

TEMA 4 - ORGANIZACIÓN, CONSULTA Y MODIFICACIÓN DE LA INFORMACIÓN

Índice de contenidos

1. Acceso a la base de datos con PGAdmin
2. Consultar la información
3. Creación y modificación de datos

1. Acceso a la base de datos con PGAdmin

La base de datos de un sistema ERP es muy grande. Los datos se almacenan en las tablas de la aplicación, las vistas de las tablas y cualquier otro elemento del cual necesitemos mostrar los datos o guardarlos.

Cualquier administrador del ERP puede acceder a la base de datos de Odoo directamente. Para ello solo necesitaremos conectar con ella a través de un cliente de la base de datos. Principalmente tenemos dos alternativas:

- **psql**: si sabemos manejar la línea de comandos, con este cliente podemos acceder y manipular cualquier base de datos Postgres.
- **PgAdmin** (en cualquiera de sus versiones): es un cliente gráfico de Postgres. Se trata de un software multiplataforma con el que podremos ver gráficamente las bases de datos de nuestro Postgres.

1.1. Instalación y configuración de pgAdmin

pgAdmin es un software gratuito distribuido bajo licencia PostgreSQL. Podemos encontrarlo en los repositorios oficiales, concretamente en la **página de descargas de pgAdmin**.

En nuestro caso instalaremos **la versión correspondiente a nuestro sistema operativo (Windows, macOS o Linux)**.

La opción "Container" que aparece en la web está pensada para instalaciones avanzadas donde también se desea ejecutar pgAdmin dentro de Docker, algo que **no necesitamos en este módulo**.

Una vez descargado el instalador, lo ejecutamos y abrimos pgAdmin.

¿Para qué sirve pgAdmin?

pgAdmin es una herramienta gráfica que nos permite **administrar y trabajar con bases de datos PostgreSQL** de manera visual.

Gracias a pgAdmin podemos:

- Ver todas las bases de datos creadas en un servidor PostgreSQL.
- Consultar tablas, columnas y tipos de datos.
- Ejecutar consultas SQL sin tener que usar la terminal.
- Crear nuevas bases de datos, tablas y usuarios.
- Importar y exportar datos.
- Realizar copias de seguridad (backups) y restauraciones.
- Analizar la estructura interna de las bases de datos, algo muy útil para comprender cómo Odoo organiza la información.

Esta herramienta facilita mucho el aprendizaje, ya que permite al alumnado observar directamente la relación entre lo que ocurre en Odoo y los datos almacenados en PostgreSQL.

Conexión de pgAdmin con PostgreSQL dentro del contenedor de Odoo

Odoo no incluye su propia base de datos interna. Siempre utiliza un contenedor adicional que ejecuta PostgreSQL.

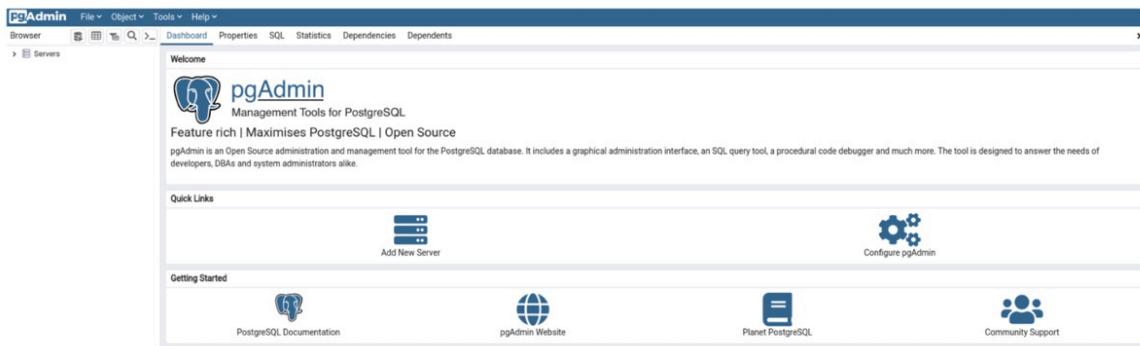
En el archivo `docker-compose.yml` suele aparecer algo parecido a:

```
1 db:  
2   image: postgres:15  
3   environment:  
4     POSTGRES_DB: postgres  
5     POSTGRES_USER: odoo  
6     POSTGRES_PASSWORD: odoo  
7   ports:  
8     - "5432:5432"
```

Ese mapeo de puertos permite que pgAdmin, instalado en nuestro equipo, pueda conectarse sin problemas.

Añadir un nuevo servidor en pgAdmin

Cuando abrimos pgAdmin veremos la ventana principal con varias opciones. Desde el menú lateral añadimos un **nuevo servidor** y configuraremos los datos necesarios.



En la pestaña **Connection** rellenamos:

- **Nombre:** nombre identificativo para la conexión, por ejemplo *Odoo-Docker*.
- **Host name/address:** `localhost`
(pgAdmin se conecta al puerto publicado por Docker).
- **Port:** `5432`.
- **Maintenance database:** `postgres`.
- **Username:** `odoo`.
- **Password:** `odoo`.

General Tab (Left)	Connection Tab (Right)
Name: Odoo	Host name/address: IpServidor
Server group: Servers	Port: 5432
Background: X	Maintenance database: postgres
Foreground: X	Username: odoo
Connect now?: checked	

Guardamos la conexión para reutilizarla siempre que queramos acceder a la base de datos.

Notas importantes cuando trabajamos con Docker

- **El contenedor debe estar en ejecución**

Podemos comprobarlo con:

```
1 docker ps
```

- **Debemos tener publicado el puerto 5432**

Si no aparece "5432:5432" en el `docker-compose.yml`, pgAdmin no podrá conectarse desde fuera del contenedor.

La solución es añadirlo, guardar y reiniciar:

```
1 docker compose down  
2 docker compose up -d
```

- **La conexión siempre se hace desde el equipo anfitrión**

En contextos más avanzados se podría usar la IP del contenedor, pero en este módulo utilizaremos siempre `localhost`.

¿Sabías que...?

Cada vez que creamos una base de datos nueva en Odoo (por ejemplo, una empresa distinta), aparece automáticamente en la lista de pgAdmin.

