

Unidad 6



Visión general de Odoo ERP

Sistemas de gestión
empresarial



Índice



- 6.1. ORM (Object Relational Mapping)
- 6.2. Modelos
- 6.3. Interfaces de usuario: menú y opciones de menú
- 6.4. Interfaces de usuario: vistas
 - 6.4.1. Herencia de vistas
 - 6.4.2. Prioridad de las vistas
 - 6.4.3. Vistas complejas
- 6.5. Interfaces de usuario
 - 6.5.1. Acciones de ventana
 - 6.5.2. Tableros
 - 6.5.3. Informes. Qweb
- 6.6. Serialización e identificadores externos
- 6.7. Secuencias o contadores
- 6.8. Documentos y plantillas



Introducción

Odoo ERP está desarrollado en lenguaje Python y utiliza la base de datos PostgreSQL. Además, usa XML y Qweb, motor de plantillas basado en XML que permite generar páginas HTML.

Hay cuatro conceptos básicos para entender su estructura: plantillas, menús y acciones. Básicamente, se pueden definir de la siguiente manera:

- > Un modelo es un Objeto que contiene su propia descripción y sus relaciones con otros Objetos.
- > Una vista es lo que la aplicación muestra en la pantalla a través del navegador.
- > Una opción de menú es cada elección posible que se puede hacer dependiendo de la posición del usuario.
- > Una acción (ventana o menú) es lo que se ejecuta al pulsar dicha opción.

Tanto las vistas como los menús y las acciones se crean en archivos XML.

La herencia es la característica que un objeto adquiere de otro, por tanto, en un entorno de programación, permite que una clase adquiera propiedades de otra clase.

Una de las características más importantes de Odoo es la capacidad de añadir objetos sin modificar los objetos de base, de forma que se añaden las modificaciones por encima de dichos objetos.

Por tanto, el procedimiento estándar para adaptar los módulos de Odoo a las necesidades de una empresa en particular consiste en crear nuevos objetos agregando modificaciones a los existentes

Al finalizar esta unidad

- + Conoceremos los conceptos básicos sobre los que se desarrolla Odoo ERP
- + Entenderemos el concepto de modelo y los tipos de campo que lo componen.
- + Estudiaremos el funcionamiento de menús, opciones de menú y acciones de ventana.
- + Diferenciaremos las diferentes interfaces de usuario.
- + Comprenderemos en qué consiste el proceso de serialización.



6.1.

ORM (Object Relational Mapping)

Odoo está estructurada en tres niveles: datos, lógica y presentación.

- La capa de datos es el nivel más bajo. Odoo se basa en la base de datos PostgreSQL a través de la cual gestiona el almacenamiento y mantenimiento de datos. En cuanto a los archivos binarios (archivos de documentos, imágenes, etc.), generalmente los almacena en una carpeta llamada `filestore`.
- La capa lógica ubicada en el servidor Odoo es responsable de las interacciones con la capa de datos (solicitar y recibir datos). Este es el mapeador relacional de objetos (ORM) que se describirá en los siguientes párrafos. Realiza los cálculos y validaciones y envía los datos de vuelta al cliente. Odoo usa Python.
- La capa de presentación es la capa que expone los datos y permite que el usuario interactúe a través de una aplicación cliente que interactúa con la API ORM a través de llamadas a procedimientos remotos (RPC). Odoo usa XML, JavaScript y Qweb.

ORM o Object Mapping Engine es un framework que ofrece funciones para vincular la base de datos relacional a la estructura de la entidad de la aplicación para que los datos de la aplicación se asocien automáticamente con la base de datos.

Este mapeo automático vía ORM simplifica mucho el trabajo del programador, ya que las solicitudes CRUD (Create, Read, Update y Delete, en español Crear, Leer, Actualizar, Eliminar) pueden ser tratadas como si fueran objetos que pueden ser administrados a través de un lenguaje de programación. Permite olvidarse de la complejidad de las SQL al evitar escribirlas, lo que resulta en una reducción del código que accede a la base de datos. Sin embargo, puede generar código inesperado que puede afectar el rendimiento de la aplicación.

