

SISTEMA DE GESTIÓN EMPRESARIAL

TÉCNICO EN DESARROLLO DE APLICACIONES MULTIPLATAFORMA

Administración e informes en Odoo

Usuarios y permisos

Qweb

Errores comunes

Agregar campos en formularios

En este punto agregaremos un **campo a un formulario** existente desde Odoo. Es simple, pero el problema es que, si actualizamos el módulo o actualizamos la versión de Odoo, este cambio se perderá.

Por ejemplo, al formulario de '**contactos**' le agregaremos el **campo "Facebook"**, justo **debajo del campo "enlace a página web"**.

Para ello, lo primero que deberemos hacer es colocar Odoo en **modo desarrollador**. A continuación, identificaremos qué modelo usa el formulario pulsando sobre '**metadatos**'.

En nuestro caso es el '**res.partner**'. Seguidamente, nos vamos al modelo en concreto (**ajustes/técnico/estructura de la base de datos/modelos**) y, tras **pulsar editar**, aparecerá al final de todos los campos un vínculo denominado '**agregar línea**', el cual pulsaremos. Esto nos abrirá una nueva ventana donde deberemos informar los **atributos del nuevo campo**.

Si observamos detenidamente, todos los **campos personalizados** que se creen desde el front-end **comenzarán su nombre con 'x_'**. Una vez **guardados** todos los cambios Odoo habrá **creado el nuevo campo en la base de datos**.

Notas propias:

En Odoo, puedes **ver qué modelo está asociado a un formulario** siguiendo estos pasos:

1. Inicia sesión en Odoo como administrador o usuario con permisos suficientes.
2. Navega a la **aplicación que contiene el formulario** del que deseas conocer el modelo asociado.
3. **Abre el formulario** en cuestión.
4. Una vez que estés en el formulario, puedes hacer **clic en el botón "Editar" o "Modificar"** si tienes permisos para editar los registros.
5. En la **URL del navegador**, verás algo como:

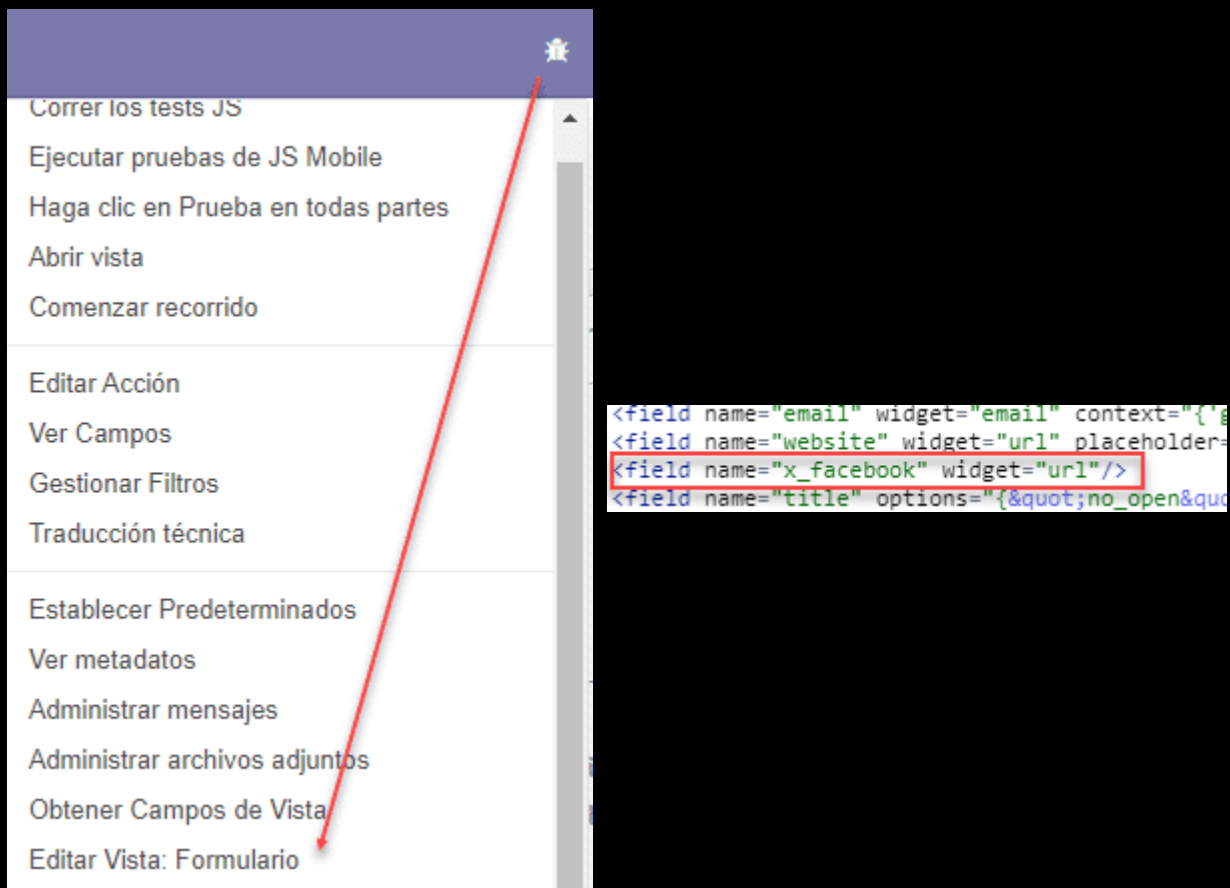
`/web#id=123&view_type=form&model=sale.order`, donde *sale.order* es el **modelo asociado** al formulario en este ejemplo.

