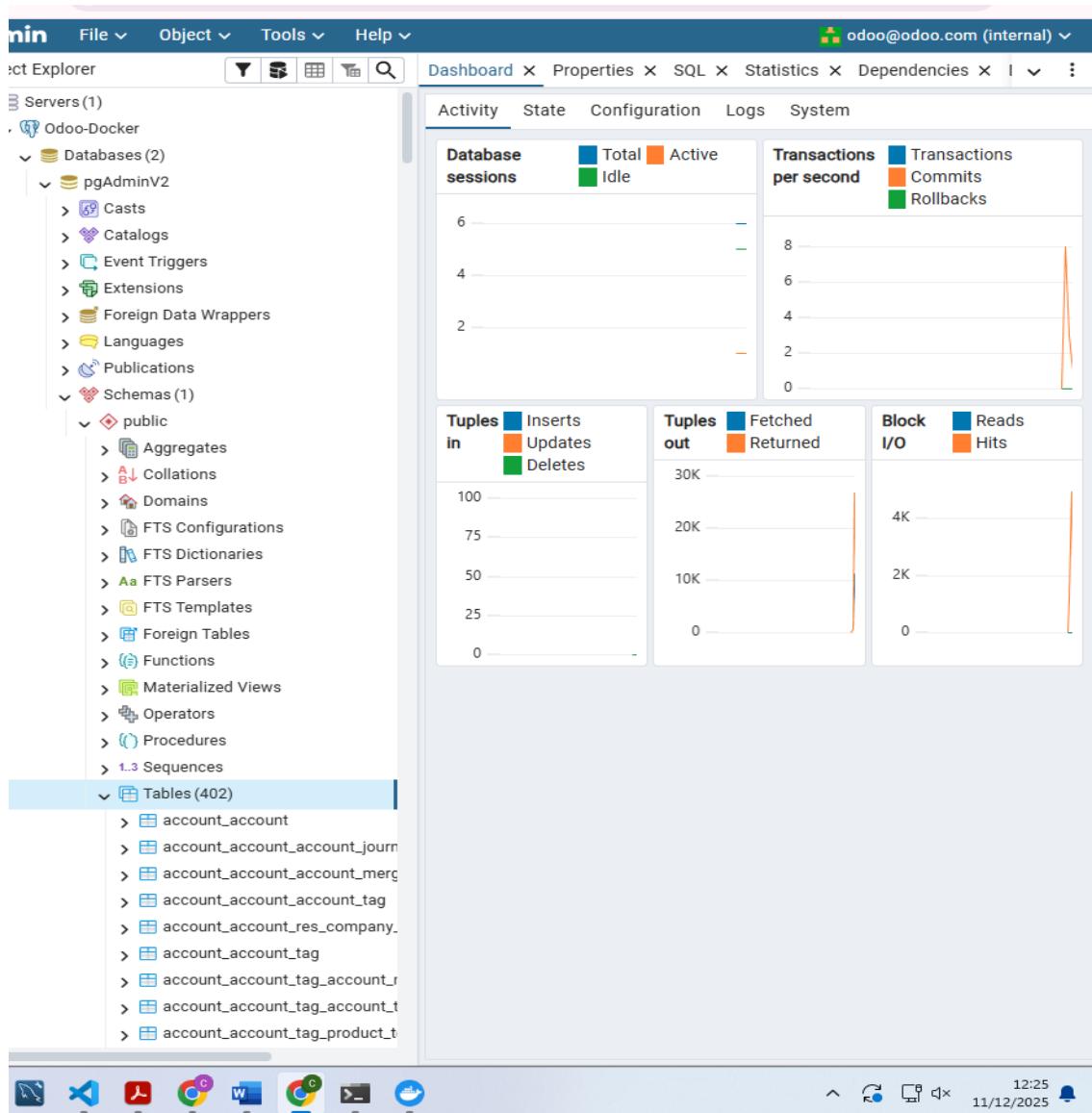
1. ¿Qué bases de datos aparecen?

La de postgres, que es la de mantenimiento por defecto.