La principal ventaja es que modificar el modelo (por ejemplo, agregar un campo) no implica modificar todas las órdenes INSERT, SELECT y UPDATE para agregar la modificación.

Usando ORM, al agregar un atributo a una clase, se encargará de modificar las órdenes. En algunos ORM, también debe editar los archivos XML.

En este sentido, Odoo cambia CRUD por métodos Postgres en `models.Model`.

CRUD	Models.Model (postgres métodos)
Create	Create
Read	Search, browse
Update	Write
Delete	unlink



6.2.

Modelos

En Odoo, cualquier dato es accesible a través de modelos. Los modelos conforman las tablas de la BBDD. Cada modelo es un objeto que contiene información descriptiva en campos, información acerca de las relaciones con otros objetos, así como de la lógica de la empresa.

Estos objetos se declaran mediante modelos Python, que se pueden consultar en Ajustes/Técnico/Estructura de la base de datos/Modelos.

Ajustes				
Opciones generales Usuarios y compañías Traducciones Herramientas de ludificación Técnico				
Modelos				
<div>CREAR</div> <div> <div>↑ Filtros</div> <div>≡ Agrupar por</div> <div>★ Favoritos</div> </div>				
<input type="checkbox"/> Modelo	Descripción del modelo	Tipo	Modelo transitorio	
<input type="checkbox"/> project.update	Actualización del proyecto	Objeto base		
<input type="checkbox"/> stock.picking	Albarán	Objeto base		
<input type="checkbox"/> account.move	Asiento contable	Objeto base		
<input type="checkbox"/> mail.channel	Canal de conversaciones	Objeto base		
<input type="checkbox"/> res.company	Compañías	Objeto base		
<input type="checkbox"/> res.partner	Contacto	Objeto base		
<input type="checkbox"/> account.account	Cuenta	Objeto base		
<input type="checkbox"/> account.analytic.account	Cuenta analítica	Objeto base		
<input type="checkbox"/> slide.channel	Curso	Objeto base		
<input type="checkbox"/> hr.department	Departamento	Objeto base		
<input type="checkbox"/> gamification.challenge	Desafío de ludificación	Objeto base		
<input type="checkbox"/> stock.scrap	Desechar	Objeto base		
<input type="checkbox"/> account.journal	Diario	Objeto base		
<input type="checkbox"/> hr.employee	Empleado	Objeto base		

Imagen 2. Listado modelos Odoo.

De la misma forma que los modelos se pueden modificar, también se pueden crear desde el inicio. La descripción de un modelo incluye una descripción de los campos, módulos, autorizaciones de acceso, entre otras.

Ajustes

Opciones generales Usuarios y compañías Traducciones Herramientas de ludificación Técnico

Modelos / Nuevo

GUARDAR

DESCARTAR

Descripción del modelo

Modelo

Pedido

Modelo transitorio

Hilo de mensajes

Actividad de correo

Lista negra de correo

Tipo

Objeto personalizado

En las aplicaciones

Campos

Permisos de acceso

Reglas de registro

Notas

Vistas

Formularios de página web

Nombre de campo	Etiqueta de campo	Tipo de campo	Requerido	Sólo lectura	Indexado	Tipo
x_name	Name	Carácter				Campo personalizado

Agregar línea

CREAR UN MENÚ

Imagen 3. Creación modelo.

Son clases Python que se corresponden con tablas Postgres. Se escriben en Python y todos extienden la clase models.Model. Es el conjunto de campos y atributos que definen una empresa, es decir, determina de qué manera se almacenan, organizan y manipulan los datos.



Se nombran, indicando con un prefijo la aplicación a la que pertenecen, en singular aunque hay excepciones procedentes de las primeras versiones del módulo raíz, por ejemplo, el objeto `res.users` contiene los datos correspondientes a usuarios.