6. Alternativamente, si tienes acceso al **modo de desarrollador** en Odoo, puedes habilitarlo y luego, al hacer **clic derecho** sobre cualquier **campo del formulario**, seleccionar la opción **"Inspeccionar"** o "Inspect" para ver el **código HTML asociado**. Dentro del código HTML, generalmente encontrarás la **referencia al modelo** asociado.

Agregar el nuevo campo a la vista del formulario

Para agregar el nuevo campo **a la vista** del formulario, volveremos al formulario de 'contactos' y nos posicionaremos sobre el campo "enlace a página web" para ver cuál es el **nombre del campo en concreto**, ya que el nuevo campo lo posicionaremos tras él. En este caso es 'website':

Una vez encontrado, pulsaremos sobre '**Editar Vista: Formulario**' y buscaremos la etiqueta de 'website' para colocar justo debajo de la etiqueta del nuevo campo, tal y como se muestran en las figuras 7 y 8.



Como se puede observar, se le ha agregado el atributo **widget="url"**, ya que lo que contendrá, será un **vínculo a una página web**, con la funcionalidad de pulsar y navegar desde ese punto. Con esto ya habremos agregado a la vista el nuevo campo.

Finalmente, se guardarán todos los cambios y se refrescará el formulario. En ese momento, tal y como se aprecia en la figura 9, el nuevo campo 'Facebook' ya aparecerá y le podremos agregar la información.

Resultado de agregar el nuevo campo

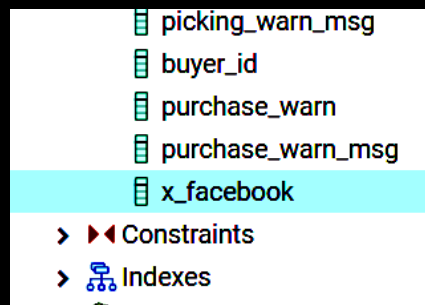
Facebook	
indica el enlace a su dirección de facebook	
Field: x_facebook	+1 (650) 691-3277 SMS
Objeto: res.partner	info@yourcompany.example.com
Tipo: text	http://www.example.com
Widget: URL (url)	
Facebook	https://www.facebook.com/yourcompany
Categorías	

Verificación de la existencia del nuevo campo en la base de datos

Ahora podremos pasar a comprobar que el campo existe en la base de datos y que, además, el dato se ha almacenado correctamente.

