

Sistemas de Gestión Empresarial

PERSONALIZACIÓN DE ODOO

Contenido

1. Introducción.....	3
1.1. Modelo de objetos en Odoo.....	3
2. Manejo de usuarios, grupos y permisos	3
3. Configuración de menús.....	8
4. Vistas en Odoo.....	11
5. Creación de un nuevo objeto en Odoo	17
5.1. Creación de una nueva entidad.....	18
5.2. Creación de una nueva vista	21
5.3. Creación de una nueva acción	22
5.4. Creación de un nuevo elemento de menú	22
5.5. Creación de relaciones	23
6. Tableros en Odoo.....	25
7. Informes en Odoo.....	26
7.1. Creación de informes	29
8. Herencia de vistas.....	32

1. Introducción

En este bloque de contenidos se verá una serie de puntos relativos a la personalización del ERP para su adaptación a una empresa. Se trata, en definitiva, de identificar los procesos más importantes de una empresa y ver de qué manera la aplicación puede llevarlos a cabo.

Hay una serie de módulos necesarios para poder llevar a cabo las tareas comunes de cualquier empresa, como ventas, compras, gestión de stock y almacén, facturación, etc. Pero, si bien es cierto que la aplicación puede funcionar sólo con el módulo base, en la mayoría de las ocasiones necesitaremos algún otro módulo para trabajar. Los módulos que necesitemos pueden estar precargados o ser necesario descargarlos de Internet y cargarlos en la aplicación para que se puedan instalar.

En este bloque vamos a ver como personalizar Odoo para que se adapte lo máximo posible a la empresa en la que se quiera implantar. Veremos que, sin necesidad de tocar código, podemos cambiar el ERP de muchas maneras para que éste se amolde lo máximo posible al caso que tengamos entre manos.

1.1. Modelo de objetos en Odoo

La base de datos de un sistema ERP es de gran envergadura. Almacena las tablas con los datos de la aplicación, vistas de las diferentes tablas y otros elementos como funciones o disparadores que realizan operaciones sobre los datos. Debido a esta gran cantidad de información almacenada, se hace necesario una organización entre sus componentes.

Lo que se hace es establecer una serie de normativas o nomenclatura para organizar la información que los desarrolladores deben seguir a la hora de modificar el código fuente o el esquema de la base de datos. Por ejemplo, incluir un prefijo en los componentes de la base de datos para saber a qué módulo pertenecen, o establecer una serie de campos dentro de una tabla como obligatorios, para poder asegurar el funcionamiento correcto de la aplicación.

En los sistemas de planificación empresarial desarrollados en un lenguaje orientado a objetos, cualquier dato es accesible a través de objetos. Por ejemplo, en Odoo tenemos un objeto `res.partner` para acceder a los datos concernientes a los contactos, colaboradores o socios, teniendo en cuenta que cada uno de los objetos es en realidad una tabla de la base de datos. Como se puede ver, va precedido de un prefijo y se indica el módulo al que pertenece.

Otro aspecto destacable es que los módulos instalados en el sistema influyen en la base de datos que vamos a tener. Esto quiere decir que la instalación de nuevos módulos puede conllevar (y normalmente conlleva), la modificación de la estructura de datos de Odoo.

En los siguientes apartados se verá cómo editar estos objetos y sus vistas correspondientes.

2. Manejo de usuarios, grupos y permisos

Se va a ver en este apartado cómo trabajar con usuarios, grupos y permisos. Vamos a partir de una base de datos recién creada que tiene instalados únicamente los módulos de inventario,

compra y ventas.

Si ahora vamos a usuarios, se ve que existe un usuario administrador, el cual se crea automáticamente al crear la base de datos, que tiene unos permisos sobre las aplicaciones.

The screenshot shows the Odoo User Management interface. At the top, it displays the user's name, 'Nombre: Administrator', and email address, 'Dirección de correo electrónico: prueba@prueba.es'. Below this, there are tabs for 'Permisos de acceso', 'Preferencias', and 'Seguridad de la cuenta'. Under the 'Permisos de acceso' tab, the user has access to several modules:

- SALES:** Ventas (Administrador)
- INVENTORY:** Inventario (Administrador), Compra (Administrador)
- ACCOUNTING:** Contabilidad (Administrador de Facturación)
- ADMINISTRATION:** Administración (Ajustes)

Si editamos este usuario, se ve que cada uno de estos permisos se puede modificar. Para probarlo, se puede crear un nuevo usuario, y darle únicamente permisos de usuario en inventario, según se ve en la imagen.

The screenshot shows the Odoo User Management interface for a new user named 'Juan'. The user's name is 'Juan' and their email address is 'juan@prueba.es'. The 'Permisos de acceso' tab is selected, showing the following permissions:

- SALES:** Ventas (User)
- INVENTORY:** Inventario (User), Compra (User)
- ACCOUNTING:** Contabilidad (User)
- ADMINISTRATION:** Administración (User)

Si ahora nos conectamos con ese usuario, se puede ver que únicamente tiene acceso al inventario, a pesar de estar instalados los módulos de ventas y compras. Puede ser buena idea utilizar otro navegador para conectarse con otros usuarios, así no hay que estar cerrando y

abriendo sesión. No sirve con otras pestañas dentro del mismo navegador, debe ser otro navegador. Al entrar en el inventario, el nuevo usuario tiene muy poco acceso sobre los productos, por ejemplo, no puede crear nuevos productos.

Para ver qué significan los permisos de usuario en inventario, se debe activar el *modo desarrollador*. Con el modo desarrollador activado, si vamos desde Ajustes a la pestaña Usuarios y compañías, vemos que aparece una nueva opción, que es **Grupos**. Al seleccionar esta opción, se pueden ver los grupos y qué puede hacer cada uno. Por ejemplo, si se selecciona el grupo *Inventario / Usuario*, se ven todas sus opciones, indicando en la pestaña *Permisos de acceso* las acciones que puede llevar a cabo un usuario en el módulo Inventario. Como antes se indicó, resulta más cómodo modificar los permisos en el llamado ‘navegador 1’, y luego, en el ‘navegador 2’, pulsar F5 para actualizar y ver el comportamiento del nuevo usuario, sin necesidad de entrar y salir.

Nombre	Modelo	Permiso de lectura	Permiso de escritura	Permiso de creación	Permiso de eliminación
stock.scrap.user	Desechar	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
stock.lot user	Lote/serie	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
purchase.order	Pedido de compra	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
sale.order stock worker	Pedido de venta	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
product.template stock user	Producto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
stock.picking user	Traslado	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
product_product_stock_user	Variante de producto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
access.stock.warn.insufficient.qty.scrap	Advertir de cantidad de desecho insuficiente	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
access.stock.return.picking	Albarán de devolución	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
access.stock.assign.serial	Asignar números de serie a stock	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
access.stock.change.product.qty	Cambiar cantidad de producto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Aquí es donde se puede indicar que un usuario puede tener acceso a la Plantilla de producto para crear un nuevo producto, eliminarlo, etc.

Aplicación ? Inventario

Nombre ? Usuario

Compartir grupo ?

Usuarios Heredado Menús Vistas Permisos de acceso Reglas de registro Notas

1-40 / 42 < >

Nombre	Modelo	Permiso de lectura	Permiso de escritura	Permiso de creación	Permiso de eliminación	
stock.scrap.user	Desechar	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	
stock.lot user	Lote/serie	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
purchase.order	Pedido de compra	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
sale.order stock worker	Pedido de venta	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
product.template stock user	Producto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
stock.picking user	Traslado	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
product_product_stock_user	Variante de producto	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
access.stock.warn.insufficient.qty.scrap	Advertir de cantidad de desecho insuficiente	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	

Supongamos ahora que no queremos modificar los permisos de ‘Usuario’, eso afectaría a todos los usuarios, sino que queremos crear un nuevo rol, que sea ‘**Usuario avanzado**’, que sí tenga permisos para escribir, crear y eliminar plantillas de productos. Para ello, desde la opción Grupos se debe crear un nuevo grupo, de nombre **UsuarioAvanzado**, que sea de la aplicación *Inventario*, que herede los permisos del *usuario normal* para esa aplicación *Inventario* y que tenga un permiso extra sobre el objeto *Plantilla de producto (product.template)* para poder escribir, crear y eliminar nuevos productos, y dándole de nombre al nuevo permiso, por ejemplo **product_custom**.

Ajustes Opciones generales Usuarios y compañías Traducciones Técnico

Grupos / Nuevo

Aplicación ? Inventario

Nombre ? UsuarioAvanzado

Compartir grupo ?

Usuarios **Heredado** Menús Vistas Permisos de acceso Reglas de registro Notas

Los usuarios agregados a este grupo se añaden automáticamente a los siguientes grupos. ?

Nombre del grupo

Inventario / Usuario

Agregar una línea

The screenshot shows the Odoo application settings interface. A new group is being created with the following details:

- Aplicación:** Inventario
- Nombre:** UsuarioAvanzado
- Compartir grupo:**
- Permisos de acceso:** This tab is selected. It shows the following table:

Nombre	Modelo	Permiso de lectura	Permiso de escritura	Permiso de creación	Permiso de eliminación
Product_custom	Producto	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Añadir una línea					

Ahora, si agregamos al usuario creado al principio al grupo *UsuarioAvanzado*, al actualizar el navegador 2, ya se puede ver que el usuario tiene permisos para crear, modificar y eliminar productos.