Existen tres tipos de modelos:

- > **Modelo (Model).** Modelo estándar más común, almacenado en la BBDD.
- > **Transitorios (TransientModel).** Basados en la clase `models.TransientModel`, empleados para almacenar datos temporalmente que se eliminan periódicamente de forma automática.
- > **Abstractos (AbstractModel).** Basados en la clase `models.AbstractModel` y utilizados para definir clases abstractas que pueden ser heredadas por múltiples modelos.

En el menú Técnico/Estructura de base de datos/Modelos se encuentra la lista de los modelos disponibles en la base de datos, cada uno está formado por datos y características, encontramos los siguientes campos:

- > **Campos básicos.** Formados por datos numéricos o de tipo texto como Integer, Boolean, Char, Float.
- > **Campos reservados.** creados automáticamente para su uso por el sistema, como `id`, `create_date`, `create_uid`, `write_date`
- > **Campos avanzados.** Odoo requiere un campo `name` en todos los modelos, como Binary, Html, Image, Monetary, Selection, Date.
- > **Campos calculados.** Toman su valor mediante una función, no de la BBDD. Se trata de un campo declarado de forma estándar pero que tiene entre sus argumentos uno llamado `compute` que referencia la función de cálculo definida en otra parte del código.
- > **Campos relacionales.** Definen las relaciones entre modelos vinculando los datos que contienen cada uno de ellos. En Odoo existen tres tipos:
 - » **Many2one.** Relaciones "de muchos a uno": relación unidireccional que selecciona un registro de otra tabla. Ejemplo: varios empleados pertenecen a un departamento.
 - » **Many2many.** Relaciones "de muchos a muchos": relación bidireccional. Ejemplo: los clientes pueden comprar distintos productos y cada producto puede ser adquirido por varios clientes.
 - » **One2many.** Relación "de uno a muchos": relación en un sentido que selecciona uno o varios registros de otra tabla. Ejemplo: un cliente ha efectuado varios pedidos. Un campo One2many debe tener un Many2one relacionado con él.



Los campos admiten atributos. Los más comunes son:

- > **String.** Etiqueta visible.
- > **Default.** Valor por defecto estático o calculado.
- > **Required.** (True) Obligatorio.
- > **Help.** ToolTip de ayuda.
- > **Readonly.** Solo lectura.
- > **Index.** Si el campo se indexa en la BBDD.
- > **Compute.** Transforma en campo calculado.
- > **Store.** (True) Almacena el resultado de campos calculados.

Las principales clases de Odoo podemos conocerlas utilizando pgAdmin, los más empleados son los que se muestran en la siguiente tabla:

Modelos Odoo		
Modelo	Tabla postgres	Descripción
res.partner	res_partner	Usuarios, contactos, clientes y proveedores
res.groups	res_groups	Grupos de usuarios
res.company	res_company	Compañías
ir.module.module	ir_module_module	Módulos disponibles y su estado de instalación
ir.ui.view	ir_ui_view	Tipos de vistas
ir.ui.menu	ir_ui_menu	Elementos de menú
ir.actions.act_window	ir_act_window_view	Acciones de ventana
ir.model	ir_model	Modelos usados
ir.model.fields	ir_model_fields	Campos funcionales y sus respectivas tablas
ir.model.data	ir_model_data	Tabla relacional identificador único-nombre
ir.model.acces	ir_model_acces	Permisos de usuarios
ir.sequence	ir_sequence	Campos que utilizan secuencias
ir.config.parameter	ir_config_parameter	Parámetros del sistema

6.3.

Interfaces de usuario: menú y opciones de menú

Un menú es un conjunto de opciones que aparecen en cualquier interfaz de Odoo, en áreas específicas como la parte superior o izquierda. Las opciones de menú son cada uno submenús de un menú.

Las opciones de menú que aparecen en la pantalla de la aplicación Odoo se asignan internamente a grupos y usuarios. Así, las más básicas son accesibles para todos los usuarios, pero existen opciones que pueden especificarse para usuarios concretos.



Imagen 4. Ejemplo de menú.

Se almacenan en el modelo `ir.ui.menu`.

En Ajustes/Técnico/Interfaz de usuario/Elementos de menú obtenemos una lista de opciones de menú disponibles, sus nombres responden a una estructura en árbol que los ubica en la Aplicación, menú y submenú.