Esto permite revisar estructuras, consultar tablas y realizar copias de seguridad.

1.2. Los datos de Odoo

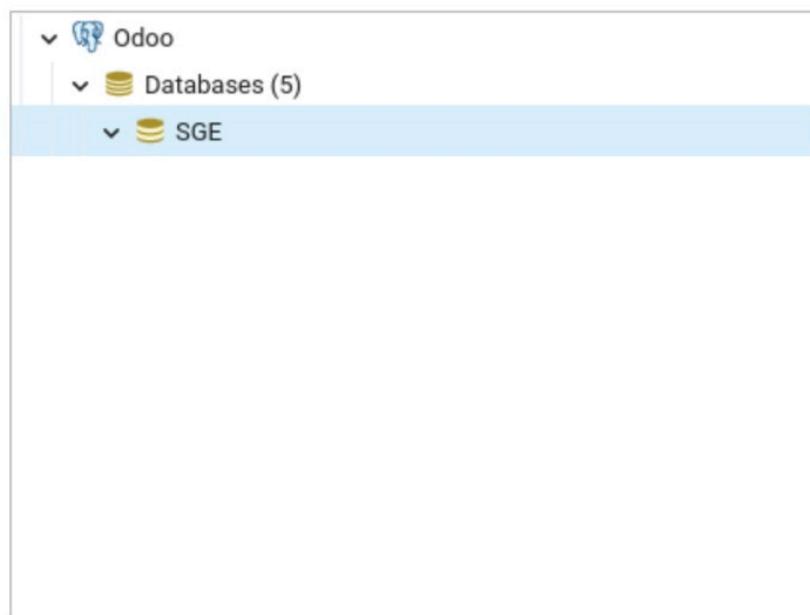
Para poder trabajar con Odoo es importante entender **qué datos guarda, dónde los guarda y cómo podemos consultarlos**.

Toda la información del ERP –clientes, productos, facturas, usuarios, etc.– se almacena en **PostgreSQL**, el sistema gestor de bases de datos que utiliza Odoo.

Desde pgAdmin podremos explorar esa información, consultarla, ver cómo se organiza y, si fuese necesario, modificarla.

1.2.1. Las tablas de la base de datos

Al abrir pgAdmin y conectarnos al servidor que hemos configurado, veremos las bases de datos que tenemos disponibles. Cada vez que creamos una base de datos en Odoo, esta aparece también en PostgreSQL.



Si desplegamos la base de datos y entramos en **Schemas** → **public** → **Tables**, veremos **todas las tablas** que utiliza Odoo.

Un ERP no es una aplicación pequeña: un simple módulo puede crear tres o cuatro tablas nuevas, y al instalar varios módulos el número de tablas puede llegar a ser enorme.

Desde pgAdmin podemos:

- ver las bases de datos existentes,
- crear nuevas,
- eliminar bases de datos,
- crear tablas,
- borrar tablas,
- consultar la información almacenada en ellas.

1.2.2. Identificar las tablas

Odoo organiza sus datos utilizando clases Python. Cada clase que existe en el código del ERP se transforma en una tabla dentro de PostgreSQL. A su vez, cada atributo de esa clase se convierte en una columna de la tabla.

Este detalle nos permite localizar con rapidez en qué tabla se guarda cierta información. Para ello debemos recordar dos ideas:

1. Cómo se nombran las clases Python de Odoo

- Siempre en minúscula.
- Las palabras compuestas usan guion bajo.
- La estructura es:

nombre_del_modulo.nombre_de_la_clase

- En la base de datos, los puntos se convierten en guiones bajos.

Por ejemplo:

1	website.menu	→	website_menu
2	sale.order	→	sale_order
3	product.template	→	product_template

2. Cómo se nombran los atributos de las clases

- También van siempre en minúscula.
- Siguen el mismo estilo: palabras unidas con guion bajo.
- Si el atributo se guarda en la base de datos, aparece como una columna con el mismo nombre.

Ejemplo

Todas las tablas relacionadas con el módulo `website` empiezan por:

1	website_
---	----------

> website
> website_base_unit
> website_configurator_feature
> website_country_group_rel
> website_lang_rel
> website_menu
> website_page
> website_rewrite
> website_robots
> website_route
> website_sale_extra_field
> website_sale_payment_acquirer_onboarding_wi:
> website_snippet_filter
> website_track
> website_visitor

De esta manera resulta muy fácil localizar las tablas que pertenecen a un módulo concreto.

Si seleccionamos una tabla, pgAdmin nos muestra las columnas que contiene. Si además elegimos la opción **View/Edit Data**, podemos ver los datos que hay dentro. Esto nos ayuda a entender la estructura interna de Odoo.

> website_menu
> Columns (16)
id
name
url
page_id
new_window
sequence
website_id
parent_id
parent_path
mega_menu_content
mega_menu_classes
create_uid
create_date
write_uid
write_date
theme_template_id

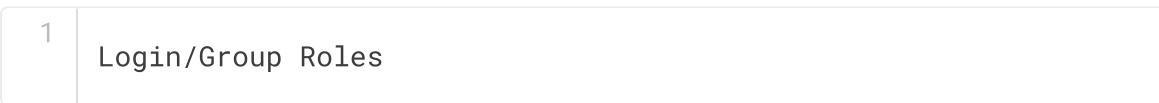
Por ejemplo, si consultamos la tabla de productos veremos exactamente los productos que tenemos dados de alta en el ERP. Lo que estamos haciendo realmente es ejecutar una consulta SQL.

Data Output Explain Messages Notifications																						
id	[PK] integer	message_main_attachment_id	integer	default_code	character varying	active	boolean	product_tmpl_id	integer	barcode	character varying	combination_indices	character varying	volume	numeric	weight	numeric	can_image_variant_1024_be_zoomed	boolean	create_uid	integer	create_time
1	1		[null] [null]	false		1	[null]				[null]		[null]	[null]	[null]	[null]	[null]	false		2	2022	
2	2		[null] [null]	true		2	2000000000008				[null]		[null]	[null]	[null]	[null]	[null]	false		2	2022	
3	3		[null] [null]	true		3	3000000000007				[null]		[null]	[null]	[null]	[null]	[null]	false		2	2022	

1.2.3. Los usuarios del SGBD

Además de las tablas, PostgreSQL también gestiona **usuarios**, que son quienes pueden acceder o no a las bases de datos.