The screenshot shows the Odoo user profile interface for the user 'Juan'. The user's profile includes the following information:

- Aplicación:** Inventario
- Nombre:** UsuarioAvanzado
- Compartir grupo:**
- Usuarios:** This tab is selected. It shows the following table:

Nombre	Inicio de sesión	Idioma
Juan	juan@prueba.es	Spanish / Español
Añadir una línea		

The screenshot shows the Odoo inventory products page. The top navigation bar includes tabs for **Inventario**, **Información general**, **Operaciones**, and **Productos**. The main area displays a product card for 'Producto 1' with the following details:

- NUEVO** button
- Filtros**, **Agrupar por**, and **Favoritos** buttons
- Productos** search bar
- 1-1 / 1** pagination
- Producto 1** card with icons, a star icon, and the following text:
Precio: 1,00 €
Stock real: 5,00 Unidades

3. Configuración de menús

En este apartado vamos a ver cómo trabajar con los menús. Todo lo referente a los menús se encuentra en la pestaña **Técnico**, desde Ajustes, en la opción *Elementos de menú* que está dentro del bloque *Interfaz de Usuario* (lógicamente con el modo de desarrollador previamente activado).

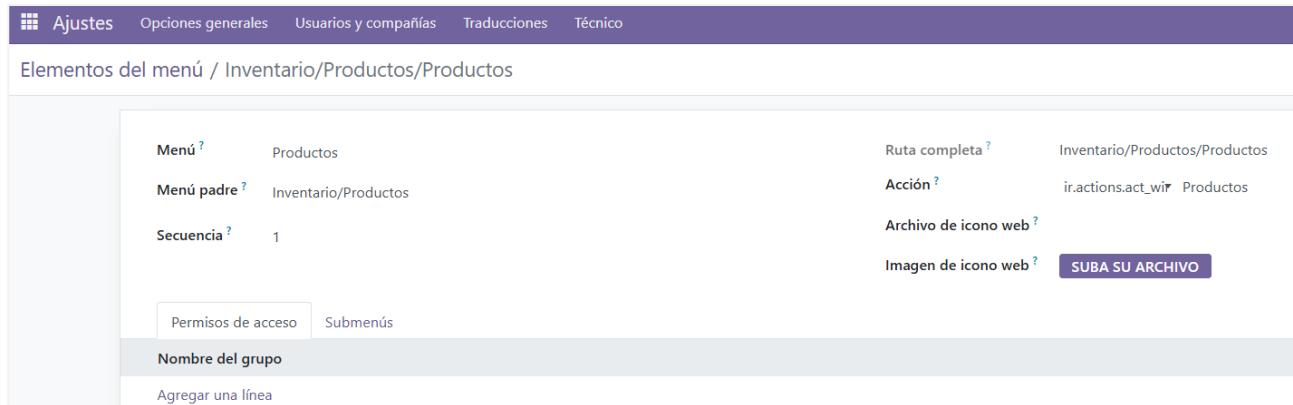
Todo lo que se ve aquí son las opciones de menú, **únicamente para las aplicaciones instaladas** (solo está instalado *Inventario, compras y ventas*), de igual forma que en el apartado anterior para los grupos. Cuantas más aplicaciones haya instaladas, más opciones de menú aparecen aquí.

Menú
Menú
Compra/Pedidos/Solicitudes de presupuesto
Ajustes/Usuarios y compañías/Usuarios
Ajustes/Opciones generales
Facturación / Contabilidad/Configuración/Ajustes
Tableros/Tableros
Inventario/Configuración/Ajustes
Inventario/Información general
Compra/Configuración/Ajustes
Compra/Pedidos
Ajustes/Técnico/Conversaciones
Facturación / Contabilidad/Informes/Informes de extractos bancarios
Ventas/Configuración/Pedidos de venta/Plantillas de presupuesto

Todas las opciones de menú que aparecen aquí tienen una jerarquía en modo árbol. Si seleccionamos uno, por ejemplo, *Ajustes/Usuarios y compañías/Usuarios* nos muestra una pantalla informativa sobre el menú, indicando entre otras características, la acción que realiza (pinchar sobre *Usuarios*). Es una acción de ventana (`actions.act_window`) que nos lleva a una vista, que será la de los usuarios, en este caso.

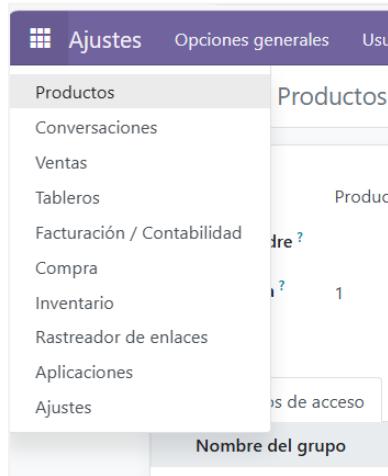
Para probar la configuración de menús, vamos a añadir una opción de menú que sea productos, para poder editar el inventario directamente. Para ello, se va a duplicar la opción *Productos*, que

hay en Productos, de la aplicación Inventario. Buscamos esa vista en los elementos de menú y accedemos a ella.



Desde aquí, lo que se va a hacer es **duplicarla**, que es una de las dos opciones que se pueden hacer si se pincha en el combo Acción. Se dejan todas las opciones que pone, a excepción del padre, que se le quita y se deja vacío, ya que queremos que sea un elemento del menú raíz.

Si ahora se refresca la pantalla (F5) y se accede al ícono de los accesos directos, ya se verá la opción **Productos** ahí.



Ahora bien, cabe preguntarse por qué aparece esa opción ahí. Y la respuesta es que, en Odoo, si un elemento de menú no está asignado a ningún grupo, **por defecto aparece para todos los grupos**. De hecho, si en otro navegador accedemos con el usuario creado en el punto 2, también tendrá el acceso directo a la opción de menú que acabamos de duplicar.

Vamos a hacer ahora un pequeño cambio para que ese elemento de menú (Productos) solo lo puedan ver los usuarios que pertenezcan al grupo **UsuarioAvanzado** creado antes. Para ello, se edita el grupo en cuestión, y en la pestaña Menús buscamos el que acabamos de crear, que sería *Productos a secas*, y se lo asignamos a este grupo.

The screenshot shows the 'Grupos' configuration page for the 'UsuarioAvanzado' group. It displays fields for 'Aplicación' (Inventory), 'Nombre' (UsuarioAvanzado), and 'Compartir grupo' (unchecked). Below these, there are tabs for 'Usuarios', 'Heredado', 'Menús', 'Vistas', 'Permisos de acceso', 'Reglas de registro', and 'Notas'. The 'Menús' tab is active, showing a list of menus. The 'Productos' menu item is highlighted with a red oval.

A partir de ahora, únicamente los usuarios que pertenezcan a ese grupo pueden ver la opción Productos de forma directa. Ni siquiera el usuario administrador, ya que no está en el grupo UsuarioAvanzado, tiene acceso directo a esa opción de menú.

De esta forma ya se puede configurar el menú y todas sus opciones para dejarlo según el gusto personal o de la empresa en cuestión.

4. Vistas en Odoo

Dado que todos los datos del programa están almacenados en objetos, ¿cómo se muestran dichos objetos al usuario? Tanto la información de las tablas como la de cualquier otro objeto de la aplicación o módulo se muestra a través de vistas.

Cada objeto tiene su propia interfaz, por ejemplo, no se muestran de la misma manera los datos de las Empresas que los datos de una Factura. Las interfaces pueden ser:

- **Estáticas:** se crean dentro del código de la aplicación y no pueden ser modificadas.
- **Dinámicas:** pueden ser modificadas por parte del usuario, para lo cual se almacena la descripción de la vista en un lenguaje de descripción de datos que permita su modificación, como por ejemplo XML.

Por tanto, las interfaces dinámicas son construidas de forma dinámica por la descripción XML de la pantalla del cliente. Para ello no es necesario ser unos expertos en ese lenguaje, podemos hacer objetos sencillos simplemente tomando como ejemplo otros objetos que haya creados en la aplicación. No obstante, muchas aplicaciones proveen la forma de crear las descripciones de manera gráfica sin necesidad de introducir código manual.

Es importante destacar que no todas las vistas están asociadas a cada objeto. También hay que señalar que existen una gran cantidad de vistas en el ERP ya creadas. Por ejemplo, si se crea un cliente, se puede ver con tres tipos de vistas:

1. Vista tipo Kanban

Nombre	Teléfono	Correo electrónico	Comercial	Actividades	Ciudad	País
Support Inet4U		info@inetfouru.com				
Support Inet4U, Carlos Cuadrado		carlos@inetfouru.com				

2. Vista tipo lista (list) *En las vistas se suele denominar tipo lista pero en las plantillas (modo de vista) se identifica como tipo árbol (tree)*

Nombre	Teléfono	Correo electrónico	Comercial	Actividades	Ciudad	País
Support Inet4U		info@inetfouru.com				
Support Inet4U, Carlos Cuadrado		carlos@inetfouru.com				

3. Vista tipo formulario (form)

The screenshot shows a customer profile in Odoo. At the top, there are navigation tabs: Ventas, Pedidos, A facturar, Productos, Informes, and Configuración. Below the tabs, it says 'Clientes / Support Inet4U, Carlos Cuadrado'. On the right, there are summary statistics: \$ 0 Ventas, 0 Compras, Todavía No H..., 0,00 € Facturado. The main area displays contact information for 'Carlos Cuadrado' (Individual). Fields include: Puesto de trabajo (Director ventas España), Teléfono (6), Móvil (6), Correo electrónico (carlos@inetfuru.com), Sitio web (p. ej. https://www.odoo.com), Título (p. ej. Señor), and Etiquetas (p. ej. "B2B", "VIP", "consultoría", ...). Below the form, there are tabs: Contactos y direcciones, Vistas y compras, Facturación / Contabilidad, and Notas internas.

Aunque hay más tipos de vistas (calendario, Gantt, etc.), estas tres son las que más vamos a usar. Todas las vistas que hay en el ERP se pueden ver de forma similar a los elementos de menú, es decir, desde la pestaña *Técnico*, *interfaz de usuario*, opción *Vistas*.