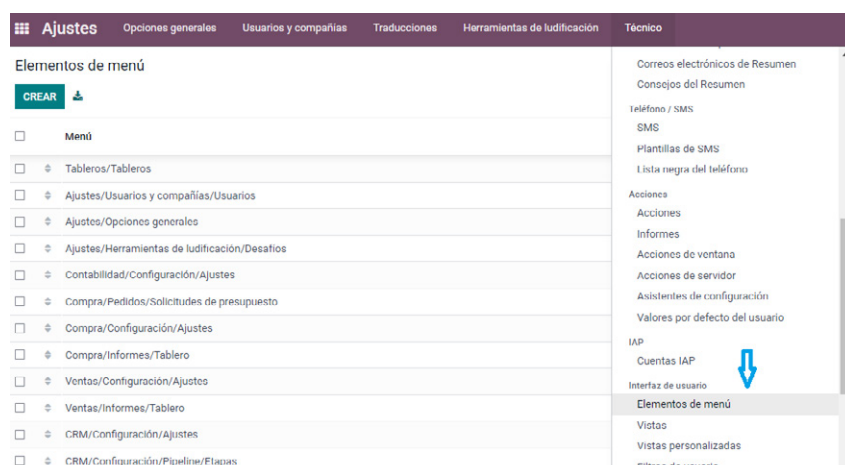


Imagen 5. Elementos menú.

Los menús pueden ser modificados o creados desde cero, además a cada uno podemos asignarle una serie de usuarios específicos, en caso contrario serán visible por todos los usuarios de la aplicación. El menú está relacionado con las vistas como estudiaremos en el siguiente punto.



6.4.

Interfaces de usuario: vistas

Una vista es un diseño XML que define cómo verá el usuario los registros de un modelo.

Existen tres vistas básicas: Kanban, Árbol y Formulario, aunque también existen vistas de búsqueda y vistas calendario, de Gantt, de Gráfico, de Actividad, etc.

```
< record id= "MODEL_view_TYPE" model = "ir.ui.view" >
  < field name = "name" > NAME </field>
  < field name = "model" > MODEL </field>
  < field name = "arch" type = "xml" >

    <VIEW_TYPE>
      < VIEW_SPECIFICICATIONS />
    </VIEW_TYPE>
  </field>
</record>
```

El único campo obligatorio es el "name" que identifica a la vista por su nombre. El aspecto de la vista estará formado por instrucciones XML.

Todas las pantallas de Odoo están construidas como vista. Una vista actúa sobre un modelo y especifica qué campos se mostrarán y con qué diseño gráfico. Una vista es el resultado de una consulta a una o más tablas de la BBDD.

En Ajustes/Técnico/Interfaz de usuario/Vistas se obtiene la relación de vistas disponibles, cuyo nombre responde al formato modelo.nombre_descriptivo.extension, donde extensión puede ser tree, kanban, search...