Y la mia, pgAdminV2

2. ¿Cuál corresponde a tu instalación de Odoo?

pgAdminV2 que es mi base de datos



2.2. Explora las tablas

Ve a:

Schemas → public → Tables

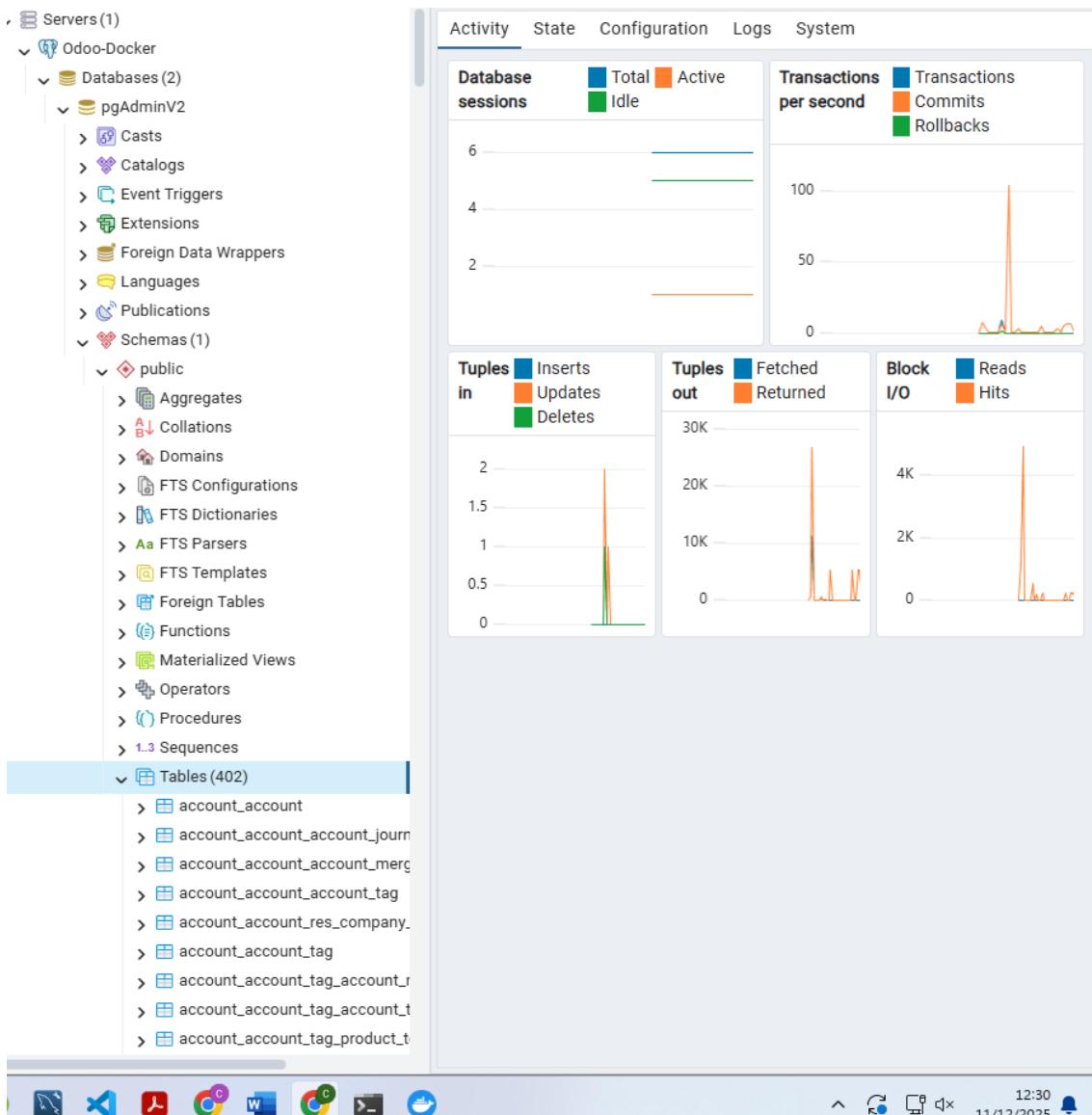
Aquí en mi proyecto de Moe no me salían tablas, además me dio muchos problemas porque Odoo decía que estaba corrupto y me tiraba el servidor localhost:8069, para que funcione tuve que borrar esta base de datos del PgAdmin y en la powershell tirar los servidores y levantarlos 2 veces, porque con una no me funcionaba, utilice los siguientes comandos

```
PS C:\Users\alumno\Desktop\docker_test> docker compose down -v
time="2025-12-11T12:21:37+01:00" level=warning msg="C:\\\\Users\\\\alumno\\\\Desktop\\\\docker_test\\\\
docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it
to avoid potential confusion"
[+] Running 7/7
✓ Container pgadmin_container    Removed          1.3s
✓ Container odoo_app             Removed          0.7s
✓ Container odoo_db              Removed          0.3s
✓ Volume docker_test_odoo_db_data Removed          0.0s
✓ Volume docker_test_pgadmin_data Removed          0.0s
✓ Volume docker_test_odoo_addons Removed          0.0s
✓ Network docker_test_default   Removed          0.3s
PS C:\Users\alumno\Desktop\docker_test> docker compose up -d
time="2025-12-11T12:21:43+01:00" level=warning msg="C:\\\\Users\\\\alumno\\\\Desktop\\\\docker_test\\\\
docker-compose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it
to avoid potential confusion"
[+] Running 2/3
✓ Network docker_test_default   Created          0.0s
✓ Volume docker_test_odoo_addons Created          0.0s
[+] Running 7/7
r_test_pgadmin_data Creating      0.0s
✓ Network docker_test_default   Created          0.0s
✓ Volume docker_test_odoo_addons Created          0.0s
✓ Volume docker_test_pgadmin_data Created          0.0s
✓ Volume docker_test_odoo_db_data Created          0.0s
✓ Container odoo_db             Started          0.5s
✓ Container pgadmin_container   Started          0.6s
✓ Container odoo_app            Started          0.7s
PS C:\Users\alumno\Desktop\docker_test> |
```