The screenshot shows the 'Vistas' (Views) list in the Odoo Technical interface. The left pane lists views categorized by type: Formulario, Buscar, Kanban, Árbol, and Reporte. The right pane shows a detailed view of a specific view record, with fields for Nombre de la vista (res.config.settings.view.form.inherit.base.setup), Tipo de vista (Formulario), and Modo (res.config.settings.view.form.inherit.base.setup). A large dropdown menu on the right lists various view types and their corresponding model names, such as 'Telefonos / SMS', 'SMS', 'Plantillas de SMS', 'Lista negra del teléfono', 'Acciones', 'Informes', 'Acciones de ventana', 'Acciones de servidor', 'Asistentes de configuración', 'Valores por defecto del usuario', 'IAP', 'Cuentas IAP', 'Interfaz de usuario', 'Elementos del menú', 'Vistas', 'Vistas personalizadas', 'Filtros de usuario', and 'Recorridos'. The dropdown also includes a search bar at the top.

Aquí se ve que existen más de 1000 vistas para las aplicaciones que tenemos instaladas. Cada una de ellas corresponde a una pantalla, con la complejidad añadida de que una vista puede heredar de otras. Para identificar las diferentes vistas, hay que saber que cada vista tiene un nombre y actúa sobre un objeto del sistema.

Para ver el contenido de la vista, se debe tener activado el modo desarrollador. Entonces, desde la vista que se quiera analizar se va a la opción **Editar vista**, según muestra la imagen.

The screenshot shows the Odoo application interface in developer mode. The top navigation bar includes 'Inventario', 'Información general', 'Operaciones', 'Productos', 'Informes', and 'Configuración'. On the far right, it shows the user 'Administrator (personalización)' with a gear icon. Below the navigation is a toolbar with icons for search, refresh, and other functions. The main area is titled 'Productos' and contains a card for 'Producto 1' with fields: Precio: 1,00 € and Stock real: 5,00 Unidades. A purple 'NUEVO' button is visible. To the right of the card is a context menu with several options:

- Ejecutar pruebas JS
- Ejecutar pruebas JS Mobile
- Ejecutar prueba de clic en todos lados
- Abrir vista
- Desactivar recorridos
- Comenzar recorrido
- Editar Acción
- Ver campos
- Gestionar filtros
- Ver permisos de acceso
- Ver reglas de registro
- Obtener vista
- Editar vista:Kanban** (This option is highlighted with a red oval)
- Editar SearchView
- Activar depuración de activos
- Activar pruebas para depuración de activos

At the bottom right of the interface, there is a page number '1-1 / 1' and some navigation icons.

En resumen, aquí se ven las opciones de la vista (nombre, modelo, si hereda o no de otra, etc.) y su contenido, que no es más que un **código XML** que representa todo lo que se está viendo en la pantalla en cuestión. Este código se puede editar modificando el contenido de la vista que se mostrará. Por ejemplo, supongamos que en la **vista kanban** de productos, no se quiere mostrar la imagen.

Para hacer este cambio, se debe buscar en el código la parte que corresponde a la imagen del producto. Una vez localizada se puede borrar o hacer invisible. El problema que podría tener borrar esa parte de código es que, si otra vista depende ella, se modificará y podría no interesarnos. Para ello, lo mejor es hacer que sea **invisible**. Si una vez cambiado se actualiza (F5) se verá que ya no aparece el ícono de imagen en la vista kanban de productos.

Editar vista:Kanban

Nombre de la vista?	Product.template.product.kanban	Campo hijo?	
Tipo de vista?	Kanban	Vista heredada?	
Modelo?	product.template	Modo de herencia de la vista?	Vista base
Secuencia?	16	Datos del modelo?	Product.template.product.kanban
Activo?	<input checked="" type="checkbox"/>	ID externo?	product.product_template_kanban_view

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 estándar a través de vistas heredadas o mediante la personalización en Odoo Studio.

- [Arquitectura](#)
- [Permisos de acceso](#)
- [Vistas heredadas](#)

```

1 <kanban sample="1" class="o_kanban_product_template">
2   <field name="id"/>
3   <field name="product_variant_count"/>
4   <field name="currency_id"/>
5   <field name="activity_state"/>
6   <progressbar field="activity_state" colors="{'planned': '#cccccc', 'success': '#99ff99', 'today': '#ffff99', 'overdue': '#ff9999'}"/>
7   <t t-name="kanban-box">
8     <div class="oe_kanban_card oe_kanban_global_click">
9       <div class="o_kanban_image me-1" invisible="1">
10      
11    </div>
12    <div class="oe_kanban_details">
13      <div class="o_kanban_record_top mb-0">
14        ...

```

GUARDAR **DESCARTAR**

- [Inventario](#)
- [Información general](#)
- [Operaciones](#)
- [Productos](#)
- [Informes](#)
- [Configuración](#)

Productos

NUEVO

Producto 1	
Precio: 1,00 €	
Stock real: 5,00 Unidades	

Lógicamente también es posible añadir nuevos campos a una vista.

Podemos editar la **vista de lista** correspondiente a los productos. Es también una vista sencilla, es decir, no hereda de ninguna otra, y según se puede ver en el código XML, cada uno de los campos hace referencia a una columna del objeto producto. Si se pretende, por ejemplo, que no aparezca el campo nombre en la vista, no hay más que comentar o eliminar ese campo y actualizar. En la lista de campos ya no se reflejará el nombre.

Editar vista:List

Nombre de la vista ?	product.template.product.tree	Campo hijo ?	
Tipo de vista ?	Árbol	Vista heredada ?	
Modelo ?	product.template	Modo de herencia de la vista ?	Vista base
Secuencia ?	16	Datos del modelo ?	product.template.product.tree
Activo ?	<input checked="" type="checkbox"/>	ID externo ?	product.product_template_tree_view

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 estándar a través de vistas heredadas o mediante la personalización en Odoo Studio.

- Arquitectura
- Permisos de acceso
- Vistas heredadas

```

1 <tree string="Product" multi_edit="1" sample="1">
2   <header>
3     <button string="Print Labels" type="object" name="action_open_label_layout"/>
4   </header>
5     <field name="product_variant_count" invisible="1"/>
6     <field name="sale_ok" invisible="1"/>
7     <field name="currency_id" invisible="1"/>
8     <field name="cost_currency_id" invisible="1"/>
9     <field name="priority" widget="priority" optional="show" nolabel="1"/>
10    <!--<field name="name" string="Product Name"/>-->
11    <field name="default_code" optional="show"/>
12    <field name="product_tag_ids" widget="many2many_tags" options="{'color_field': 'color'}" optional="show"/>
13    <field name="barcode" optional="hide" attrs="{'readonly': [('product_variant_count', '>', 1)]}">
14    <field name="company_id" options="{'no_create': True}" groups="base.group_multi_company" optional="hide"/>
15    <field name="list_price" string="Sales Price" widget="monetary" options="{'currency_field': 'currency_id'}" optional="show"/>

```

GUARDAR **DESCARTAR**

Como vista más compleja, se puede ver la **vista tipo formulario** de un producto. Esta vista no se compone de un solo XML, sino de varios, ya que el formulario va paginado con pestañas, tiene diferentes opciones, etc, y como se puede ver ya no es una vista sencilla, sino que hereda de otra, en concreto hereda de la vista *product.template.common.form*. Si se va a esta vista padre, se puede ver que el código correspondiente a las diferentes páginas de producto (Información General, Ventas, Compra, Inventario) está en esta vista, y no en la vista de formulario *product.template.product.form*.

Editar vista:Form

Nombre de la vista ?	product.template.product.form	Campo hijo ?	
Tipo de vista ?	Formulario	Vista heredada ?	product.template.common.form
Modelo ?	product.template	Modo de herencia de la vista ?	Vista base
Secuencia ?	8	Datos del modelo ?	product.template.product.form
Activo ?	<input checked="" type="checkbox"/>	ID externo ?	product.product_template_only_form_view

Abrir: Vista heredada

```

74 <field name="cost_currency_id" invisible="1"/>
75 <field name="product_variant_id" invisible="1"/>
76 </group>
77 </group>
78 <group string="Internal Notes">
79 <field colspan="2" name="description" nolabel="1" placeholder="This note is only for internal pu
80 </group>
81 </page>
82 <page string="Sales" attrs="{'invisible':[(('sale_ok','=',False))]}" name="sales" invisible="1">
83 <group name="sale">
84 <group string="Upsell & Cross-Sell" name="upsell" invisible="1"/>
85 </group>
86 <group>
87 <group string="Sales Description" name="description">
88 <field colspan="2" name="description_sale" nolabel="1" placeholder="This note is added to sa
89 </group>
90 </group>
91 </page>
92 <page string="Purchase" name="purchase" attrs="{'invisible': [(('purchase_ok','=',False))]}" invisible="1">
93 <group name="purchase">
94 <group string="Vendor Bills" name="bill"/>
95 </group>
96 </page>
97 <page string="Inventory" name="inventory" groups="product.group_stock_packaging" attrs="{'invisible':[((
98 <group name="inventory">
99 <group name="group_lots_and_weight" string="Logistics" attrs="{'invisible': [('type', 'not in',
100 <label for="weight" attrs="{'invisible':[(('product_variant_count', '>', 1), ('is_product_
101 <div class="o_row" name="weight" attrs="{'invisible':[(('product_variant_count', '>', 1),
102 <field name="weight" class="oe_inline"/>
103 <field name="weight_uom_name"/>
104 </div>
105 <label for="volume" attrs="{'invisible':[(('product_variant_count', '>', 1), ('is_product_
106 <div class="o_row" name="volume" attrs="{'invisible':[(('product_variant_count', '>', 1),
107 <field name="volume" string="Volume" class="oe_inline"/>
108 <field name="volume_uom_name"/>
109 </div>
110 </group>
111 <group name="packaging" string="Packaging" colspan="4" attrs="{'invisible':['|', ('type', 'not in',
112 <field colspan="2" name="packaging_ids" nolabel="1" context="{'tree_view_ref':product.product_p
113

```

GUARDAR Y CERRAR **DESCARTAR**

Se puede comprobar también que el objeto **product.template** tiene tres vistas asociadas, una tipo formulario, una tipo árbol y una vista tipo kanban. A su vez, para poder seguir la jerarquía de vistas y sus respectivas herencias, se puede seleccionar la opción **Vistas heredadas** para ver las vistas que heredan de una vista concreta.