Ajustes Opciones generales Usuarios y compañías Traducciones Herramientas de ludificación Técnico				
Vistas				
<input type="button" value="CREAR"/> <input type="button" value="Filtros"/> <input type="button" value="Agrupar por"/> <input type="button" value="Favoritos"/>				
<input type="checkbox"/>	Nombre de la vista	Tipo de vista	Modelo	ID externo
<input type="checkbox"/>	res.config.settings.view.form.inherit.base.setup	Formulario	res.config.settings	base_setup.res_config_settings_view_form
<input type="checkbox"/>	Upload Bank Statements	Formulario	account.bank.statement.import	account_bank_statement_import.account_bank_statement_import.view
<input type="checkbox"/>	account.analytic.line.form	Formulario	account.analytic.line	analytic.view_account_analytic_line_form
<input type="checkbox"/>	account.analytic.line.form	Formulario	account.analytic.line	hr_timesheet_hr_timesheet_line_form
<input type="checkbox"/>	account.bank.statement.form	Formulario	account.bank.statement	account.view_bank_statement_form
<input type="checkbox"/>	account.journal.form	Formulario	account.journal	account.view_account_journal_form
<input type="checkbox"/>	account.journal.group.form	Formulario	account.journal.group	account.view_account_journal_group_form
<input type="checkbox"/>	account.journal.group.tree	Árbol	account.journal.group	account.view_account_journal_group_tree
<input type="checkbox"/>	account.journal.kanban	Kanban	account.journal	account.account_journal_view_kanban
<input type="checkbox"/>	account.journal.search	Busqueda	account.journal	account.view_account_journal_search
<input type="checkbox"/>	account.journal.tree	Árbol	account.journal	account.view_account_journal_tree
<input type="checkbox"/>	account.move.line.tree.grouped	Árbol	account.move.line	account.view_move_line_tree_grouped
<input type="checkbox"/>	calendar.event.form	Formulario	calendar.event	calendar.view_calendar_event_form
<input type="checkbox"/>	crm.lead.kanban.lead	Kanban	crm.lead	crm.crm_case_kanban_view_leads
<input type="checkbox"/>	crm.lead.tree.opportunity	Árbol	crm.lead	crm.crm_case_tree_view_oppor
<input type="checkbox"/>	crm.stage.form	Formulario	crm.stage	crm.crm_stage_form

Imagen 6. Listado de vistas.



Podemos obtener información acerca del nombre, tipo y modelo de vista, su estructura XML y los diferentes permisos de acceso para los usuarios. Desde la opción de Herramientas de desarrollo podemos editar las diferentes vistas.

Imagen 7. Código vista.

6.4.1. Herencia de vistas

Podemos trabajar sobre vistas estándar de Odoo de forma que podemos modificarlas obteniendo nuevas vistas. Para crear vistas heredadas se emplea lenguaje xpath que permite elaborar especificaciones de acceso y aplicarlas a un documento XML, a través del cual se indican las modificaciones a realizar en la vista base.

Abrir: Vistas que heredan de ésta

Nombre de la vista	res.config.settings.view.form.inherit.web.recaptcha
Tipo de vista	Formulario
Clave	
Modelo	res.config.settings
Secuencia	16
Activo	<input checked="" type="checkbox"/>
Campo hijo	
Vista heredada	res.config.settings.view.form.inherit.base.setup
Modo de herencia de la vista	Vista de extensión
Datos del modelo	res.config.settings.view.form.inherit.web.recaptcha
ID externo	google_recaptcha.res_config_settings_view_form

Ten en cuenta que no recomendamos editar la arquitectura de la vista estándar. Las futuras actualizaciones del módulo van a sobrescribir los cambios que hagas.
Recomendamos aplicar las modificaciones que haga a vistas estándar a través de vistas heredadas o por medio de la personalización de Odoo Studio.

Estructura	Permisos de acceso	Vistas heredadas
ID	Secuen...	Nombre de la vista
ID externo		
Agregar línea		

Imagen 8. Vistas heredadas.

Accediendo a una vista heredada, podemos ver el tipo de herencia y el código xpath, como se muestra en la siguiente imagen:



Abrir: Vistas que heredan de ésta

Nombre de la vista	res.config.settings.view.form.inherit.web.recap
Tipo de vista	Formulario
Clave	
Modelo	res.config.settings
Secuencia	16
Activo	<input checked="" type="checkbox"/>
Campo hijo	
Vista heredada	res.config.settings.view.form.inherit.base.setup
Modo de herencia de la vista	Vista de extensión
Datos del modelo	res.config.settings.view.form.inherit.web.recaptcha
ID externo	google_recaptcha.res_config_settings_view_form

Ten en cuenta que no recomendamos editar la arquitectura de la vista estándar. Las futuras actualizaciones del módulo van a sobrescribir los cambios que hagas.
Recomendamos aplicar las modificaciones que haga a vistas estándar a través de vistas heredadas o por medio de la personalización de Odoo Studio.