En pgAdmin, dentro de la conexión que hemos creado, podemos abrir el apartado:



Ahí veremos los usuarios existentes en la base de datos.

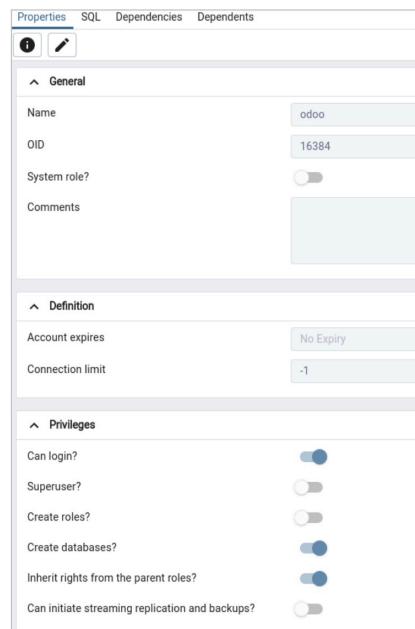
Login/Group Roles (10) <ul style="list-style-type: none"> odoo pg_execute_server_program pg_monitor pg_read_all_settings pg_read_all_stats pg_read_server_files pg_signal_backend pg_stat_scan_tables pg_write_server_files postgres 	<pre> 4 CREATE ROLE odoo WITH 5 LOGIN 6 NOSUPERUSER 7 INHERIT 8 CREATEDB 9 NOCREATEROLE 10 NOREPLICATION; </pre>
--	--

El usuario **odoo** es el que utiliza el ERP para conectarse a PostgreSQL y trabajar con las tablas.

Desde este mismo menú podemos:

- crear usuarios nuevos,
- modificarlos,
- asignar permisos,
- revisar qué operaciones puede realizar cada uno.

Si seleccionamos un usuario y abrimos **Properties**, aparecen todas sus características (si puede iniciar sesión, si puede crear bases de datos, si hereda permisos, etc.).



Si pulsamos en el ícono del **lápiz**, podremos editarlas.

2. Consultar la información en Odoo

En los apartados anteriores vimos cómo acceder a la base de datos. Ahora vamos a centrarnos en cómo consultar esa información desde el **cliente web de Odoo**, que es la herramienta que usaremos en el día a día.

Antes de empezar, conviene tener claro **qué datos maneja Odoo y cómo se organizan**.

2.1. La información en Odoo

Para poder modificar o crear nuevas funciones dentro del ERP necesitamos entender cómo estructura Odoo sus datos.

Odoo utiliza la arquitectura **MVC** (Modelo - Vista - Controlador):

✓ **Modelo**

Representa los datos del sistema.

En Odoo, cada modelo corresponde a una tabla de la base de datos y está definido mediante **clases en Python**.

✓ Vista

Define cómo se muestran esos datos al usuario.

Las vistas se escriben en **XML** y determinan si vemos un formulario, una lista, un kanban, etc.

✓ Controlador

Gestiona la lógica que conecta las vistas con los modelos.

En módulos sencillos no siempre se utiliza, pero cuando aparece está escrito también en **Python**.

Sabías que...?

Esta separación permite que podamos cambiar una vista sin tocar los datos o modificar campos del modelo sin rediseñar toda la interfaz.

En el cliente web veremos sobre todo **modelos y vistas**.

Los **controladores** forman parte del código del módulo y normalmente no se gestionan desde la interfaz.

Además de estos elementos básicos, veremos otros componentes importantes que nos permiten interactuar con Odoo desde el cliente web.

A. Menús

En el cliente web encontramos menús en la parte superior y en la zona izquierda.

Cuando pulsamos en uno de ellos, Odoo cambia de pantalla porque se ha realizado una acción concreta.

Cada menú está asociado a una **acción**, que es la que indica qué modelo se abre y qué vistas se muestran al usuario.

Estos menús nos llevan a módulos, listas de datos o pantallas específicas.



B. Menú principal

Cada vez que instalamos o creamos un módulo nuevo, Odoo añade un nuevo elemento al **menú principal**.



Este menú se organiza mediante **submenús**, que aparecen en la parte superior de la pantalla cuando entramos en un módulo.



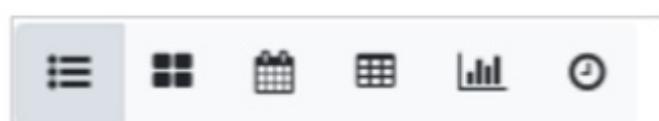
Estos submenús no contienen datos por sí mismos, sino que sirven como puntos de acceso a listas, formularios u otras pantallas del sistema.

Por ejemplo, dentro del módulo *CRM* podemos ver submenús como *Ventas*, *Oportunidades*, *Informes*, etc.

C. Vistas

Las vistas son las diferentes formas de mostrar los datos de un modelo.

Un mismo modelo puede tener varias vistas asociadas y podemos cambiar entre ellas según lo que necesitemos consultar.



Todas estas vistas muestran los mismos datos, pero organizados de forma distinta según la tarea que estemos realizando.

Tipos de vistas más comunes:

- **Lista (árbol)**: muestra varios registros a la vez, en forma de tabla.
- **Formulario**: muestra un solo registro.
- **Kanban**: tarjetas en columnas, útil para flujos de trabajo.
- **Calendario**: muy útil para eventos o tareas con fechas.

La elección de una vista u otra cambia totalmente cómo vemos la información.

Ejemplo: vista *Kanban* frente a vista *Árbol* en el módulo de productos.

Nombre del producto	Referencia interna	Responsable	Precio de venta	Coste	Cantidad a mano	Cantidad pronosticada
<input type="checkbox"/> Placa base intel		<input checked="" type="radio"/> Administrador	54,00 €	0,00 €	0,00	-3,00
<input type="checkbox"/> RAM		<input checked="" type="radio"/> Administrador	36,00 €	0,00 €	0,00	-1,00

D. Acciones

Las **acciones** son las instrucciones que Odoo ejecuta cuando el usuario realiza algo:

iniciar sesión, abrir una ventana, pulsar un botón...