Para ello, abriremos **PgAdmin** y desplegaremos la tabla, que en este caso es res_partner.

Tal y como se aprecia en la figura, el campo x_facebook **se ha agregado correctamente al final de la tabla** res_partner.



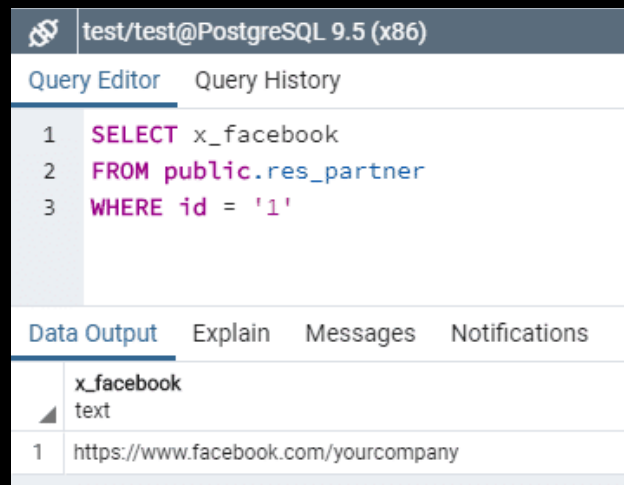
Ahora, para comprobar que se ha informado correctamente el valor, vamos a buscar **cuál es el ID** del elemento que hemos modificado para así **seleccionar solo ese registro** con una **sentencia** que selecciona a la tabla res_partner.

Tal y como ya sabemos, el ID lo podemos ver en los **metadatos** del elemento:

Buscando el ID del elemento en su metadata

Metadata (res.partner)	
ID:	1
XML ID:	base.main_partner
Sin actualización:	true (cambiar)

Ya solo nos queda realizar la **sentencia SQL** con la herramienta '**Query Tool**' de PgAdmin para obtener la información.



The screenshot shows the PgAdmin Query Tool interface. At the top, the connection is identified as 'test/test@PostgreSQL 9.5 (x86)'. Below this, there are two tabs: 'Query Editor' and 'Query History'. The 'Query Editor' tab is active, displaying a SQL query with line numbers 1, 2, and 3. The query is:
1 SELECT x_facebook
2 FROM public.res_partner
3 WHERE id = '1'
Below the query editor, there are four tabs: 'Data Output', 'Explain', 'Messages', and 'Notifications'. The 'Data Output' tab is active, showing a table with one column, 'x_facebook', of type 'text'. The first row of data contains the URL 'https://www.facebook.com/yourcompany'.

	x_facebook
1	https://www.facebook.com/yourcompany

Exportar datos

Se necesita **extraer en un fichero Excel**, el nombre del producto, el coste, el precio de venta y la cantidad a mano (en ese orden).

Para ello, en el módulo de Ventas, nos dirigiremos a los productos. A continuación, abriremos la vista de lista (**View list**), **seleccionaremos** todos los **productos** y pulsaremos sobre **acción/exportar**:

Referencia interna	Nombre	Coste	Cantidad a mano	Cantidad pronosticada
FURN_6666	Acoustic Bloc Screens	210,00	0,000	0,000
Deposit	Adelanto	100,00		
FURN_0002	Alfombrilla de Escritorio	0,00	0,000	0,000

Seguidamente, en la ventana emergente, **seleccionaremos los campos** que queremos exportar y su **orden**, el **formato** de salida del documento (XLSX en este caso) y finalmente pulsaremos sobre el botón 'Exportar'.

☐ Quiero actualizar datos (exportación compatible con importación)

Formato de exportación: ☒ XLSX ☐ CSV

Campos disponibles

Búsqueda

Abasteciendo cant. max. +

Abasteciendo cant. min. +

of LdM donde son usados +

Variantes de producto +

Campos a exportar

Plantilla: ▼

+ Nombre

+ Coste

+ Precio de venta

+ Cantidad a mano

Dando como resultado un fichero Excel.