Estructura Permisos de acceso Vistas heredadas

Editar traducciones

```

1 <?xml version="1.0"?>
2 <data><xpath expr="//div[@id='website_recaptcha_settings']/div[hasclass('text-muted')]" position="inside">
3   <p>If no keys are provided, no checks will be done.</p>
4 </xpath>
5 <div id="recaptcha_warning" position="replace">

```

Imagen 9. Vista código xpath.

6.4.2. Prioridad de las vistas

Dado que se pueden crear diferentes tipos de vistas a partir de cada modelo, es necesario controlar cuál de ellas se muestra por defecto al activar un elemento de menú o un botón.

Una vez en la vista predeterminada, se puede cambiar usando los botones de tipo de vista que aparecen en la parte superior derecha

La vista tiene un campo llamado Secuencia. Cuando más de una vista está asociada con un modelo, la vista con el valor más bajo en este campo se muestra de manera predeterminada, a menos que la acción de la ventana esté asociada con el elemento del menú que especifica el modo de visualización para mostrar, porque la acción de la ventana tiene prioridad sobre la secuencia.

6.4.3. Vistas complejas

Son aquellas que se componen de varias páginas, el código <page> </page> definirá cada una de las pestañas.

Nombre de la vista	res.partner.view.purchase.buttons	Campo hijo	
Tipo de vista	Formulario	Vista heredada	res.partner.form
Clave		Modo de herencia de la vista	Vista de extensión
Modelo	res.partner	Datos del modelo	res.partner.view.purchase.buttons
Secuencia	9	ID externo	purchase.res.partner.view.purchase.buttons
Activo	<input checked="" type="checkbox"/>		

Estructura Permisos de acceso Vistas heredadas

Editar traducciones

```

<?xml version="1.0"?>
<data><div name="button_box" position="inside">
  <button class="oe_stat_button" name="350" type="action" groups="purchase.group_purchase_user" icon="fa credit-card">
    <field string="Purchases" name="purchase_order_count" widget="statinfo"/>
  </button>
</div>
<page name="internal_notice" position="inside">
  <group colspan="2" col="3" groups="purchase.group_warning_purchase">
    <separator string="Warning on the Purchase Order" colspan="4"/>
    <field name="purchase_warn" nolabel="1"/>
    <field name="purchase_warn_msg" colspan="3" nolabel="1" attrs="{ 'required': [{'purchase_warn': '1'}, False], ('purchase_warn', '1'), 'no_message'}" />
  </group>
</page>
</data>

```

Imagen 10. Vista compleja.



6.5.

Interfaces de usuario

6.5.1. Acciones de ventana

Las Acciones son los métodos desde los que se abren las vistas.

Una Acción de ventana contiene las instrucciones para construir la interfaz que se mostrará cuando se presione un menú o botón. También se puede hablar de Acción de menú, que asocia una Opción de menú con la vista que se abre al pulsarla. Una Opción de menú no siempre está asociada con una acción. Son archivos XML que contienen instrucciones para crear una nueva vista al seleccionar un elemento de menú o un botón.

Las acciones de ventana se almacenan en el modelo `ir.actions`, `act_window` y también disponen de un procedimiento de herencia.

Los tipos de acciones más importantes son:

- > **Window:** abre una vista en una nueva ventana.
- > **Report:** imprime un informe.
- > **Wizard:** ejecuta un asistente para realizar un determinado trabajo.

Por tanto, para crear una acción de ventana que abra una vista se debe introducir al menos los siguientes datos:

- > Nombre de la acción.
- > Tipo de la acción
- > Nombre de la vista.
- > Tipo de vista

6.5.2. Tableros

Los tableros de vista Odoo contienen otras vistas dentro para proporcionar en una única pantalla información diversa.

Por tanto, se trata de un cuadro de mandos que se puede diseñar con XML, pero que tiene su propio módulo llamado Tableros, donde se pueden agregar las vistas seleccionadas de cualquier tipo en una sola.

6.5.3. Informes. Qweb

La mayoría de las empresas necesitan realizar cambios en los documentos e informes estándar proporcionados por un sistema ERP.

Los informes en Odoo son un tipo de vista que muestra información no editable. Cada Aplicación ofrece una amplia variedad de informes que se pueden ver en pantalla o convertir en archivos pdf.