Responde:

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

402



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

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

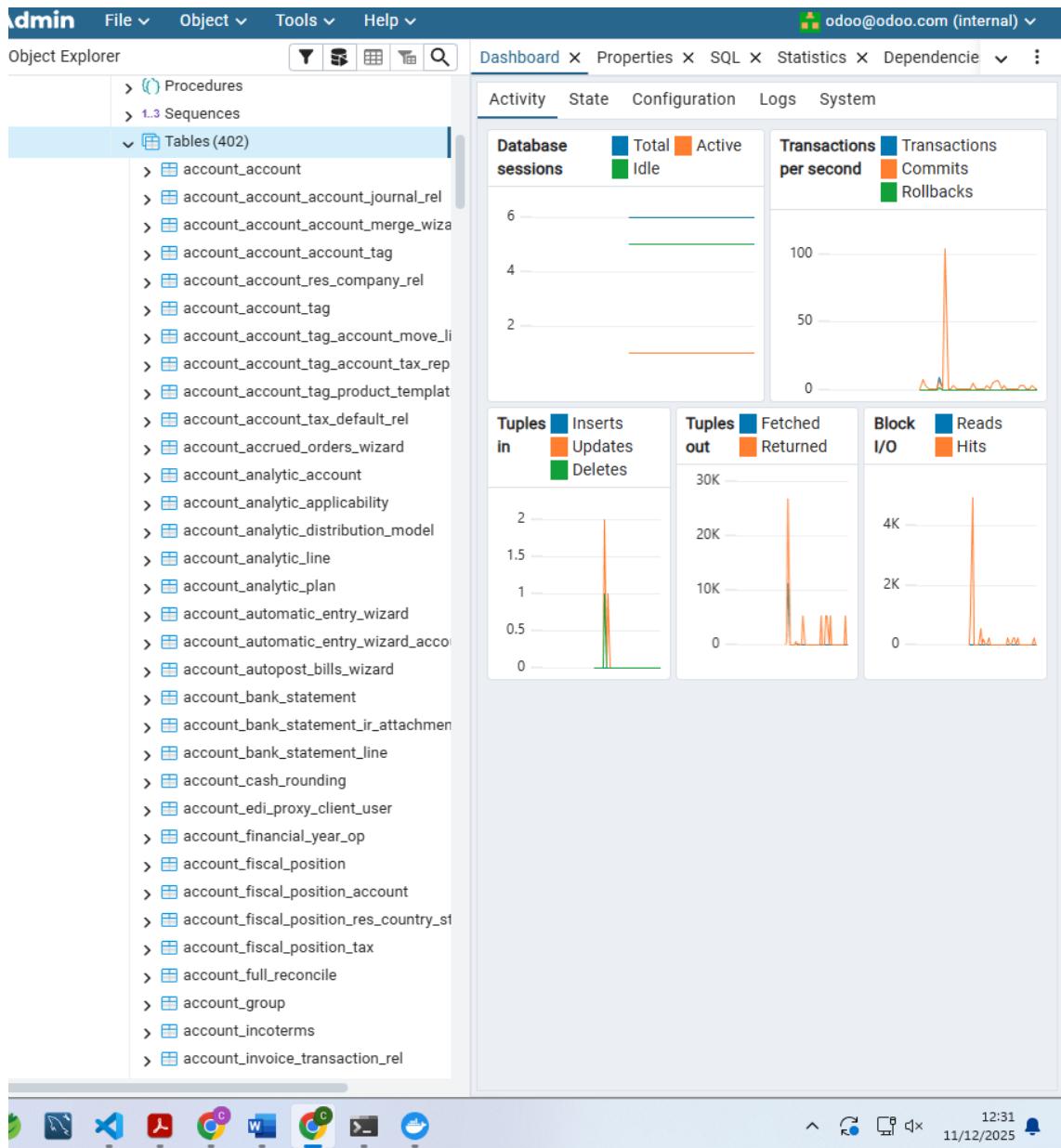
Para `sale_order`: Sé que pertenece a **Ventas** porque empieza por el prefijo `sale_`.