Administrador de usuarios

En el presente punto trataremos la administración de usuarios en Odoo, abordaremos un primer bloque exponiendo cómo crearlos y desactivarlos, y a continuación nos adentraremos en la gestión de permisos:

Crear usuarios: La creación de un nuevo usuario se llevará a cabo desde el **módulo de ajustes**, en **'usuarios y compañías/Usuarios'** o en **'administrar usuarios'**:

Al pulsar sobre el botón 'Crear' lo primero que deberemos informar será el **nombre del nuevo usuario**, su dirección de **email** (el que usará para **iniciar sesión** en Odoo) y su **fotografía**.

Eliminar usuarios: Para eliminar un usuario bastará con seleccionarlo y pulsar en **acción/suprimir** (*Nota: no eliminar al usuario principal (admin)):

Gestión de grupos: Los Grupos en Odoo son bloques donde encapsular una **serie de reglas y permisos** para luego ser **asignados a usuarios**. Un **usuario puede tener asignado varios** grupos al igual que **un grupo puede estar asignado a varios** usuarios.

La creación, eliminación o edición de éstos se llevará a cabo desde el **módulo de ajustes**, concretamente en la sección **'usuarios y compañías/Grupos'**.

Odoo ya instala una serie de **grupos predefinidos** al instalar los módulos. De esta forma, por ejemplo, al instalar el módulo de ventas, se creará el grupo 'Ventas / administrador', el cual tiene en su interior una serie de reglas y permisos que permitirán dotar al usuario que tenga este grupo asignado una serie de permisos de administración en dicho módulo.

Gestión de permisos de los usuarios

Los permisos podremos gestionarlos tanto en el **momento de crear** el usuario, como **más adelante**.

Para ello, en primer lugar, deberemos entrar en el usuario en concreto al que queramos gestionarle los permisos (desde el módulo de ajustes, en 'usuarios y compañías/Usuarios' o en 'administrar usuarios'). A continuación, en la sección **'Permisos de acceso'** tendremos las siguientes **secciones**:

Multicompañía:

Porque un usuario puede estar o no en **diferentes compañías**.

Seleccionaremos las compañías permitidas y la empresa predeterminada.

Tipo de Usuario: En los que diferenciaremos al:

- **'usuario interno'** (aquel que trabajará internamente con Odoo),

- 'Portal'
- y 'Publico'.

A estos **dos últimos** no se **nos permitirá gestionar permisos**, ya tienen unas reglas preestablecidas.

<Módulos>:

Aparecerán los módulos que tengamos **instalados** en Odoo y los diferentes **niveles de permisos** a configurar dentro de ellos. Así, por ejemplo, si tuviésemos el módulo de Fabricación instalado, se podría configurar los **privilegios** en dicho módulo como:

- 'administrador',
- 'usuario'
- o <blanco> (si no queremos que el usuario tenga acceso a este módulo).

Configuración técnica, permisos extra y otros (technical):

Finalmente se muestran una serie de opciones técnicas que podremos marcar/desmarcar para ofrecerle al usuario más o menos privilegios.

Al **guardar**, ya se nos habrá **informado los 'Grupos'** con los **permisos** en el usuario, los cuales podremos editar.

Creación de usuarios y gestión de permisos

En este ejemplo, veremos cómo crear un usuario y gestionar sus permisos. Siguiendo los pasos indicados en los apuntes, nos dirigimos al módulo de **ajustes**, y seleccionamos la **sección de usuarios** para administrarlos. También podemos acceder desde "Usuario y Compañía" -> "Usuarios". Una vez dentro, visualizamos los usuarios existentes. Actualmente, hay dos usuarios internos y uno de portal creados.