Una acción siempre tiene algunos atributos importantes:

- **Tipo**: indica qué clase de acción es y qué campos necesita.
- **Nombre**: descripción que identifica la acción en la interfaz.

Una de las acciones más usadas es `ir.actions.act_window`, que sirve para abrir un modelo utilizando sus vistas.

Cada acción de este tipo puede tener asociadas varias vistas.

E. Modelos

Un **modelo** es una clase en Python que se relaciona con una o varias tablas de la base de datos.

Cada modelo define:

- sus **campos** (que serán las columnas de la tabla),
- sus **relaciones** con otros modelos,
- su **comportamiento** (funciones escritas en Python).

Normalmente, un modelo se asocia a una única tabla, aunque también puede participar en relaciones más complejas.

Desde el cliente web no modificamos el comportamiento del modelo, pero sí podemos consultar su estructura y sus campos.

Ejemplo típico:

El modelo *res.partner* almacena contactos. En esta tabla encontramos clientes, proveedores y empleados, diferenciados por campos internos.

F. Identificadores

Todos los elementos almacenados en la base de datos tienen un **ID** único.

Este ID es un número que Odoo asigna automáticamente y sirve como referencia interna.

Además existen los **identificadores externos**, que se usan sobre todo cuando importamos datos o instalamos módulos. Por ejemplo, cuando un módulo necesita referirse a un menú, una vista o un informe, utiliza su identificador externo en lugar del ID numérico interno.

Permiten referenciar un registro mediante un nombre en lugar de un número.

Cuando Odoo encuentra un identificador externo, lo traduce al ID real que tiene en la base de datos.

G. Informes

Odoo puede generar informes para mostrar datos del ERP en diferentes formatos.

Los informes se crean con **HTML/QWeb**, un lenguaje de plantillas que permite diseñar el aspecto final del documento.

Cuando queremos descargar un informe en PDF, Odoo utiliza la librería **wkhtmltopdf**.

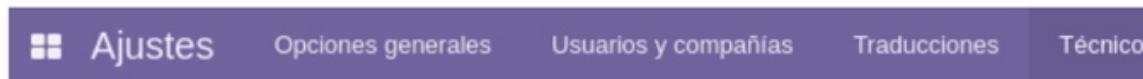
- ▶ Actividad: Instalar **wkhtmltopdf**

2.2. Consultar la información

Una vez conocemos qué tipos de datos maneja Odoo, podemos empezar a localizarlos dentro del **cliente web**. Esta parte es clave, ya que muchas veces necesitamos revisar información antes de modificarla o crear funciones nuevas.

Para acceder a los elementos más técnicos y poder editarlos, es necesario **activar el modo desarrollador**. Este modo nos permite ver los elementos internos de Odoo que normalmente están ocultos, como modelos, vistas, acciones e informes.

Lo podemos activar desde el menú **Ajustes**, en la vista general de configuración.



Cuando el modo desarrollador está activo, aparece un nuevo menú llamado **Técnico**, desde el cual podemos acceder a prácticamente cualquier elemento que forma parte del sistema: modelos, vistas, acciones, secuencias, informes, reglas de acceso y mucho más.

Dentro del menú Técnico veremos un listado muy amplio de componentes. La cantidad total depende de los módulos instalados, ya que cada módulo añade sus propios modelos, vistas, acciones, informes, etc.

Conversaciones	Teléfono / SMS	Relaciones ManyToMany
Mensajes	SMS	Adjuntos
Subtipo	SMS Templates	Registro
Valores de seguimiento	Lista negra del teléfono	Profiling
Tipos de actividad	Acciones	Automatización
Actividades	Acciones	Acciones planificadas
Notificaciones	Informes	Scheduled Actions Triggers
Seguidores	Acciones de ventana	Informes
Lista negra de correo electrónico	Acciones de servidor	Formato de papel
Valoraciones	Asistentes de configuración	Informes
User Settings	Valores por defecto del usuario	Informes
Clients	IAP	Secuencias e identificadores
RTC sessions	Cuentas IAP	Identificadores externos
ICE servers	Interfaz de usuario	Secuencias
Message Reactions	Elementos de menú	Parámetros
Correo electrónico	Vistas	Parámetros del sistema
Correos electrónicos	Vistas personalizadas	Propiedades de la compañía
Servidores de correo saliente	Filtros de usuario	Seguridad
Servidores de correo entrante	Recorridos	Reglas de registro
Plantillas de correo electrónico	Estructura de la base de datos	Permisos de acceso
Seudónimos	Precisión decimal	Recurso
Canales	Activos	Tiempos de Trabajo
Canales/Socios	Modelos	Resource Time Off
Snailmail Letters	Campos	Recursos
Correos electrónicos de Resumen	Selección de campos	Calendario
Consejos del Resumen	Restricciones del modelo	Tipos de reunión

Cuando entramos en un elemento concreto, como **Menús**, podemos ver en la parte superior derecha el total de elementos disponibles y una tabla con todos ellos. Este listado nos ayuda a entender cómo organiza Odoo toda la estructura del ERP.

Elementos de menú	
<input type="button" value="Crear"/>	<input type="button" value="Importar"/>
<input type="button" value=""/>	<input type="button" value=""/>
<input type="checkbox"/>	Menú
<input type="checkbox"/>	+ Tableros/Tableros
<input type="checkbox"/>	+ Ajustes/Usuarios y compañías/Usuarios
<input type="checkbox"/>	+ Ajustes/Opciones Generales
<input type="checkbox"/>	+ CRM/Configuración/Ajustes
<input type="checkbox"/>	+ CRM/Configuración/Flujo/Etapas
<input type="checkbox"/>	+ Facturación / Contabilidad/Configuración/Contabilidad analítica/Cuentas analíticas
<input type="checkbox"/>	+ Facturación / Contabilidad/Configuración/Ajustes
<input type="checkbox"/>	+ Inventario/Información general
<input type="checkbox"/>	+ Inventario/Configuración/Ajustes
<input type="checkbox"/>	+ Punto de venta/Configuración/Ajustes

A. Vistas

Si abrimos el apartado de **Vistas**, veremos que cada modelo tiene varias vistas asociadas.