Para `account_account`: Sé que es de **Facturación/Contabilidad** porque lleva el prefijo `account_`.

Ventas



Contabilidad



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.

He seleccionado la table *res_partner*

The screenshot shows the pgAdmin 4 interface. In the Object Explorer on the left, under the 'res_partner' table, the 'Data' node is selected. The main pane displays the data from the 'res_partner' table:

	id [PK] integer	company_id integer	create_date timestamp without time zone	name character varying
1	1	[null]	2025-12-11 11:23:08.030115	My Company
2	2	1	2025-12-11 11:23:08.936391	OdooBot
3	3	1	2025-12-11 11:23:08.936391	Administrator
4	4	[null]	2025-12-11 11:23:08.936391	Public user
5	5	[null]	2025-12-11 11:23:08.936391	Default User Template
6	6	[null]	2025-12-11 11:23:08.936391	Portal User Template
7	7	[null]	2025-12-11 11:24:13.806749	Simplified Invoice Partner (...)

Total rows: 7 Query complete 00:00:00.105 CRLF Ln 1, Col 1

- ¿Cuántos registros aparecen?

Aparecen 7 registros → “Total rows 7”

- Describe tres columnas que te llamen la atención:

- nombre de la columna

- tipo de dato

- qué información representa dentro del ERP

1. Columna de Identificación

- **Nombre de la columna:** id
- **Tipo de dato:** integer (Entero)
- **Qué información representa:** Es la **Clave Primaria (Primary Key)**. En el ERP, este número es único de ese registro. Odoo lo utiliza internamente para relacionar este contacto con otros modelos (por ejemplo, para asignar este id a una factura o un pedido de venta).

2. Columna de Auditoría

- **Nombre de la columna:** create_date
- **Tipo de dato:** timestamp without time zone
- **Qué información representa:** Es un "campo mágico" de Odoo. El ERP rellena esta columna automáticamente en el momento exacto en que se crea el registro para llevar un registro de auditoría (saber cuándo se dio de alta al cliente o usuario).

3. Columna de Información Principal

- **Nombre de la columna:** name
- **Tipo de dato:** character varying
- **Qué información representa:** Es el dato más importante para el usuario. Almacena el **nombre del contacto** y es el campo que se muestra por defecto cuando buscas a alguien en la barra de búsqueda de Odoo

The screenshot shows the PgAdmin 4 interface connected to a PostgreSQL database. The left sidebar shows the Object Explorer with various schema and table nodes. The main area has a 'Query' tab open with the following SQL code:

```
1 SELECT * FROM public.res_partner
2 ORDER BY id ASC
```