Ajustes					Opciones generales	Usuarios y compañías	Traducciones	Técnico	★	✖	✖	✖
Vistas												
	NUEVO								Activo	Vista Product.template.product	Buscar...	
									Filtros	Agrupar por	Favoritos	
Nombre de la vista	Tipo de vista	Modelo	ID externo	Vista heredada								
product.template.product.form	Formulario	product.template	product.product_template_only_form_view	product.template.common.form								
product.template.product.kanban	Kanban	product.template	product.product_template_kanban_view									
product.template.product.tree	Árbol	product.template	product.product_template_tree_view									

De esta manera, modificando las vistas, se puede personalizar cómo se muestra casi todo en el ERP, es decir, a partir de los campos que tenemos en base de datos, cómo se muestran y se vuelcan en pantalla.

Por último, y para comprobar que todo está relacionado, si vamos de nuevo a *Elementos de menú*, buscamos el elemento de menú *Productos* que habíamos creado, que nos llevaba a la vista

de productos. Si se edita y se va al enlace externo de la acción, se ve que ahí tenemos los tres tipos de vistas asociados a productos (kanban, tree, form).

Ajustes Opciones generales Usuarios y compañías Traducciones Técnico

Elementos del menú / Productos

Menú ? Productos
Menú padre ?
Secuencia ? 1
Ruta completa ? /product/product
Acción ? ir.actions.act_window Productos
Archivo de ícono web ?
Imagen de ícono web ? SUBA SU ARCHIVO

Permisos de acceso Submenús
Nombre del grupo
Inventario / UsuarioAvanzado
Añadir una línea

Enlace interno

Ajustes Opciones generales Usuarios y compañías Traducciones Técnico

Elementos del menú / Productos / Productos

Nombre de la acción ? Productos
ID externo ? stock.product_template_action_product
Objeto ? product.template
Uso de la acción ?
Tipo de acción ? ir.actions.act_window
Ventana destino ? Ventana actual

Opciones generales Seguridad

VISTAS FILTROS
Modo de vista ? kanban,tree,form
Ref. de la vista ?
Ref. de vista de búsqueda ? product.template.search.stock.form
Valor del dominio ?
Valor de contexto ? {"search_default_consumable": 1, 'default_detailed_'}
Límite ? 80
Filtro ?

5. Creación de un nuevo objeto en Odoo

Se va a ver ahora cómo trabajar con las vistas, pero creando nuestras propias entidades, modificando la base de datos y creando vistas para esas nuevas entidades, es decir, no modificar las existentes sino tener unas nuevas.

Vamos a suponer que queremos dar de alta una entidad coche, queremos crear unas vistas para los coches y unos elementos de menú que nos permitan acceder a las vistas de los coches. Para hacer algo así desde Odoo, lo primero que se debe saber es que no es necesario tocar nada de la base de datos, **se puede configurar desde la propia aplicación**.

Como siempre para este tipo de acciones, lo primero sería activar el modo desarrollador. Una vez activado, vamos de nuevo a *Ajustes, Técnico*, bloque *Estructura de la base de datos*. En *modelos* están todas las tablas de la base de datos, con sus campos. Por ejemplo, si se busca la tabla **product.template**, con la que se trabajó en apartados anteriores, y se abre, se puede ver que aparecen todos sus campos, con el tipo de campo correspondiente. Aquí, los tipos que indican *many2one* y *one2many* se refieren a tipos de relaciones en una base de datos relacional, que es lo que se maneja en Odoo.

The screenshot shows the Odoo Technical Settings (Ajustes, Técnico) interface, specifically the Modelos / Producto page. The top navigation bar includes links for Ajustes, Opciones generales, Usuarios y compañías, Traducciones, Técnico, and Administra. Below the navigation is a toolbar with icons for Imprimir, Acción, and navigation arrows. The main content area has tabs for Campos, Permisos de acceso, Reglas de registro, Notas, and Vistas. The Campos tab is selected, showing a detailed list of fields for the product.template model. Each field is listed with its name, label, type, required status, and other properties like 'Sólo lectura' and 'Indexado'. The 'Tipo' column includes Odoo object names like account, product, purchase, and sale, followed by specific field names such as _last_update, account_tag_ids, active, activity_date_deadline, activity_exception_decoration, activity_exception_icon, and activity_ids.

Nombre de campo	Etiqueta de campo	Tipo de campo	Requerido	Sólo lectura	Indexado	Tipo
_last_update	Última modificación el	fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
account_tag_ids	Etiquetas de cuenta	many2many	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
active	Activo	booleano	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
activity_date_deadline	Fecha límite de la siguiente actividad	fecha	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
activity_exception_decoration	Decoración de Actividad de Excepción	selección	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
activity_exception_icon	Icono	carácter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
activity_ids	Actividades	one2many	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base

5.1. Creación de una nueva entidad

Desde esta pantalla de los modelos se puede crear una nueva **entidad**, en este caso crearemos la entidad **Coches**. Se le pone el prefijo **x_** en el modelo para que, según la nomenclatura de tablas de la base de datos, se identifique que es una entidad creada por nosotros.

The screenshot shows the Odoo 'Modelos / Nuevo' (Models / New) screen. At the top, there are buttons for Imprimir (Print), Acción (Action), and Nuevo (New). The main area contains fields for 'Descripción del modelo' (Model description) set to 'Coche', 'Tipo' (Type) set to 'Objeto personalizado' (Custom Object), and 'Modelo' (Model) set to 'x_coche'. Below these are checkboxes for various properties like 'Pedido' (Order), 'Modelo transitorio' (Transient Model), and several related to emails. A table below lists a single field 'x_name' with type 'carácter' (character), marked as required. At the bottom, tabs for Campos (Fields), Permisos de acceso (Access Rights), Reglas de registro (Registration Rules), Notas (Notes), and Vistas (Views) are visible.

Y se le añaden tres campos:

- Campo modelo (**x_modelo**), de tipo carácter (char). La etiqueta es para que aparezca en las vistas ya directamente. Por último, el modelo se pone como requerido.

The screenshot shows the 'Crear Campos' (Create Fields) interface. It starts with a basic field configuration for 'Nombre de campo' (Name) 'x_modelo' and 'Etiqueta de campo' (Label) 'Modelo', both of type 'carácter' (character). Below this are sections for 'PROPIEDADES BASE' (Basic Properties) where 'Requerido' (Required) is checked, and 'PROPIEDADES AVANZADAS' (Advanced Properties) which is currently empty. At the bottom, buttons for GUARDAR Y CERRAR (Save and Close), GUARDAR Y NUEVO (Save and New), and DESCARTAR (Discard) are available.

- Un segundo campo de nombre **x_fechaalta** (cuándo se mete el coche en el sistema), de tipo fecha, etiqueta Fecha Alta y también requerido.
- Un tercer campo de tipo entero (integer), nombre **x_pasajeros**, etiqueta Pasajeros y que sea opcional (no requerido).

Los nombres de estos campos son los que se van a usar en las vistas. La idea es crear vistas personalizadas del nuevo objeto, para poder decidir qué se muestra en cada una y cómo se muestra, ya que, si pinchamos en la opción *Crear un menú*, Odoo crea vistas por defecto con las opciones que él considera. En cuanto se guarda el nuevo objeto, se ve que aparecen los campos creados por nosotros y una serie de campos internos, que no se deben tocar.

Modelos / Coche				Imprimir	Acción								
	Descripción del modelo ?	Coche	Tipo ?	Objeto personalizado									
	Modelo ?	x_coche	En las aplicaciones ?										
	Pedido ?	id											
	Modelo transitorio ?	<input type="checkbox"/>											
	Tiene un hilo de correos ?	<input type="checkbox"/>											
	Tiene actividad de correo ?	<input type="checkbox"/>											
	Tiene lista negra de correo ?	<input type="checkbox"/>											
<table border="1"> <tr> <td>Campos</td> <td>Permisos de acceso</td> <td>Reglas de registro</td> <td>Notas</td> <td>Vistas</td> <td></td> <td></td> </tr> </table>							Campos	Permisos de acceso	Reglas de registro	Notas	Vistas		
Campos	Permisos de acceso	Reglas de registro	Notas	Vistas									
Nombre de campo	Etiqueta de campo	Tipo de campo	Requerido	Sólo lectura	Indexado	Tipo							
_last_update	Last Modified on	fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base							
create_date	Created on	fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base							
create_uid	Created by	many2one	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base							
display_name	Display Name	carácter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base							
id	ID	entero	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base							
write_date	Last Updated on	fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base							
write_uid	Last Updated by	many2one	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base							
x_fechaalta	Fecha alta	fecha	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo personalizado							
x_modelo	Modelo	carácter	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo personalizado							
x_name	Name	carácter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo personalizado							
x_pasajeros	Pasajeros	entero	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo personalizado							

Faltaría una única cuestión, y es el hecho de que se debe indicar qué grupo o grupos van a tener permiso de acceso sobre esta nueva entidad creada. En este caso, vamos a decir que solo tendrán acceso al objeto **x_coche** los usuarios pertenecientes al grupo **Inventario/UsuarioAvanzado**.