Para crear uno nuevo, hacemos clic en "Crear". Como ejemplo, asignaremos el nombre "test2" y una dirección de correo electrónico como "test2@test.com". Luego, en los permisos de acceso, si el usuario trabaja en múltiples compañías, las agregamos junto con la predeterminada.

Después, seleccionamos los módulos instalados y definimos los permisos para cada uno. Por ejemplo, en contabilidad, no asignaremos permisos por el momento. Procedemos a la **configuración técnica**, donde marcamos los permisos necesarios. Una vez guardado, el usuario se crea con los permisos predeterminados.

Se muestran los grupos asignados al usuario, junto con los 137 permisos correspondientes. Por ejemplo, dentro del grupo de "permisos extras", se encuentra "creación de contactos", con sus respectivos permisos:

- leer,

- escribir,
- crear,
- o eliminar).

Al visualizar los permisos del usuario "test2", se muestra cada permiso individualmente en lugar del grupo al que pertenecen. Por ejemplo, los permisos para "creación de contactos" incluyen leer, escribir, crear y eliminar.

Si editamos y otorgamos permisos adicionales, como administración de facturación y ventas, los grupos se incrementarán de 8 a 16 debido a los grupos asignados por defecto a estas secciones. Por ejemplo, al agregar permisos de administración en ventas, se incluye el grupo de "ventas administradoras" con permisos específicos.

Los **permisos pueden ser editados** en cualquier momento. Al hacer clic en un permiso, podemos modificarlo, como restringir la capacidad de leer o crear.

También podemos enviar una invitación por correo electrónico al usuario desde este punto.

Finalmente, desde aquí también es posible eliminar al usuario. Al hacerlo, el usuario "test2" ya no estará en la lista de usuarios.

Crear un informe con QWeb

QWeb es el **motor de plantillas** utilizado por Odoo. Es un motor de plantillas **XML** y se utiliza principalmente para generar **fragmentos y páginas HTML**. En el siguiente enlace podréis ver la documentación necesaria para **programar con Qweb**:

<https://www.odoo.com/documentation/13.0/reference/qweb.html#calling-sub-templates.>

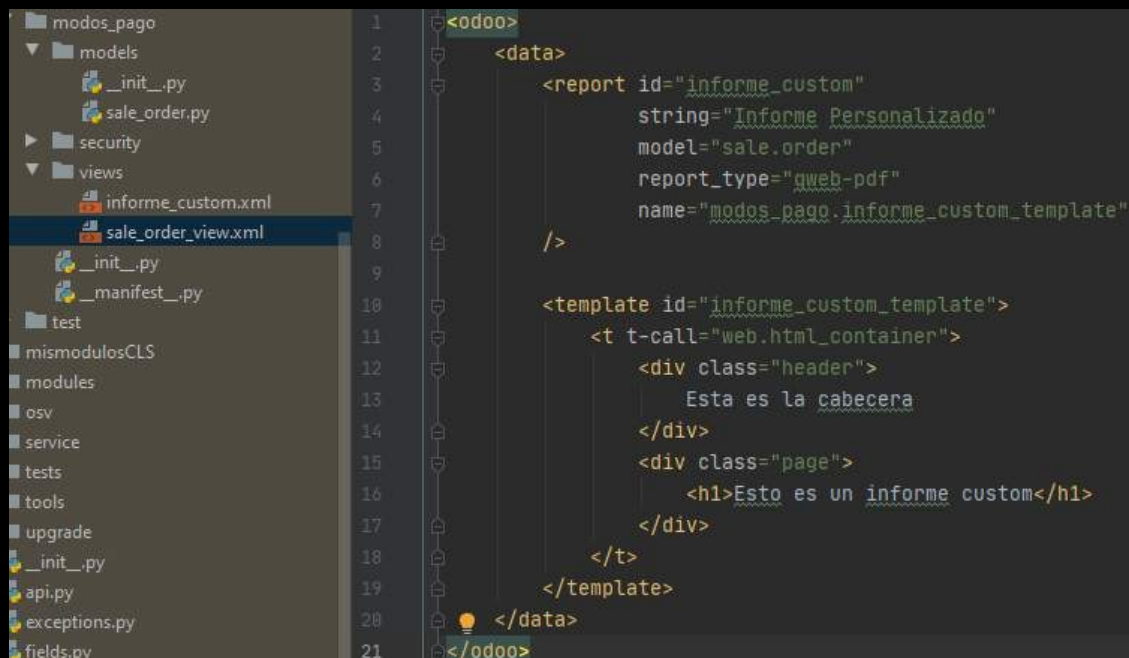
Para crear un informe, en primer lugar, identificaremos el **modelo** en el que queremos que aparezca nuestro informe.