The results pane displays the data from the res_partner table:

id	company_id	create_date	name	title	parent_id	user_id	state_id	country_id	industry_id	color	commercial_partner_id	create_uid	write_uid	con
1	[null]	2025-12-11 23:08:020115	My Company	[null]	[null]	[null]	68	[null]	0	0	1	[null]	1	M
2	1	2025-12-11 23:08:93691	Odoo	[null]	[null]	[null]	[null]	[null]	0	0	2	1	1	Ot
3	3	2025-12-11 23:08:93691	Administrator	[null]	[null]	[null]	[null]	[null]	0	0	3	1	1	Ad
4	4	2025-12-11 23:08:93691	Public user	[null]	[null]	[null]	[null]	[null]	0	0	4	1	1	Pu
5	5	2025-12-11 23:08:93691	Default User Template	[null]	[null]	[null]	[null]	[null]	0	0	5	1	1	Dt
6	6	2025-12-11 23:08:93691	Portal User Template	[null]	[null]	[null]	[null]	[null]	0	0	6	1	1	Pt
7	7	2025-12-11 24:13:805749	Simplified Invoice Partner ...	[null]	[null]	[null]	[null]	[null]	0	0	7	1	1	Sip

Total rows: 7 Query complete 00:00:00.105

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 sustituyen los puntos(.) por guiones bajos(_)

Clase Python (Modelo): sale.order

Tabla PostgreSQL: sale_order

Odoo toma el nombre técnico del modelo definido en el código y aplica este cambio automáticamente para crear la tabla en la base de datos.

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

Ejemplo 1 (Módulo de Actividades/Correo):

- Clase Python (Modelo): **mail.activity**
- Tabla PostgreSQL: **mail_activity**
- (Esta tabla gestiona las actividades planificadas como "Llamada", "Reunión" o "Por hacer" que ves en el buzón de Odoo).

Ejemplo 2 (Módulo de SMS):

- Clase Python (Modelo): **sms.account.code**
- Tabla PostgreSQL: **sms_account_code**
- (Siguiendo la lógica: el punto entre sms, account y code se sustituye por un guion bajo _ en la base de datos).

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

Agrupación por módulos (Prefijos): Al usar la primera parte del nombre como prefijo (ej: account_, sale_, res_), todas las tablas de un mismo módulo aparecen juntas ordenadas alfabéticamente en la base de datos.

Predictibilidad: Si conoces el nombre funcional del objeto en Odoo (ej: una "Orden de Venta"), puedes deducir casi siempre cómo se llamará la tabla (sale_order) sin necesidad de consultar documentación técnica.

4. Usuarios de la base de datos

Ve a:

Login/Group Roles

Responde:

1.¿Qué usuarios existen en el servidor PostgreSQL?:

odoo: El usuario creado específicamente para la aplicación.

Roles de sistema (pg_): Aparecen varios roles predefinidos de PostgreSQL para mantenimiento y monitoreo, como pg_monitor, pg_read_all_data, pg_signal_backend, etc.

The screenshot shows the pgAdmin 4 interface. The left pane is the Object Explorer, displaying the database structure. The 'Login/Group Roles' node under 'public' is expanded, showing roles like 'odoo', 'pg_checkpoint', 'pg_database_owner', etc. The right pane shows a query results grid for a SELECT statement. The top bar shows the connection details: 'odoo@odoo.com (internal)'. The bottom status bar shows 'CRLF' and 'Ln 2, Col 17'.

	<code>id [PK] integer</code>	<code>company_id integer</code>	<code>create_date timestamp without time zone</code>
1	1	[null]	2025-12-11 11:23:08.030115
2	2	1	2025-12-11 11:23:08.936391
3	3	1	2025-12-11 11:23:08.936391
4	4	[null]	2025-12-11 11:23:08.936391
5	5	[null]	2025-12-11 11:23:08.936391
6	6	[null]	2025-12-11 11:23:08.936391
7	7	[null]	2025-12-11 11:24:13.806749