Descripción del modelo ?		Coche	Tipo ?	Objeto personalizado		
Modelo ?	x_coche		En las aplicaciones ?			
Nombre	Grupo	Permiso de lectura	Permiso de escritura	Permiso de creación	Permiso de eliminación	
x_coche.UsuarioAvanzado	Inventario / UsuarioAvanzado	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

5.2. Creación de una nueva vista

Lo siguiente ahora sería crear las **vistas**. Para ello se va a las vistas, es decir, *Técnico, Interfaz de usuario, Vistas*. Se decide que para esta entidad vamos a tener vista de tipo formulario (form) y árbol (tree). Comenzamos por la vista tipo árbol:

- Esta vista trabaja sobre el modelo creado antes **x_coche**.
- En *Estructura* se le debe añadir el código XML, indicando que se vea solo el modelo y la fecha de alta, sin mostrar los pasajeros.

Nombre de la vista? coche.tree

Tipo de vista? Árbol

Modelo? x_coche

Secuencia? 16

Activo?

Campo hijo?

Vista heredada?

Modo de herencia de la vista? Vista base

Datos del modelo?

ID externo?

Considera 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 estándar a través de vistas heredadas o mediante la personalización en Odoo Studio.

Arquitectura Permisos de acceso Vistas heredadas

```

1 <tree string="coches">
2   <field name="x_modelo"/>
3   <field name="x_fechaalta"/>
4 </tree>

```

Y ahora se crea la vista de tipo formulario, mostrando todos los campos, en este caso. En las vistas de formulario los campos se deben agrupar, indicándolo con `<group></group>` en el código XML.

Nombre de la vista? coche.form

Tipo de vista? Formulario

Modelo? x_coche

Secuencia? 16

Activo?

Campo hijo?

Vista heredada?

Modo de herencia de la vista? Vista base

Datos del modelo?

ID externo?

Considera 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 estándar a través de vistas heredadas o mediante la personalización en Odoo Studio.

Arquitectura Permisos de acceso Vistas heredadas

```

1 <form string="Coche">
2   <sheet name="Product_form">
3     <group>
4       <field name="x_modelo"/>
5       <field name="x_fechaalta"/>
6       <field name="x_pasajeros"/>
7     </group>
8   </sheet>
9 </form>

```

5.3. Creación de una nueva acción

Una vez creada ya la entidad con sus tres campos, y dos vistas para ella, una de tipo árbol y otra de tipo formulario, faltaría definir la **acción** que lleva a esas dos vistas. Las acciones están en **Técnico**, **Acciones**, y más concretamente, las que vamos a crear son *Acciones de ventana*. Una acción de ventana es una acción que va a enlazar una opción de menú (que será lo último en crear) con las vistas o ventanas correspondientes.

Se crea una nueva acción de ventana llamada Coches, que trabaja sobre el ID x_coche. Si se quisieran añadir más vistas, además de tree y form, se crearían y se añadirían aquí.

Nombre de la acción ?	Coches	Uso de la acción ?	
ID externo ?		Tipo de acción ?	ir.actions.act_window
Objeto ?	x_coche	Ventana destino ?	Ventana actual
Opciones generales		Seguridad	
VISTAS		FILTROS	
Modo de vista ?	tree,form	Valor del dominio ?	
Ref. de la vista ?		Valor de contexto ?	{}
Ref. de vista de búsqueda ?		Límite ?	80
		Filtro ?	<input type="checkbox"/>

5.4. Creación de un nuevo elemento de menú

Por último, ya solo falta crear el **elemento de menú**. Vamos a elementos de menú y lo creamos, indicando por ejemplo que pertenece a los productos del inventario, es decir, su menú padre es **Inventario/Productos**. Y se le asigna la acción de ventana que se acaba de crear. Además, se debe recordar también que, si no se asigna a ningún grupo, aparece para todos (punto 3 de este bloque de contenido).

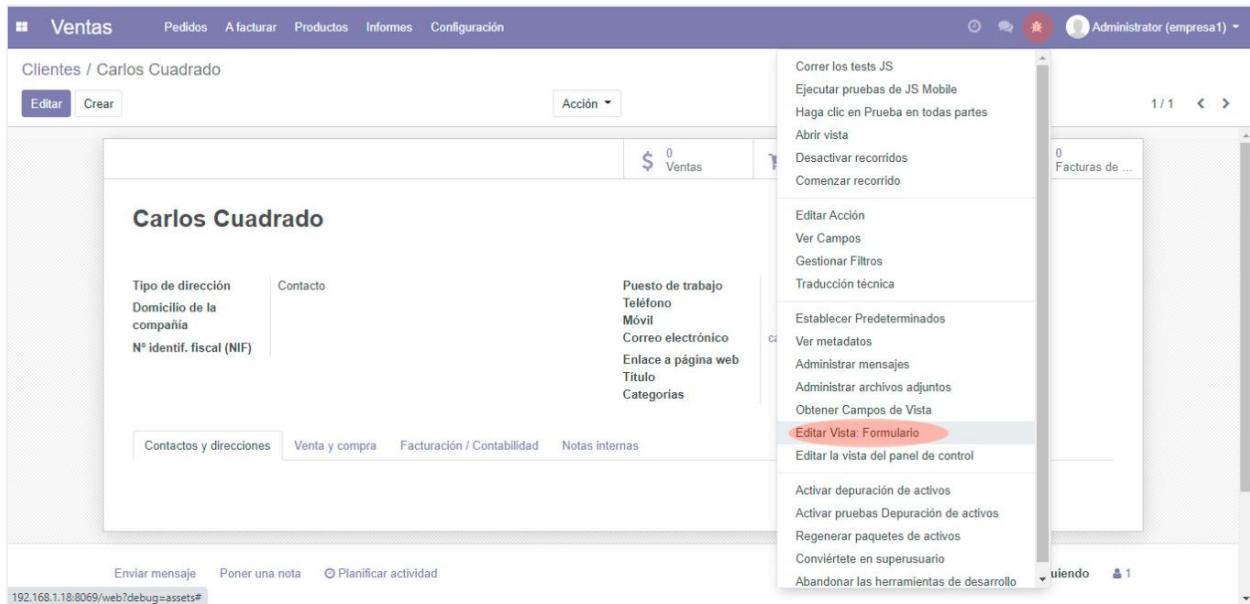
Elementos del menú / Nuevo			
Menú ?	Coches	Ruta completa ?	Inventario/Productos/Coches
Menú padre ?	Inventario/Productos	Acción ?	ir.actions.act_w Coches
Secuencia ?	10	Archivo de icono web ?	
Permisos de acceso		Imagen de icono web ?	
		SUBA SU ARCHIVO	
Nombre del grupo			
Añadir una línea			

Si ahora se refresca (F5) se ve que aparece Coches en **Inventario/Productos**, pero únicamente para **Inventario/Usuario Avanzado**, que es al que le dimos permisos de acceso el nuevo objeto x_coche (punto 5.1).

5.5. Creación de relaciones

Para completar lo visto en los cuatro subapartados anteriores, se va a ver aquí cómo crear una **relación** en un objeto de Odoo que creamos nosotros mismos. Se va a ampliar ahora el ejemplo suponiendo que el coche tendrá un propietario, que será uno de los clientes del sistema.

Se debe establecer la relación entre el objeto coche y el objeto cliente. Para ello, lo primero es saber qué entidad representa el objeto Cliente. Una de las formas sería, teniendo activo el modo desarrollador como siempre para este tipo de acciones, entrar en la vista del cliente y editar la vista, de forma similar a la realizada en el apartado 4.



Haciendo esto se ve que el objeto que representa un cliente, o lo que es lo mismo, el modelo que aparece como una tabla en la base de datos, es el **res.partner**. Contra este objeto es contra el que se tiene que hacer la relación.

Lo siguiente que se debe hacer es modificar el modelo **x_coche** para meterle un campo más que represente la relación. Esta relación, correspondiendo con los tipos de relación que hay en una base de datos del modelo relacional, puede ser *one2many*, *many2one* o *many2many*. En este caso, vamos a suponer que un coche solo tiene un propietario, luego sería una relación *many2one*. Esto quiere decir que un propietario puede tener varios coches, y un coche solo pertenece a un propietario. Y hay que indicarle con qué objeto se establece la relación, que sería el objeto *res.partner*.

Crear Campos

Nombre de campo ?	x_propietario	Tipo de campo ?	many2one																								
Etiqueta de campo ?	Propietario	Campo ayuda ?																									
<input checked="" type="radio"/> Propiedades <input type="radio"/> Permisos de acceso <input type="radio"/> Varios																											
PROPIEDADES BASE <table border="1"> <tr> <td>Requerido ?</td> <td><input type="checkbox"/></td> <td>Modelo relacionado ?</td> <td>res.partner</td> </tr> <tr> <td>Sólo lectura ?</td> <td><input type="checkbox"/></td> <td>Al eliminar ?</td> <td>Establecer a NULL</td> </tr> <tr> <td>Almacenado ?</td> <td><input checked="" type="checkbox"/></td> <td>Dominio ?</td> <td><input type="checkbox"/></td> </tr> <tr> <td>Indexado ?</td> <td><input type="checkbox"/></td> <td colspan="2"></td> </tr> <tr> <td>Copiado ?</td> <td><input checked="" type="checkbox"/></td> <td colspan="2"></td> </tr> <tr> <td>Activar el seguimiento de pedidos ?</td> <td>0</td> <td colspan="2"></td> </tr> </table>				Requerido ?	<input type="checkbox"/>	Modelo relacionado ?	res.partner	Sólo lectura ?	<input type="checkbox"/>	Al eliminar ?	Establecer a NULL	Almacenado ?	<input checked="" type="checkbox"/>	Dominio ?	<input type="checkbox"/>	Indexado ?	<input type="checkbox"/>			Copiado ?	<input checked="" type="checkbox"/>			Activar el seguimiento de pedidos ?	0		
Requerido ?	<input type="checkbox"/>	Modelo relacionado ?	res.partner																								
Sólo lectura ?	<input type="checkbox"/>	Al eliminar ?	Establecer a NULL																								
Almacenado ?	<input checked="" type="checkbox"/>	Dominio ?	<input type="checkbox"/>																								
Indexado ?	<input type="checkbox"/>																										
Copiado ?	<input checked="" type="checkbox"/>																										
Activar el seguimiento de pedidos ?	0																										
PROPIEDADES AVANZADAS																											

Por último, hay que cambiar la vista, porque si no, no se va a ver. Lo añadimos únicamente a la vista de formulario.