Nombre de la vista	Tipo de vista	Modelo	ID externo	Vista heredada
<input type="checkbox"/> res.config.settings.view.form.inherit.base.setup	Formulario	res.config.settings	base_setup.res_config_settings_view_form	res.config.settings.view.form
<input type="checkbox"/> + Upload Bank Statements	Formulario	account.bank.statement.import	account_bank_statement_import.account_bank_statement_import_view	
<input type="checkbox"/> + account.analytic.line.form	Formulario	account.analytic.line	analytic.view_account_analytic_line_form	
<input type="checkbox"/> + account.bank.statement.form	Formulario	account.bank.statement	account.view_bank_statement_form	
<input type="checkbox"/> + account.journal.form	Formulario	account.journal	account.view_account_journal_form	
<input type="checkbox"/> + account.journal.group.form	Formulario	account.journal.group	account.view_account_journal_group_form	
<input type="checkbox"/> + account.journal.group.tree	Árbol	account.journal.group	account.view_account_journal_group_tree	
<input type="checkbox"/> + account.journal.kanban	Kanban	account.journal	account.account_journal_view_kanban	

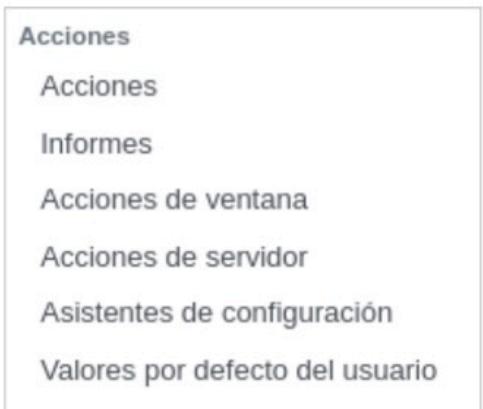
Esto sucede porque diferentes tareas requieren formas distintas de mostrar la información.

Además, algunas vistas están **heredadas** de otras, lo que permite personalizar solo una parte sin volver a crearla desde cero.

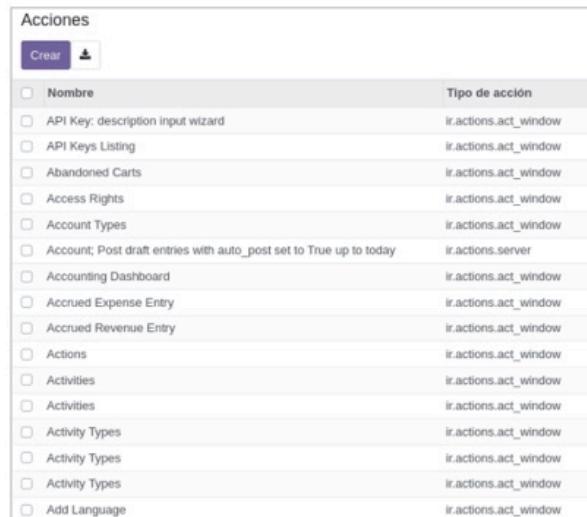
Este sistema de herencia es muy habitual en Odoo y facilita mucho la adaptación del sistema a las necesidades de cada empresa.

B. Acciones

En el menú Técnico también encontramos las **Acciones**, que son las instrucciones que definen lo que ocurre cuando el usuario hace algo dentro del sistema.



Acciones	
Acciones	
Informes	
Acciones de ventana	
Acciones de servidor	
Asistentes de configuración	
Valores por defecto del usuario	



Acciones	
Nombre	Tipo de acción
<input type="checkbox"/> API Key: description input wizard	ir.actions.act_window
<input type="checkbox"/> API Keys Listing	ir.actions.act_window
<input type="checkbox"/> Abandoned Carts	ir.actions.act_window
<input type="checkbox"/> Access Rights	ir.actions.act_window
<input type="checkbox"/> Account Types	ir.actions.act_window
<input type="checkbox"/> Account; Post draft entries with auto_post set to True up to today	ir.actions.server
<input type="checkbox"/> Accounting Dashboard	ir.actions.act_window
<input type="checkbox"/> Accrued Expense Entry	ir.actions.act_window
<input type="checkbox"/> Accrued Revenue Entry	ir.actions.act_window
<input type="checkbox"/> Actions	ir.actions.act_window
<input type="checkbox"/> Activities	ir.actions.act_window
<input type="checkbox"/> Activities	ir.actions.act_window
<input type="checkbox"/> Activity Types	ir.actions.act_window
<input type="checkbox"/> Activity Types	ir.actions.act_window
<input type="checkbox"/> Activity Types	ir.actions.act_window
<input type="checkbox"/> Add Language	ir.actions.act_window

Las acciones pueden clasificarse según su función:

- **Acciones normales**
- **Acciones de ventana (act_window)**: las más frecuentes, porque abren vistas de un modelo.
- **Acciones de servidor**
- **Asistentes de configuración**
- **Acciones vinculadas a informes**

En la tabla de acciones podemos ver su nombre interno, el tipo, el modelo al que afecta y la vista que abre.

Esta información es muy útil cuando personalizamos módulos.

C. Los modelos

Los modelos son uno de los elementos más importantes del sistema, porque representan las tablas de la base de datos.

Modelos			
	Descripción del modelo	Tipo	Modelo transitorio
<input type="checkbox"/> Modelo		Objeto base	
<input type="checkbox"/> stock.picking	Albarán	Objeto base	
<input type="checkbox"/> account.move	Asiento contable	Objeto base	
<input type="checkbox"/> calendar.event	Calendario de eventos	Objeto base	
<input type="checkbox"/> mail.channel	Canal de conversaciones	Objeto base	
<input type="checkbox"/> res.partner	Contacto	Objeto base	
<input type="checkbox"/> account.account	Cuenta	Objeto base	
<input type="checkbox"/> account.analytic.account	Cuenta analítica	Objeto base	
<input type="checkbox"/> stock.scrap	Desechar	Objeto base	
<input type="checkbox"/> account.journal	Diario	Objeto base	
<input type="checkbox"/> crm.team	Equipo de ventas	Objeto base	
<input type="checkbox"/> account.bank.statement	Extracto bancario	Objeto base	
<input type="checkbox"/> mail.thread.cc	Gestión de correo electrónico CC	Objeto base	
<input type="checkbox"/> mail.thread	Hilo de mensajes	Objeto base	

En el listado de modelos podemos ver:

- el nombre del modelo,
- su descripción,
- el número de registros que contiene.