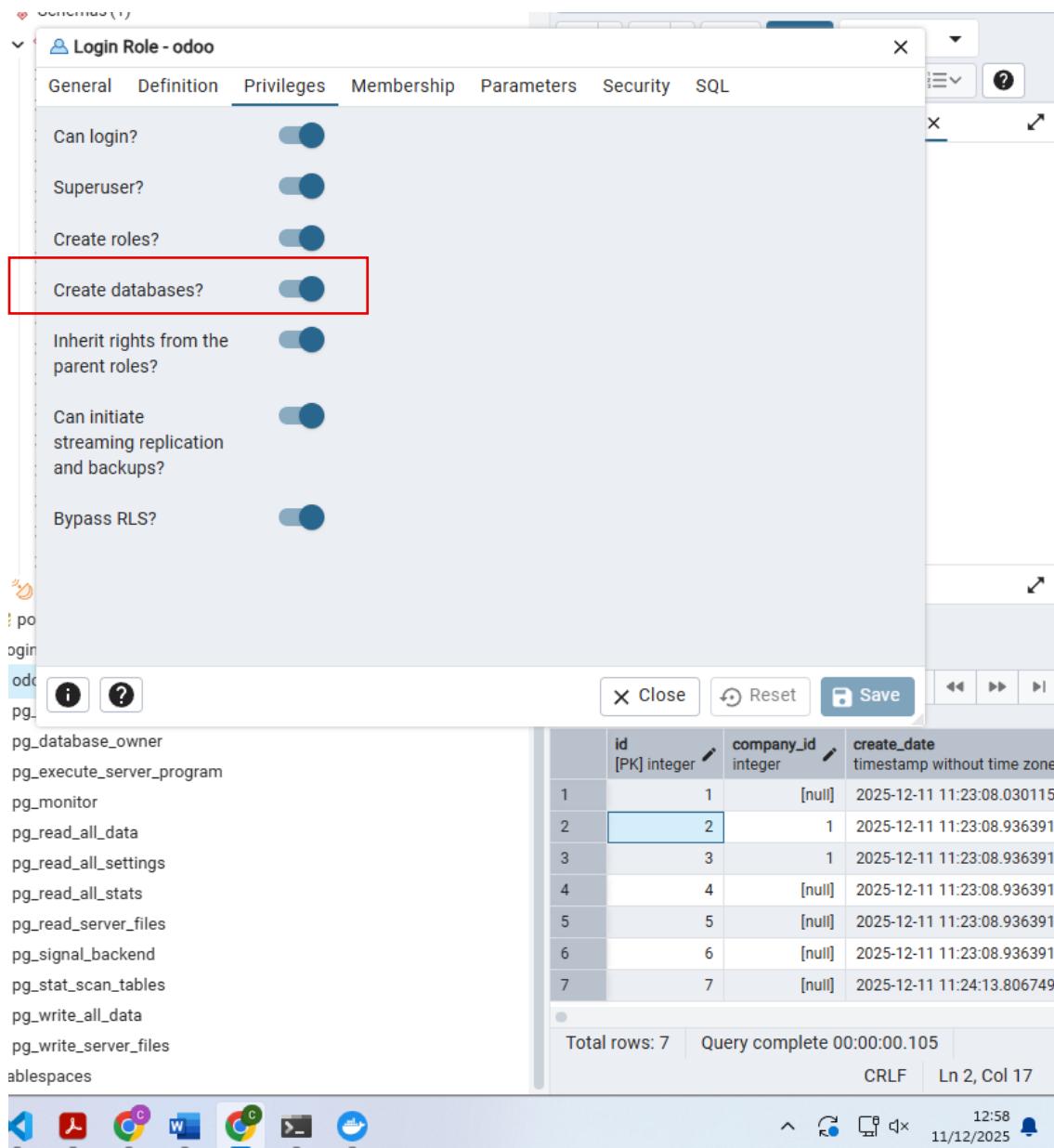
2. ¿Cuál es el usuario que utiliza Odoo para conectarse?

El usuario es **odoo**.

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

- ¿Puede crear bases de datos?

Sí. En la imagen se ve el interruptor de "Create databases?" activado (en azul). Esto es fundamental para Odoo, ya que el propio sistema puede crear nuevas bases de datos desde su interfaz web (por ejemplo, al crear una nueva empresa)



The screenshot shows the Odoo PostgreSQL role configuration interface. The top navigation bar includes tabs for General, Definition, Privileges, Membership, Parameters, Security, and SQL. The 'Privileges' tab is selected. A red box highlights the 'Create databases?' checkbox, which is checked (blue). Other checkboxes shown are 'Can login?', 'Superuser?', 'Create roles?', 'Inherit rights from the parent roles?', 'Can initiate streaming replication and backups?', and 'Bypass RLS?'. Below the checkboxes is a table listing PostgreSQL roles:

	id [PK] integer	company_id integer	create_date timestamp without time zone
1	1	[null]	2025-12-11 11:23:08.030115
2	2	1	2025-12-11 11:23:08.936391
3	3	1	2025-12-11 11:23:08.936391
4	4	[null]	2025-12-11 11:23:08.936391
5	5	[null]	2025-12-11 11:23:08.936391
6	6	[null]	2025-12-11 11:23:08.936391
7	7	[null]	2025-12-11 11:24:13.806749

Total rows: 7 Query complete 00:00:00.105 CRLF Ln 2, Col 17

- ¿Tiene permisos de login?