Nombre de la vista ?	coche.form	Campo hijo ?	
Tipo de vista ?	Formulario	Vista heredada ?	
Modelo ?	x_coche	Modo de herencia de la vista ?	Vista base
Secuencia ?	16	Datos del modelo ?	
Activo ?	<input checked="" type="checkbox"/>	ID externo ?	
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 estándar a través de vistas heredadas o mediante la personalización en Odoo Studio.			
<input checked="" type="radio"/> Arquitectura <input type="radio"/> Permisos de acceso <input type="radio"/> Vistas heredadas			
<pre> 1 <form string="Coche"> 2 <sheet name="Product_form"> 3 <group> 4 <field name="x_modelo"/> 5 <field name="x_fechalista"/> 6 <field name="x_pasajeros"/> 7 <field name="x_propietario"/> 8 </group> 9 </sheet> 10 </form> </pre>			

Si ahora se refresca la aplicación para probarlo (F5), se comprueba que se le puede asignar un propietario a un coche, listando como posibles propietarios todos los registros que hay en

`res.partner`. Se puede observar que en `res.partner` se tratan más elementos que los únicamente clientes: se muestra el administrador, los clientes, la compañía e incluso proveedores, si los hubiera. En la vista de cliente se filtra para que únicamente se muestren los clientes, dentro de todos los objetos a los que hace referencia `res.partner`.

6. Tableros en Odoo

En este apartado se va a estudiar cómo tratar unas vistas de Odoo un poco especiales, los tableros. Un tablero no es más que una vista que tiene varias vistas incrustadas dentro. Además, los tableros son vistas totalmente funcionales que sirven para tener una visión un poco más general, un resumen global de alguna aplicación del ERP. Por ejemplo, para la aplicación Inventario, el tablero sería este:

The screenshot shows the Odoo Inventory dashboard. At the top, there's a header bar with tabs for 'Inventario', 'Información general', 'Operaciones', and 'Datos principales'. On the right side of the header, there are user icons and a dropdown for 'Juan (empresa1)'. Below the header, the main area has a title 'Resumen de inventario'. There are two cards: 'Recepciones' (with '0 A Procesar') and 'Órdenes de entrega' (with '0 A Procesar'). At the bottom of the dashboard, there are buttons for 'Filtros', 'Agrupar por', and 'Favoritos', along with a search bar and a page number '1-2 / 2'.

Para crear un tablero nuevo, aunque se podría hacer a mano utilizando XML, lo ideal es instalar una aplicación de Odoo que maneja tableros.

The screenshot shows the Odoo App Store listing for the 'Tableros' module. It features a pink icon with a circular dial-like design. The title is 'Tableros' and the subtitle is 'Build your own dashboards'. Below the title are 'Instalar' (Install) and 'Información del módulo' (Module Information) buttons. To the right, there's another card for 'Tableros board' with its own 'Instalar' button and 'Información del módulo' button.

Con esto se pueden meter una serie de vistas en un tablero. Por ejemplo, digamos que queremos ver en un nuevo tablero los pedidos de venta. Simplemente se va a la opción Pedidos de ventas y se selecciona en *Favoritos*, *Añadir a mi tablero*.

The screenshot shows the Odoo Sales dashboard for 'Pedidos de venta'. The top navigation bar includes 'Ventas', 'Pedidos', 'A facturar', 'Productos', 'Informes', and 'Configuración'. The main area lists a single order: 'S00001' from '15/01/2026' to 'Eva García' (Commercial: 'Administrator'). A context menu is open over this order, with options like 'Guardar búsqueda actual', 'Importar registros', 'Añadir a mi tablero', and 'Pedidos de venta'. A sub-menu for 'Pedidos de venta' shows a table with columns 'Total' and 'Estado de la factura', and a button 'Añadir'.

Se puede comprobar refrescando, yendo a Tableros y viendo que aparecen los pedidos de venta.

Se pueden añadir ahora pedidos de compra, realizando el mismo proceso. Si ahora se va a Tablero, se puede ver que existen dos vistas, una correspondiente a pedidos de venta y la otra correspondiente a pedidos de compra.

Número	Fecha de pedido	Cliente	Comercial	Actividades	Total	Estado de la factura
S00001	15/01/2026	Eva García	A Administrator		2,42 €	A facturar

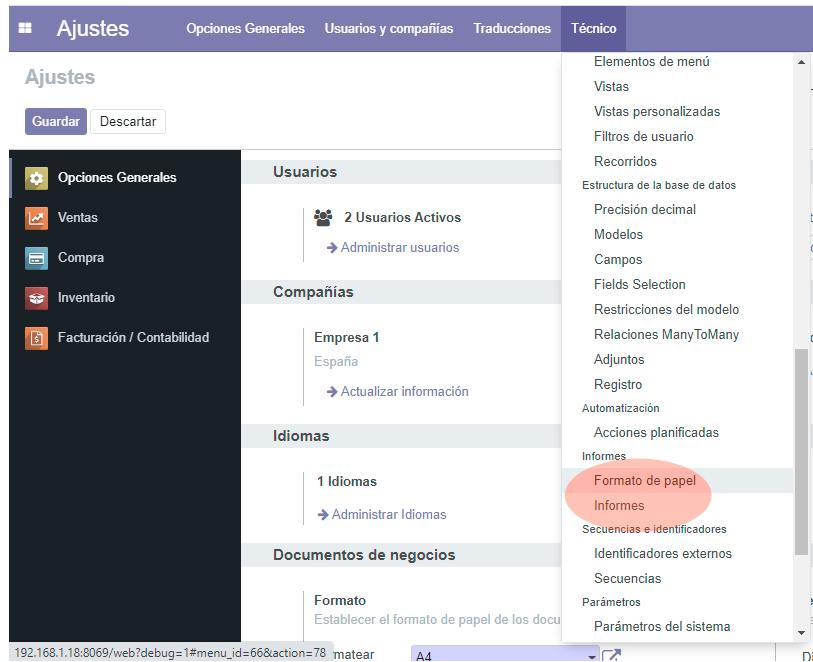
Se añade para terminar otro tipo de vista, esta vez pedidos de venta, pero vista gráfica. El proceso es el mismo.

Luego, en la pantalla Tableros, se puede cambiar el orden de las vistas y también el diseño, pudiendo modificar el ancho de las columnas, y quedando todas las vistas totalmente funcionales.

7. Informes en Odoo

Vamos a ver en este punto cómo trabajar con informes en Odoo. Un informe no es más que una forma de mostrar información de tal manera que no se pueda modificar, a diferencia de una vista, que se podría editar. Un informe sería, por ejemplo, un .pdf de un pedido que ya es para imprimir. Los informes que vienen por defecto en Odoo suelen estar disponibles desde la opción *Imprimir*, en pantallas de pedidos de venta, pedidos de compra, etc. Se va a analizar ahora cómo se pueden modificar estos informes, e incluso crear informes nuevos.

Lo primero, como siempre, es tener activo el modo desarrollador. Si miramos en la opción informes, es decir, *Técnico, Informes*, se ve que aparecen dos opciones: *Formato de papel*, donde se indica el formato del papel, e *Informes*, que si pinchamos nos muestra los informes presentes en Odoo para las aplicaciones instaladas.



Como se puede ver, los nombres de los informes son bastante descriptivos. Entramos por ejemplo al pedido de venta, Presupuesto/Pedido. En esta pantalla se muestra, entre otra información, el nombre de la plantilla, en este caso **sale.report_saleorder**. Este nombre representa el XML que define este informe. Así mismo, un informe no tiene por qué tener un solo XML, puede estar compuesto por subinformes. Para saber si es un informe compuesto, se pincha en Vistas **QWeb**, y se ve que este informe está formado por 5 informes diferentes, siendo el principal siempre el que se muestra en la pantalla inicial para ir a los subinformes. Si se selecciona cualquiera de ellos, se puede ver el código XML que lo compone.

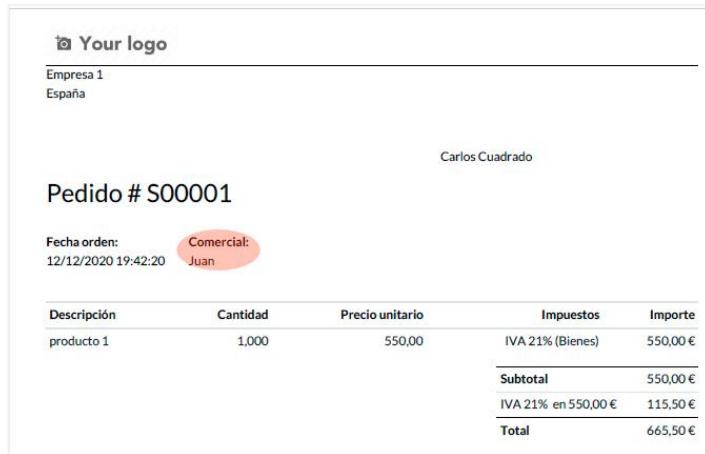
This screenshot shows the configuration for the 'Formato de papel' report. It includes fields for 'Nombre de la acción' (Presupuesto / Pedido), 'Tipo de informe' (PDF), 'Formato de papel' (A4), and 'Nombre del modelo' (sale.order). The 'Nombre de plantilla' field contains the value 'sale.report_saleorder', which is highlighted with a red circle. Other fields include 'Nombre del informe impreso' with the value '(object.state in ('draft', 'sent') and 'Presupuesto - %s' % (objectname)) or'. At the bottom, there are tabs for 'Seguridad' and 'Propiedades avanzadas', and a 'Nombre del grupo' section with a 'Añadir una línea' button.