Si seleccionamos un modelo, Odoo nos muestra todos sus **campos**, que equivalen a las columnas de la tabla.

Descripción del modelo	Atributo de producto	Tipo	Objeto base			
Modelo	product.attribute	En las aplicaciones	product, sale, website_sale			
Pedido	sequence, id					
Modelo transitorio						
Hilo de mensajes						
Actividad de correo						
Lista negra de correo						
Campos Permisos de acceso Reglas de registro Notas Vistas Formularios de página web						
Nombre de campo	Etiqueta de campo	Tipo de campo	Requerido	Sólo lectura	Indexado	Tipo
__last_update	Última modificación el	Fecha y hora	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
attribute_line_ids	Líneas	one2many	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
create_date	Creado el	Fecha y hora	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
create_uid	Creado por	many2one	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
create_variant	Modo de Creación de Variantes	Selección	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
display_name	Nombre mostrado	Carácter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
display_type	Tipo de pantalla	Selección	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
id	ID	entero	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
name	Atributo	Carácter	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
number_related_products	Number Related Products	entero	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
product_tmpl_ids	Productos relacionados	many2many	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
sequence	Secuencia	entero	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
value_ids	Valores	one2many	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
visibility	Visibilidad	Selección	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
write_date	Última actualización el	Fecha y hora	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base
write_uid	Última actualización por	many2one	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	Campo base

Cada campo indica:

- su tipo (texto, número, relación...),
- si es obligatorio,

- si aparece en vistas,
- si se almacena en base de datos, etc.

D. Identificadores

Dentro del menú Técnico también encontramos los **identificadores externos**, que se utilizan para hacer referencia a registros concretos desde código, archivos XML o datos importados.

Podemos ver:

- el identificador externo,
- el registro al que apunta,
- el módulo que lo creó,
- el número de secuencia.

Estos identificadores permiten que Odoo reconozca elementos incluso si su ID interno cambia.

Identificadores externos			
	Nombre mostrado	Nombre del modelo	ID de registro
<input type="checkbox"/> Id. completo			
<input type="checkbox"/> account.account_tag_financing	Financing Activities	account.account.tag	2
<input type="checkbox"/> account.account_tag_investing	Investing & Extraordinary Activities	account.account.tag	3
<input type="checkbox"/> account.account_tag_operating	Operating Activities	account.account.tag	1
<input type="checkbox"/> account.data_account_off_sheet	Hoja fuera de balance	account.account.type	18
<input type="checkbox"/> account.data_account_type_credit_card	Tarjeta de Crédito	account.account.type	4
<input type="checkbox"/> account.data_account_type_current_assets	Activos Circulantes	account.account.type	5
<input type="checkbox"/> account.data_account_type_current_liabilities	Pasivos Circulantes	account.account.type	9
<input type="checkbox"/> account.data_account_type_depreciation	Depreciación	account.account.type	16
<input type="checkbox"/> account.data_account_type_direct_costs	Costo de ingresos	account.account.type	17
<input type="checkbox"/> account.data_account_type_equity	Capital	account.account.type	11

E. Informes

Por último, tenemos los **informes**, que permiten generar documentos basados en datos del ERP.

En la tabla de informes podemos ver:

- nombre del informe,
- modelo al que pertenece,
- tipo de informe (por ejemplo, QWeb),
- plantilla utilizada,
- formato de salida (PDF, HTML...).

Odoo genera los informes en PDF mediante la librería **wkhtmltopdf**, que convierte la plantilla HTML/QWeb en un documento imprimible.

Informes				
	Nombre del modelo	Tipo de acción	Nombre de plantilla	Tipo de informe
<input type="checkbox"/> Nombre				
<input type="checkbox"/> Albarán	stock.picking	ir.actions.report	stock.report_deliveryslip	PDF
<input type="checkbox"/> Auditoría de Libros	account.common.journal.report	ir.actions.report	account.report_journal	PDF
<input type="checkbox"/> Código de barras de ubicación	stock.location	ir.actions.report	stock.report_location_barcode	PDF
<input type="checkbox"/> Código de barras del paquete (PDF)	stock.quant.package	ir.actions.report	stock.report_package_barcode_small	PDF
<input type="checkbox"/> Código de barras del paquete (ZPL)	stock.quant.package	ir.actions.report	stock.label_package_template_view	Texto
<input type="checkbox"/> Código de barras del paquete con contenido	stock.quant.package	ir.actions.report	stock.report_package_barcode	PDF
<input type="checkbox"/> Embalaje del Producto (PDF)	product.packaging	ir.actions.report	product.report_packagingbarcode	PDF
<input type="checkbox"/> Etiqueta del Producto (PDF)	product.template	ir.actions.report	product.report_producttemplatelabel	PDF
<input type="checkbox"/> Etiqueta del Producto (PDF)	product.template	ir.actions.report	product.report_producttemplatelabel_dymo	PDF
<input type="checkbox"/> Etiqueta del producto (ZPL)	product.product	ir.actions.report	stock.label_product_product_view	Texto

3. Creación y modificación de datos

Hasta ahora hemos aprendido a **movernos por Odoo** y a localizar sus distintos elementos técnicos. En este apartado damos un paso más y empezamos a **adaptar Odoo a las necesidades reales de una empresa**, modificando lo que ya existe y creando nuevos elementos cuando el sistema no cubre un caso concreto.

A lo largo de este apartado trabajaremos con tres ideas clave:

- **Cómo acceden los usuarios a la información**, a través de los menús.
- **Cómo se muestran y se editan los datos**, mediante las vistas.
- **Qué información gestiona el sistema**, creando entidades propias (modelos).

El objetivo no es memorizar rutas ni opciones, sino **entender el papel que juega cada elemento dentro de Odoo** y cómo se relacionan entre sí. De este modo, al finalizar el tema deberás ser capaz de **modificar la navegación, ajustar la interfaz y crear una entidad nueva**, siguiendo un orden lógico y coherente, similar al que se utiliza en un entorno profesional.