Sí. En la imagen se ve el interruptor de "Can login?" activado. Sin esto, Odoo no podría conectarse a PostgreSQL.

The screenshot shows the pgAdmin interface with the following details:

- Login Role - odoo** configuration window:
 - Privileges tab selected.
 - "Can login?" toggle switch is turned on (highlighted with a red box).
 - Other privileges listed: Superuser?, Create roles?, Create databases?, Inherit rights from the parent roles?, Can initiate streaming replication and backups?, Bypass RLS? (all are turned off).
- Database query results table:

	id [PK] integer	company_id integer	create_date timestamp without time zone
1	1	[null]	2025-12-11 11:23:08.030115
2	2	1	2025-12-11 11:23:08.936391
3	3	1	2025-12-11 11:23:08.936391
4	4	[null]	2025-12-11 11:23:08.936391
5	5	[null]	2025-12-11 11:23:08.936391
6	6	[null]	2025-12-11 11:23:08.936391
7	7	[null]	2025-12-11 11:24:13.806749

Total rows: 7 | Query complete 00:00:00.105 | CRLF | Ln 2, Col 17
- Bottom status bar: 12:59, 11/12/2025.

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

Seguridad (Principio de menor privilegio): No es seguro que una aplicación utilice el usuario root (postgres) del sistema. Si alguien lograra hackear Odoo desde la web, y siempre se usará el usuario root por defecto para las bases de datos esa persona podría tener acceso a todas las bases de datos, su información, modificar datos.... con un usuario propio limitamos el daño solo a lo que ese usuario tenga permitido tocar, protegiendo el resto del servidor.

Automatización: Odoo es un software, un "robot". Necesita un usuario y contraseña fijos escritos en su archivo de configuración (odoo.conf) para poder conectarse a la base de datos automáticamente cada vez que se inicia el servicio, sin que una persona tenga que escribir la clave.

The screenshot shows the pgAdmin interface. On the left, the 'Login/Group Roles' tree is expanded, showing a node for 'odoo' which contains various PostgreSQL system roles like 'pg_checkpoint', 'pg_database_owner', etc. Below this is a 'Tablespaces' node. On the right, a table named 'company_id' is displayed with 7 rows. The columns are 'id [PK] integer', 'company_id integer', and 'create_date timestamp without time zone'. The data shows entries from 1 to 7, all associated with company_id 1. The bottom status bar shows 'Query complete 00:00:00.105' and the date/time '11/12/2025 13:05'.

	<code>id [PK] integer</code>	<code>company_id integer</code>	<code>create_date timestamp without time zone</code>
1	1	[null]	2025-12-11 11:23:08.030115
2	2	1	2025-12-11 11:23:08.936391
3	3	1	2025-12-11 11:23:08.936391
4	4	[null]	2025-12-11 11:23:08.936391
5	5	[null]	2025-12-11 11:23:08.936391
6	6	[null]	2025-12-11 11:23:08.936391
7	7	[null]	2025-12-11 11:24:13.806749

5. Reflexión final

Responde con tus palabras:

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

Ver la estructura de árbol en pgAdmin. Me ha ayudado a visualizar que un ERP no son módulos o elementos sueltos, sino un conjunto de roles, permisos y tablas ordenadas donde se almacena todo.

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

La verdad es que sí. Me ha sorprendido ver que está todo super estructurado y organizado. Veo muy útil que las tablas sigan un orden alfabético y utilicen prefijos para agruparlas por categorías o módulos. Esto hace que sea mucho más fácil y rápido realizar búsquedas o mantenimientos en la base de datos.

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

Existe una relación directa. La interfaz de Odoo actúa haciendo la parte visual. Cuando yo relleno datos de un cliente o creo una factura en la web de Odoo y pulso 'Guardar', lo que realmente está ocurriendo por detrás es que el sistema inserta una nueva fila con esos datos en la tabla correspondiente de PostgreSQL. La interfaz solo nos facilita la tarea para no tener que escribir comandos SQL manualmente.

Webgrafía

<http://localhost:5050>

<http://localhost:8969>

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