En este caso vamos a crear un informe para el módulo de Ventas (sale.order), para que aparezca junto al informe estándar de 'presupuesto':

Aprovecharemos el módulo 'modos de pago' del tema anterior, puesto que **el informe QWeb no es más que una vista XML**. De esta forma nos ahorraremos tener que crear un módulo nuevo. Bastará con crear una nueva vista XML, que denominaremos 'informe_custom.xml', y agregar dicha vista al manifest en la sección data. Ahora informaremos el XML indicando en una primera etiqueta 'report', que se trata de un **tipo de reporte QWEB** y que pertenecerá al **modelo sale.order**.

A continuación, en una segunda etiqueta 'template' es donde crearemos el **contenido del informe**.

Implementación del XML con tipo de reporte Qweb



Un **informe Qweb**, por tanto, es un **archivo XML** que contiene una **etiqueta report_type = 'qweb-pdf'**, dentro de un elemento <report>, que está a su vez dentro de un elemento <data>.

Tras implementar los cambios actualizaremos el módulo y probaremos que el cambio ha surtido efecto.

Analizar informe QWeb desde Odoo

Vamos a ver cómo podemos analizar el código QWeb de un informe existente en Odoo. Tomaremos como ejemplo el formulario "*presupuesto / pedido*" de los pedidos de *venta* para analizar su código QWeb desde Odoo.

Para ello, seguiremos estos pasos:

1. Nos dirigimos al módulo **Ajustes**.
2. Dentro de "Ajustes", seleccionamos la opción "**Técnico**".
3. Una vez dentro, ingresamos en la sección de "**Acciones en Informes**".
4. En nuestro caso, buscamos el informe de "presupuesto / pedido".
5. Hacemos clic en él para abrirlo.
6. Dentro del informe, encontraremos un botón de "**Vista QWeb**".
7. Al hacer clic en este botón, se nos mostrará todo el **código QWeb** asociado a este informe.

Por ejemplo, la vista "report sale order" contiene una llamada en código QWeb al "report sale order document", y dentro de este, encontramos la **estructura del formulario QWeb en Odoo**.

También es posible buscar este formulario QWeb en una de las vistas del módulo **desde PyCharm**, donde encontraremos el **mismo código**.

Internal Server Error

Planteamiento: Al abrir en el navegador el `http://localhost:8069/` se produce un **error** de

'Internal Server Error'

y no entra en Odoo.

Nudo: ¿Qué puede estar pasando?

Puede deberse a varios problemas, por ejemplo:

- un **error de código**,
- o que el **servidor** se haya quedado **bloqueado**.

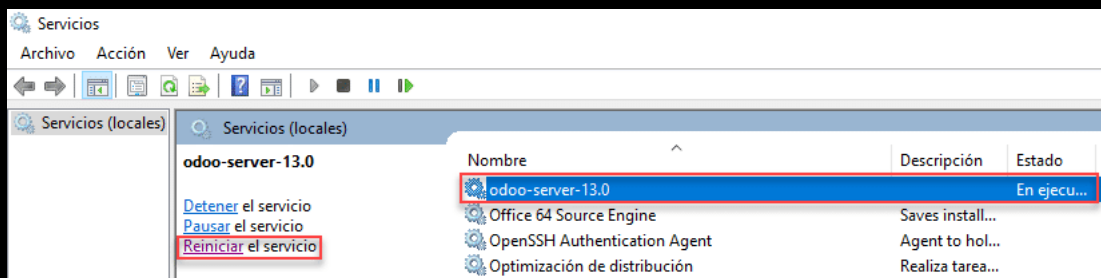
¿Cómo solucionarlo si el servidor se ha quedado bloqueado?