En los siguientes apartados veremos algunos ejemplos representativos. Aunque Odoo permite muchas más configuraciones, los casos que se presentan aquí sirven como base para comprender el proceso y poder aplicarlo después a otras situaciones.

3.1. Modificar o crear un elemento de menú

Los **menús** determinan **cómo acceden los usuarios a la información dentro de Odoo**. A través de ellos se organiza la navegación y se decide qué opciones están visibles y desde dónde se puede llegar a cada pantalla.

Modificar o crear un menú resulta útil, por ejemplo, cuando:

- queremos **reorganizar Odoo** para que sea más claro para un departamento concreto,
- necesitamos **dar acceso a una nueva funcionalidad**,
- o deseamos que una misma vista sea accesible desde distintos puntos del sistema.

Todos los elementos de menú se gestionan desde la ruta:

Ajustes > Técnico > Interfaz de usuario > Elementos de menú

Crear un nuevo elemento de menú

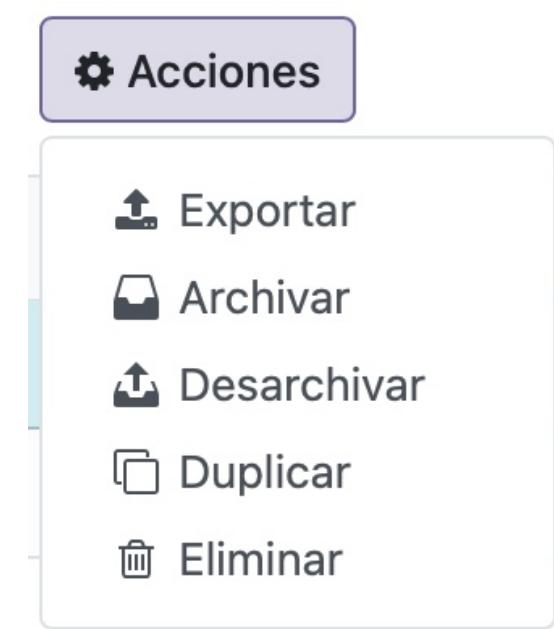
Desde esta pantalla podemos crear un menú nuevo pulsando el botón **Nuevo**.



Al crear un menú nuevo todavía no estamos definiendo qué datos se muestran, sino **el punto de acceso** desde el que el usuario podrá llegar a una determinada vista o funcionalidad.

Modificar un menú existente

También podemos **editar un menú ya creado**. Para ello, seleccionamos el elemento deseado y pulsamos el botón **Acciones**.



Desde aquí podemos:

- **Eliminar** el menú si ya no es necesario.
- **Duplicarlo**, opción especialmente interesante cuando queremos crear un menú muy similar a otro ya existente.

Duplicar un menú permite **reutilizar la configuración** y realizar pequeños cambios sin partir desde cero, algo habitual cuando se quiere ofrecer el mismo acceso desde distintos puntos de Odoo.

Editar los datos del menú

Al editar o duplicar un menú, accedemos a su ficha de configuración:

Menú	Productos	Ruta completa	Inventario/Productos/Productos
Menú padre	Inventario/Productos	Acción	ir.actions.act_window ▾ Productos
Secuencia	1	Archivo de ícono web	
Imagen de ícono web		<input type="button" value="Suba su archivo"/>	

En esta pantalla podemos realizar varias modificaciones importantes:

- **Nombre:** es el texto que verá el usuario en el menú.
- **Menú padre:**
 - Si lo dejamos en blanco, el elemento aparecerá en el **menú principal**.
 - Si indicamos otro menú como padre, el nuevo elemento quedará **anidado dentro de ese menú**, formando una jerarquía.

Esta organización permite estructurar Odoo de forma más clara, agrupando opciones relacionadas y evitando menús excesivamente largos o confusos.

En este punto es importante recordar que **el menú no contiene los datos**, sino que actúa como enlace hacia una acción que abrirá una vista, aspecto que se trabajará en los siguientes apartados.

3.2. Modificar vistas en Odoo

Las **vistas** determinan **cómo se muestran los datos al usuario** y cómo puede interactuar con ellos. Mientras que los modelos definen la información y los menús permiten acceder a ella, las vistas se centran en la **presentación y edición de los datos** dentro de la aplicación.

En Odoo, las vistas se gestionan desde la siguiente ruta:

Ajustes > Técnico > Interfaz de usuario > Vistas

Crear una nueva vista

Desde esta pantalla podemos crear una vista nueva pulsando el botón **Nuevo**.



Al crear una vista estamos definiendo **qué campos se muestran, en qué orden y con qué formato**, pero no estamos creando datos nuevos. Una misma información puede mostrarse de distintas formas según la vista que se utilice.

Modificar una vista existente

También es posible **editar vistas que ya existen** en el sistema. Para ello, seleccionamos la vista que queremos modificar y accedemos a su edición.

Nombre de la vista: product.pricelist.kanban

Tipo de vista: Kanban

Modelo de la vista: Lista de precios

Secuencia: 16

Activo:

Vista heredada

Modo de herencia de la vista: Vista base

Datos del modelo: product.pricelist.kanban

ID externo: product.product_pricelist_view_kanban

Consideré que no recomendamos editar la arquitectura de la vista estándar. Las próximas actualizaciones del módulo sobreescribirán los cambios que haga. Recomendamos que aplique las modificaciones que haga a vistas heredadas o mediante la personalización en Odoo Studio.

Arquitectura Permisos de acceso Vistas heredadas

```

1 - <kanban class="o_kanban_mobile" sample="1">
2 -   <templates>
3 -     <t t-name="card" class="flex-row fw-bolder">
4 -       <field name="name"/>
5 -       <div class="ms-auto align-items-center">
6 -         <i class="fa fa-money" role="img" aria-label="Currency" title="Currency"/> <field name="currency_id"/>
7 -       </div>
8 -     </t>
9 -   </templates>
10  </kanban>

```

Al abrir una vista podemos ver distintos datos asociados:

- **Nombre** de la vista.
- **Modelo** al que pertenece, es decir, el tipo de datos que muestra.
- **Estructura**, definida mediante código **XML**.