Ahora bien, ¿qué es eso de QWeb? QWeb es un formato en XML que utiliza Odoo para hacer informes. La documentación referente a este 'lenguaje' se puede ver en el siguiente enlace, que conviene tener a mano para manipular informes en Odoo:

<https://www.odoo.com/documentation/16.0/reference/qweb.html>

Para llamar a otros informes que estén dentro del que se está analizando se utiliza **t-call**. Con esta directiva se va estructurando la jerarquía de informes y subinformes para poder navegar por ellos.

Para poner un ejemplo de manipulación de un informe, nos basamos en el que se está tratando, Presupuesto/Pedido (sale.report_saleorder), y se va a intentar que no muestre el comercial, según se indica en la imagen.



Para quitar esto, lo primero sería localizar en el código qué es lo que muestra el comercial (Salesperson), y luego quitarlo o comentarlo (para comentar: `<!-- Código a comentar -->`). Si se busca un poco, se ve que es el siguiente trozo de código, en la plantilla **report_saleorder_document**:

```

<div t-if="doc.date_order and doc.state in ['draft', 'sent']" class="col-auto mw-100 mb-2">
    <strong>Quotation Date:</strong>
    <p class="m-0" t-field="doc.date_order" t-options="{'widget': 'date'}"/>
</div>
<div t-if="doc.validity_date and doc.state in ['draft', 'sent']" class="col-auto mw-100 mb-2" name="expiration_date">
    <strong>Expiration:</strong>
    <p class="m-0" t-field="doc.validity_date"/>
</div>
<div t-if="doc.user_id.name" class="col-auto mw-100 mb-2">
    <strong>Salesperson:</strong>
    <p class="m-0" t-field="doc.user_id"/>
</div>
</div>

<!-- Is there a discount on at least one line? -->
<t t-set="display_discount" t-value="any([l.discount for l in doc.order_line])"/>

<table class="table table-sm o_main_table">
    <thead>
        <tr>
            <th name="th_description" class="text-left">Description</th>
            <th name="th_quantity" class="text-right">Quantity</th>
            <th name="th_priceunit" class="text-right">Unit Price</th>
            <th name="th_discount" t-if="display_discount" class="text-right" groups="product.group_discount_per_so_line">
                <span>Disc.%</span>
            </th>
            <th name="th_taxes" class="text-right">Taxes</th>
            <th name="th_subtotal" class="text-right">
                <t groups="account.group_show_line_subtotals_tax_excluded">Amount</t>
                <t groups="account.group_show_line_subtotals_tax_included">Total Price</t>
            </th>
        </tr>
    </thead>

```

Si una vez quitado o comentado ese trozo de código se imprime de nuevo el informe, se puede comprobar que ya no muestra la información relativa al comercial.



Por último, hay que señalar que un informe se puede mostrar de diferentes maneras. Por defecto se muestra en .pdf, pero se puede cambiar, por ejemplo, para que se muestre desde el navegador. Para hacer eso simplemente se edita el informe en cuestión, y en **Tipo de informe** se selecciona el tipo que se prefiera, en este caso, HTML.

7.1. Creación de informes

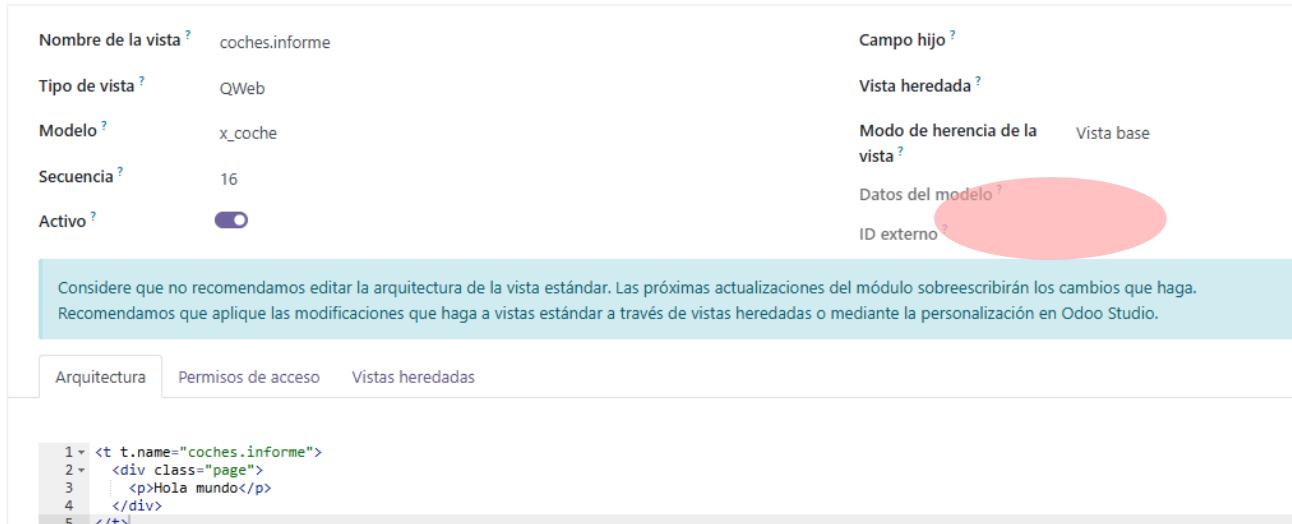
Para la creación de un informe desde cero, lo primero es activar el modo desarrollador. Vamos a los informes (**Técnico/Informes**), y en vez de tocar un informe ya hecho, pinchamos en *Crear* uno nuevo.

Siguiendo con el ejemplo de puntos anteriores, vamos a hacer un informe para sacar los datos de un coche, entidad creada por nosotros en el apartado 5 de esta documentación.

- Le indicamos un nombre y le ponemos que sea de tipo HTML, para que no esté descargando cada vez que hacemos pruebas.
- Es importante señalar sobre qué modelo va a trabajar, o lo que es lo mismo, sobre qué tabla de la base de datos va a consultar la información, en este caso, **x_coche**.
- El nombre de la plantilla debe ser el nombre de la vista que se va a crear para ese informe.

		Añadir Al Men... Vistas QWeb	
Nombre de la acción ?	Informe_coche	Nombre del modelo ?	x_coche
Tipo de informe ?	HTML	Nombre de plantilla ?	coches.informe
Seguridad		Propiedades avanzadas	
Nombre del grupo			
Añadir una línea			

Lo siguiente es crear una vista QWeb con el código XML necesario para que el informe se muestre, recordando el nombre que le pusimos a la plantilla del informe (**coches.informe**).



Nombre de la vista? coches.informe

Tipo de vista? QWeb

Modelo? x_coche

Secuencia? 16

Activo?

Campo hijo? Vista heredada?

Modo de herencia de la vista? Vista base

Datos del modelo?

ID externo?

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 estándar a través de vistas heredadas o mediante la personalización en Odoo Studio.

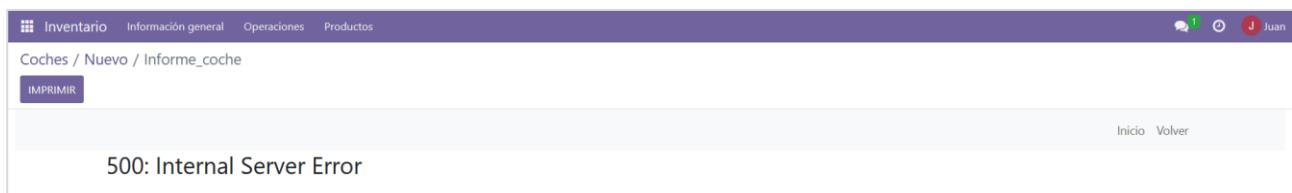
Arquitectura Permisos de acceso Vistas heredadas

```

1 <t t.name="coches.informe">
2 <div class="page">
3 | <p>Hola mundo</p>
4 </div>
5 </t>

```

Una vez hecho todo esto, creado el informe y su vista QWeb correspondiente, se va al informe del coche (Informe_coche) y se pulsa **Añadir al menú ‘imprimir’**. Sin embargo, todavía no funcionaría, ya que no enlaza bien. Para ello, lo que tiene que coincidir entre la vista y el informe es el **ID externo** de la vista, según se indica en la imagen anterior. Se puede ver, en este punto, el error que da si se pretende imprimir un informe de un coche.



Inventario Información general Operaciones Productos

Coches / Nuevo / Informe_coche

IMPRIMIR

Inicio Volver

500: Internal Server Error

Se debe conseguir para solucionar esto que enlace con el ID externo, es decir, que aparezca en la vista el ID externo.



	Nombre de la vista	Tipo de vista	Modelo	ID externo	Vista heredada
<input type="checkbox"/>	coches.informe	QWeb	x_coche		

Para enlazar el ID externo, se va a *Técnico/Secuencias e identificadores/Identificadores externos*. Para hacer esto de manera más cómoda, se puede abrir en otra pestaña del navegador, dejando en una pestaña la vista QWeb y en la otra lo referente a identificadores externos. Se debe crear

un nuevo identificador.