Desenlace: La solución pasa por **reiniciar el servicio**. Para ello, abrimos el '*administrador de tareas*' (`cntrl + alt + supr`) y en la pestaña '*servicios*' buscamos '*odoo-server-13.0*' (la versión que tengamos de Odoo).

Pulsamos sobre el mismo con el **botón derecho** y a continuación accedemos a '**abrir servicios**'.

Esto nos abrirá una ventana como la que se muestra en la figura, en la que buscaremos el '*odoo-server-13.0*' y lo **reiniciaremos**, aunque se esté ejecutando.

Reiniciando el servicio odoo-server-13.0



Si esto no funciona, puede que el error sea debido a:

- **problemas con el usuario**,
- **rol**,
- o **incidencias con el código** que tendremos que solucionar antes de reiniciar el servicio.

Esto lo indicaría el log de errores.

Errores comunes

A continuación, comentaremos 3 errores muy comunes que se dan cuando estamos desarrollando en Odoo y que a la vez son muy fáciles de solucionar:

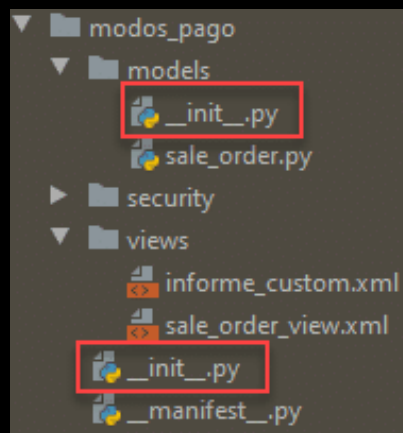
Olvidarnos de importar los modelos de Python: y por tanto **no se creen las tablas** en PostgreSQL.

De esta forma, tras crear los modelos e **instalar o actualizar** el módulo **no se vislumbrarían** los cambios.

Es importante recordar que, cada vez que se cree una **carpeta** y en su interior **haya modelos Python** se debe crear un **fichero `__init__.py`** para **importar** los modelos concretos.

- En el fichero `__init__.py` de la raíz **del módulo** se importará la **carpeta models** que contiene todos los modelos,
- pero en el `__init__.py` **de la carpeta models** se **importan los modelos** que queremos que realmente se carguen.

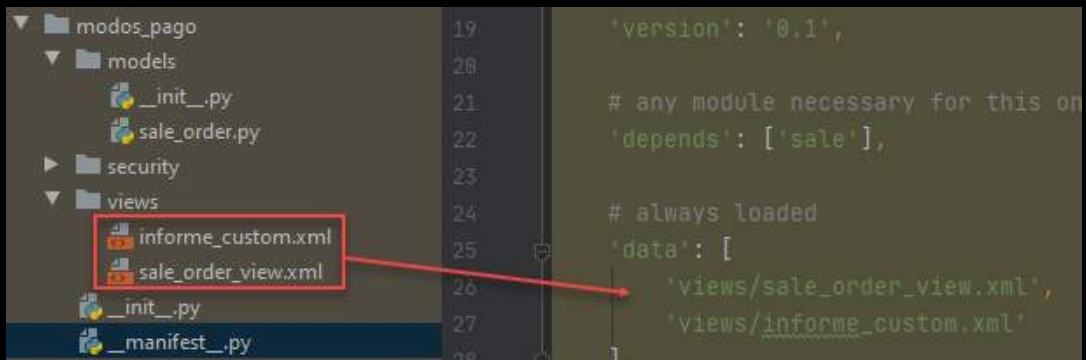
`__init__.py` en la raíz del módulo y en carpeta models



Olvidarnos de agregar o importar **en el manifest**, en la **sección 'data'**, la vista (o cualquier **fichero XML**) que hayamos creado **nueva**. Así, al instalar o actualizar el módulo **no se reflejan los cambios**.