La estructura XML de una vista

El aspecto y el comportamiento de una vista vienen definidos por su **estructura XML**. Este código es el que indica a Odoo:

- qué campos se muestran,
- si se trata de una vista de formulario, lista (tree) u otro tipo,
- cómo se organizan visualmente los elementos en pantalla.

Para modificar realmente una vista es necesario **editar su XML**, ya que cualquier cambio en la estructura visual se refleja directamente en este código.

En este módulo trabajaremos con vistas sencillas, centrándonos en:

- vistas de **lista (tree)**, para mostrar varios registros,
- y vistas de **formulario**, para crear y editar un registro concreto.

El objetivo no es dominar el XML en profundidad, sino **entender la relación entre el código de la vista y lo que el usuario ve en Odoo**, algo clave cuando se adapta la aplicación a las necesidades de una empresa.

3.3. Crear una entidad nueva en Odoo

En muchas ocasiones, una empresa necesita **gestionar información que Odoo no trae preparada por defecto**. En esos casos es necesario crear una **entidad nueva**, es decir, un nuevo tipo de dato que el sistema pueda almacenar, mostrar y gestionar.

Crear una entidad en Odoo implica construirla paso a paso, siguiendo siempre el mismo orden. Este proceso refleja cómo se trabaja en un entorno profesional y ayuda a entender la relación entre los distintos elementos técnicos del sistema.

El proceso completo consta de cuatro pasos:

01	02	03	04
Crear el modelo	Crear las vistas	Crear la acción	Crear el n

Cada uno de estos pasos cumple una función concreta:

- el **modelo** define qué datos existen,
- las **vistas** indican cómo se muestran,
- la **acción** permite abrir esas vistas,
- y el **menú** da acceso al usuario final.

A. EL MODELO

El primer paso es crear el **modelo**, que representa la estructura de los datos que queremos almacenar.

Accedemos a:

Ajustes > Técnico > Estructura de la base de datos > Modelos

Desde aquí creamos un modelo nuevo. Al hacerlo, Odoo asigna automáticamente un nombre que empieza por **x**, lo que indica que se trata de un modelo personalizado y no perteneciente a ningún módulo estándar.

A continuación:

- eliminamos el campo que aparece por defecto,
- y añadimos los campos necesarios para nuestra entidad.

Los nombres de los campos también deben comenzar por **x_**, siguiendo la convención de Odoo para elementos personalizados.

En la parte inferior aparece un botón:

Crear un menú

Este botón permite generar vistas automáticamente, pero **no lo utilizaremos**, ya que las vistas se crearán de forma manual para entender mejor el proceso.

Guardamos el modelo y observamos que **Odoo añade automáticamente otros campos necesarios para la gestión interna del sistema**. Estos campos no los hemos creado nosotros, pero forman parte del funcionamiento habitual de Odoo.

B. LAS VISTAS

Una vez creado el modelo, necesitamos definir **cómo se van a mostrar los datos al usuario**.

Accedemos a:

Ajustes > Técnico > Interfaz de usuario > Vistas

Para cada modelo se crean, como mínimo, dos vistas:

- una vista de **lista (tree)**, que muestra varios registros,
- y una vista de **formulario**, que permite crear o editar un registro concreto.

Estas vistas no heredan de ninguna otra. En cada una de ellas definimos manualmente la **estructura XML**, indicando el modelo al que pertenecen y los campos que se mostrarán.

En la vista de formulario se utiliza la etiqueta `<group>` para agrupar los campos y mejorar la presentación visual. Es importante incluir **todos los campos del modelo**, ya que desde esta vista se introducirán los datos que se almacenarán en la base de datos.

C. ACCIÓN

El siguiente paso es crear la **acción**, que será la encargada de abrir las vistas que acabamos de definir.

Las acciones se crean en:

Ajustes > Técnico > Acciones > Acciones de ventana

Desde aquí pulsamos **Nuevo** y completamos los campos necesarios. En la opción **Modo Vista** indicamos qué vistas se abrirán y en qué orden. El orden elegido será el mismo que verá el usuario cuando acceda a la entidad desde Odoo web.

La acción actúa como un **enlace entre los datos y la interfaz**, pero todavía no es visible para el usuario final.

D. MENÚ

El último paso en la creación de una entidad es **definir el menú**, que permitirá acceder a ella desde la interfaz de Odoo. El menú actúa como punto de entrada para el usuario y enlaza la navegación con la acción que abre las vistas asociadas al modelo.

Los elementos de menú se gestionan desde:

Ajustes > Técnico > Interfaz de usuario > Elementos de menú

Desde esta pantalla se puede crear un nuevo menú y configurar sus principales propiedades:

- **Nombre:** texto que verá el usuario en la interfaz.
- **Menú padre:**
 - si se deja vacío, el menú aparecerá en el nivel principal,
 - si se selecciona otro menú, el nuevo elemento quedará anidado dentro de él.
- **Acción:** indica qué se abrirá al pulsar sobre el menú.

En el campo **Acción** se selecciona una **acción de ventana** (`ir.actions.act_window`), que será la encargada de abrir las vistas creadas previamente. Esta acción es el enlace entre el menú y la información que se va a mostrar al usuario.

Visibilidad del menú y permisos en Odoo 18

En Odoo 18, la visibilidad de un menú depende de los **permisos y grupos de acceso** del usuario. Que un menú esté correctamente creado no implica que sea visible para todos los perfiles.

Para comprobar de forma rápida que el menú, la acción y las vistas funcionan correctamente, se puede acceder como **superusuario**, ya que este perfil no tiene restricciones de permisos.

En Odoo 18, el acceso como superusuario se realiza accediendo directamente a la siguiente ruta desde el navegador:

`http://TU_SERVIDOR/web/become`

Por ejemplo:

- en un entorno local:

`http://localhost:8069/web/become`

Tras acceder a esta ruta, la interfaz se recarga con privilegios de superusuario. Al actualizar el navegador, el nuevo menú aparecerá si está correctamente configurado, permitiendo acceder a la entidad creada y comprobar su funcionamiento.

Para volver al modo normal, basta con **cerrar sesión** y volver a iniciar sesión con el usuario habitual.