- Para llenar el Módulo y el Identificador externo se debe poner la combinación de palabras del nombre que se le haya dado a la vista del informe, en este caso **coches.informe**.
- En *Nombre del modelo* e *ID de registro* se debe indicar la manera en la que este identificador externo hace referencia a la vista coches.informe. Para conocer estos datos, en la pestaña de la vista del informe hay que fijarse en el modelo de la vista y en su id, **en la barra de direcciones del navegador**.

Si ahora se refresca la pantalla y se va a la vista, se debería ver que el ID externo ya se ha rellenado automáticamente.

A partir de este momento, el informe ya está configurado y se ve.

The screenshot shows the Odoo web interface. At the top, there's a purple header bar with navigation links: 'Inventario', 'Información general', 'Operaciones', and 'Productos'. On the right side of the header, there are icons for notifications (with a green '1'), a clock, and a user profile labeled 'Juan'. Below the header, the main content area has a title 'Informe_coche' and a blue 'IMPRIMIR' button. The main body contains the text 'Hola mundo'.

Lo siguiente sería editar el código XML de la vista del informe para que muestre información relevante, para lo cual es conveniente tener a mano la página de QWeb señalada antes.

- Para acceder a un objeto existe una variable XML que es la variable **docs**. Sin embargo, para no acceder directamente a la variable docs, lo mejor es crear nuestra propia variable, a la que llamamos, por ejemplo, coche:
- `<t t-set="coche" t-value="docs"/>`
- Para mostrar cada campo, con **t-field**. Y poner un **div** por cada campo para que lo ponga todo junto, en bloque.
- Se pueden añadir clases, utilizar cualquier etiqueta XML, meter estilos propios, etc.

```
<t t.name="coches.informe">
<div class="page">
    <p>Información sobre el coche</p>
    <t t-set="coche" t-value="docs"/>
    <div>
        <strong>Modelo:</strong>
        <span t-field="coche.x_modelo"/>
    </div>
    <div>
        <strong>Propietario:</strong>
        <span t-field="coche.x_propietario"/>
    </div>
</div>
</t>
```

The screenshot shows the Odoo interface with the XML code on the left and the resulting report on the right. The XML code defines a template for a report named 'coches.informe'. It starts with a 'div' tag with the class 'page'. Inside, there's a paragraph 'Información sobre el coche'. Then, it uses a 't' tag to set a variable 'coche' to the value of 'docs'. This is followed by two nested 'div' tags. Each nested 'div' contains a 'strong' tag for a field name ('Modelo:' or 'Propietario:') and a 'span' tag with a 't-field' attribute that refers to the corresponding field in the 'coche' variable. The resulting report on the right shows the heading 'Coches / Nuevo / Informe_coche' and a blue 'IMPRIMIR' button. The content area displays the text 'Información sobre el coche', 'Modelo: León 1.5', and 'Propietario: Support Inet4U, Carlos Cuadrado'.

8. Herencia de vistas

Heredar vistas sirve para poder modificar una vista existente sin tener que tocar el código XML original de la vista, es decir, se parte ya de una vista que no se modifica. A partir de esa vista se puede añadir código, pero el código original se aprovecha.

Para probar esto se va a añadir un nuevo campo (de nombre `x_peso`) en el objeto que representa un socio, que es el objeto `res.partner`, y luego se meterá en la vista de formulario.

The screenshot shows two Odoo configuration screens. The top screen displays the 'Modelos / Contacto' configuration for the 'Contacto' model (res.partner). It shows fields like 'Nombre del modelo' (Contacto), 'Tipo' (Objeto base), and 'En las aplicaciones' (account, auth_signup, base, base_vat, bus, mail, partner_autocomplete, payment, phone_validation, portal, privacy_lookup, product, purchase, purchase_stock, sale, sales_team, sms, snailmail, stock). The bottom screen shows the 'Crear Campos' (Create Fields) dialog for creating a new field 'x_peso' of type 'número flotante' (float). The 'Propiedades' tab is selected, showing properties like 'Requerido' (Required) checked, 'Almacenado' (Stored) checked, and 'Copiado' (Copied) checked. The 'PROPIEDADES BASE' section includes 'Activar el seguimiento de pedidos' (Enable tracking of orders) set to 0.

Como el objeto cliente ahora tiene un campo más, habría que ir a la vista formulario del cliente y añadir el nuevo campo. Sin embargo, no se hará así, sino que se creará una nueva vista que herede de la vista cliente y se le añadirá el campo creado, de forma que la vista original de cliente no se modifica.

Lo primero sería abrir la vista original y observar su código XML.

The screenshot shows the Odoo application interface. On the left, there's a navigation bar with links like 'Ventas', 'Pedidos', 'A facturar', 'Productos', 'Informes', and 'Configuración'. Below this is a search bar with the text 'Clientes / Support Inet4U, Carlos Cuadrado'. The main content area displays a contact record for 'Carlos Cuadrado' with details such as 'Support Inet4U', 'Contacto', and 'NIF: FR06330703844'. To the right of the contact card is a vertical sidebar with a context menu. The menu includes options like 'Ejecutar pruebas JS', 'Edita vista:Form' (which is highlighted with a red oval), and 'Activar depuración de activos'. At the bottom of the sidebar, there are tabs for 'Contactos y direcciones', 'Ventas y compras', 'Facturación / Contabilidad', and 'Notas internas'.

En el código se ve que hay un elemento de tipo **notebook** que tiene dentro pestañas, representando cada una de las pestañas un elemento de tipo **page**.

```

56
57 <field name="title" options="{'no_open': true}" placeholder="e.g. MISTER" attrs="{'invisi
58 <field name="active_lang_count" invisible="1"/>
59 <field name="lang" attrs="{'invisible': [('active_lang_count', '<=', 1)]}"/>
60 </group>
61 </group>
62
63 <notebook colspan="4">
64 <page string="Contacts & Addresses" name="contact_addresses" autofocus="autofocus">
65 <field name="child_ids" mode="kanban" context="{'default_parent_id': active_id, 'default_street': s
66 <kanban>
67 <field name="id"/>
68 <field name="color"/>
69 <field name="name"/>
70 <field name="title"/>

```

Lo que se va a hacer es meter la información del campo nuevo como si fuera otra página, concretamente a continuación de la pestaña *Notas internas*. Pero como ya se dijo, sin modificar este código, luego se va a vistas (en otra pestaña del navegador, dejamos esta pestaña abierta) y se crea una nueva vista con los siguientes datos.

The screenshot shows the Odoo Studio interface for creating a new view. The view is named 'res.partner.form.otrosdatos' and is based on the 'res.partner' model. It has a sequence of 0 and is active. The 'Arquitectura' tab is selected, showing XML code that inserts a new group and field into the 'internal_notes' page.

```

1 <xpath expr="//page[@name='internal_notes']" position="after">
2 <page name="otros_datos" string="Otros datos">
3 <group>
4   <field name="x_peso"/>
5 </group>
6 </page>
7 </xpath>

```

- Nombre de la vista. Nombre de la vista original con un sufijo.
- Tipo de vista. El mismo que la vista original.
- Modelo. El mismo que la vista original.
- Vista heredada. La vista de la que se hereda, es decir, la vista original.
- Secuencia. Cuando existen dos o más vistas que hacen referencia sobre el mismo objeto, si no se le dice nada al ERP, **Odoo muestra la vista con una secuencia menor**. Por tanto, si se quiere que se muestre esta vista en lugar de la original, se debe poner un número de secuencia menor que la secuencia que tenga la vista original, que es 1. Por tanto, aquí se indica secuencia 0.
- Estructura. Se le añade el código XML indicado en la imagen, que se comenta a continuación:
 - o El comando que se utiliza para crear, eliminar o reemplazar elementos en vistas heredadas es **xpath**. Este comando nos permite buscar dentro de un XML escribiendo una expresión. La expresión en este caso es 'internal_notes', ya que es la pestaña a partir de la cual queremos mostrar el nuevo campo **x_peso**.
 - o Por otra parte, ya se dijo en puntos anteriores (5.2) que en las vistas de formulario los campos se deben agrupar, indicándolo con **<group></group>**.

Si ahora refrescamos y vamos a ver la vista de clientes, ya se puede comprobar que aparece una pestaña de **Otros datos** mostrando el peso, y ya sería totalmente funcional.

The screenshot shows the Odoo Client Record View for 'Carlos Cuadrado'. The top navigation bar includes 'Ventas', 'Pedidos', 'A facturar', 'Productos', 'Informes', and 'Configuración'. Below the header, it displays 'Clientes / Support Inet4U, Carlos Cuadrado' with a cloud and sync icon.

Key data points shown include:

- Ventas:** \$ 1 Ventas
- Compras:** 0 Compras
- Todavía Tasa De:** (partially visible)

Below the summary, the contact information for 'Carlos Cuadrado' is listed:

- Individuo** (radio button selected) **Compañía** (radio button)
- Nombre:** Carlos Cuadrado
- Organización:** Support Inet4U
- Contacto:** NIF: FR06330703844
- Puesto de trabajo:** Director ventas España
- Teléfono:** 6
- Móvil:** 6
- Correo electrónico:** carlos@inetfouru.com
- Sitio web:** p. ej. https://www.odoo.com
- Título:** p. ej. Señor
- Etiquetas:** p. ej., "B2B", "VIP", "consultoría", ...

Below the contact details, there are tabs for 'Contactos y direcciones', 'Ventas y compras', 'Facturación / Contabilidad', 'Notas internas', and 'Otros datos'. The 'Otros datos' tab is highlighted with a red oval. A specific field 'Peso' with value '82,00' is also highlighted with a red oval.