Debemos tener en cuenta que, para que una vista se utilice en el módulo, ésta debe estar **informada en la sección 'data'** del manifest. (Interesante el comentario sobre la sección [data] del manifest: '# always loaded').

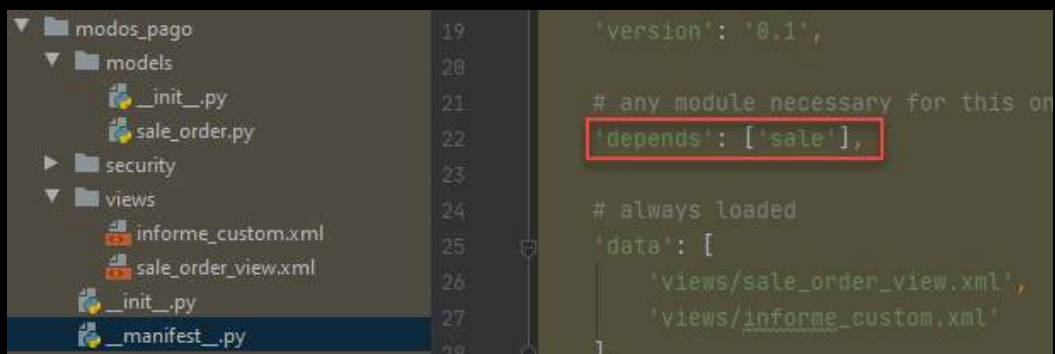
Informando en el manifest las vistas



Olvidarnos de indicar las **dependencias** en la sección 'depends' del **manifest** cuando estamos haciendo un módulo que hereda de otro.

Es necesario que indiquemos el/los módulo/s **necesario**/s para que el módulo actual **funcione correctamente**.

Informando en el manifest las dependencias



Los batch-input

Existen opciones para algunos sistemas ERP, que permiten **automatizar determinadas acciones** que necesitemos llevar a cabo en ellos.

Son los denominados batch-input, que permiten **ejecutar transacciones** o programas de forma automática, sin la necesidad de que el usuario interactúe con el programa en sí.

Por ejemplo, en SAP se pueden crear batch-inputs que simulan el proceso de informar campos en la pantalla, pulsar diferentes botones, radiobuttons, checkbox, etc. tal y como lo haría un usuario, solo que de manera automática. De esta forma, el usuario o el sistema ejecutaría el batch-input y éste se encargaría de ir **informando campos** y **pasando** por las diferentes **pantallas** de **forma automática**.

Los batch-inputs pueden ejecutarse en primer plano (mientras el usuario ve como se rellenan los campos y se van pasando por las pantallas), pero **lo normal** es que se ejecute **en segundo plano**.

Reflexión

Gracias a la gestión de permisos de los usuarios es posible **acotar en Odoo el acceso a los módulos** por parte de los trabajadores de una empresa. Para ello, lo ideal sería **crear un grupo para cada departamento** con los permisos que cada uno de ellos deba tener, y a continuación, asignar dicho grupo a todos los **usuarios del departamento**.

De esta forma, si entra un **empleado nuevo** en un departamento bastará con **agregarle el grupo** y ya tendrá los mismos **permisos** que los usuarios de su departamento. Cualquier **modificación sobre dicho grupo** afectará a **todos los empleados** del departamento.

Si un empleado perteneciese a varios departamentos, simplemente habría que agregarle los grupos de dichos departamentos para que tenga todos los permisos de esos grupos.

<https://www.odoo.com>

<https://praxyaformaplus.com/>

<https://odooerpcloud.com/>