Es por eso por lo que al instalar Odoo, también se instala la herramienta `wkhtmltopdf`.



Cada informe está asociado con el nombre del modelo XML, así como su diseño.



Imagen 11. Plantilla y acceso a Qweb.

En Odoo, es posible realizar algunos cambios sin modificar el documento. Además, Odoo dispone de un lenguaje de plantillas llamado QWeb que permite integrar datos en los informes, haciendo posible tanto editar los informes predeterminados incluidos como crear nuevos.

Qweb es el motor de plantillas, basado en XML, que a pesar de que se introdujo para mejorar las vistas Kanban, actualmente se usa para la generación de informes. La vista de tipo Qweb que los contiene por defecto es web.html_container.

Los informes disponen de encabezado y pie de página. Las últimas versiones de Odoo generan tanto los informes como las vistas del sitio web y permiten crear páginas web dinámicas vinculadas al ERP o no.

Los nombres internos de todos los informes empiezan por report_.

Informes				
<div> <div>CREAR</div> <div> <div>Buscar...</div> <div> <div>Filtros</div> <div>Agrupar por</div> <div>Favoritos</div> </div> </div> </div>				
<input type="checkbox"/> Nombre	Nombre del modelo	Tipo de acción	Nombre de plantilla	Tipo de informe
<input type="checkbox"/> Albarán	stock.picking	ir.actions.report	stock.report_deliveryslip	PDF
<input type="checkbox"/> Auditoría de Libros	account.common.journal.report	ir.actions.report	account.report_journal	PDF
<input type="checkbox"/> Código de barras de ubicación	stock.location	ir.actions.report	stock.report_location_barcode	PDF
<input type="checkbox"/> Código de barras del paquete (PDF)	stock.quant.package	ir.actions.report	stock.report_package_barcode_small	PDF
<input type="checkbox"/> Código de barras del paquete (ZPL)	stock.quant.package	ir.actions.report	stock.label_package_template_view	Texto
<input type="checkbox"/> Código de barras del paquete con contenido	stock.quant.package	ir.actions.report	stock.report_package_barcode	PDF
<input type="checkbox"/> Embalaje del Producto (PDF)	product.packaging	ir.actions.report	product.report_packagingbarcode	PDF
<input type="checkbox"/> Entradas en la hoja de horas	account.analytic.line	ir.actions.report	hr_timesheet.report_timesheet	PDF
<input type="checkbox"/> Entradas en la hoja de horas	account.analytic.line	ir.actions.report	hr_timesheet.report_timesheet_task	PDF
<input type="checkbox"/> Entradas en la hoja de horas	project.project	ir.actions.report	hr_timesheet.report_timesheet_project	PDF
<input type="checkbox"/> Entradas en la hoja de horas	project.task	ir.actions.report	hr_timesheet.report_project_task_timesheet	PDF
<input type="checkbox"/> Etiqueta de reporte de recepción	stock.move	ir.actions.report	stock.report_reception_report_label	PDF
<input type="checkbox"/> Etiqueta del Producto (PDF)	product.template	ir.actions.report	product.report_producttemplatelabel_dymo	PDF
<input type="checkbox"/> Etiqueta del Producto (PDF)	product.template	ir.actions.report	product.report_producttemplatelabel	PDF
<input type="checkbox"/> Etiqueta del producto (ZPL)	product.product	ir.actions.report	stock.label_product_product_view	Texto
<input type="checkbox"/> Extracto	account.bank.statement	ir.actions.report	account.report_statement	PDF
<input type="checkbox"/> Factura PRO-FORMA	sale.order	ir.actions.report	sale.report_saleorder_pro_forma	PDF
<input type="checkbox"/> Facturas	account.move	ir.actions.report	account.report_invoice_with_payments	PDF

Imagen 12. Lista de informes.

Pulsando la opción Vistas QWeb podemos ver el listado de vistas disponible, y cada una en su apartado de Estructura, contiene la plantilla Qweb XML.

Las plantillas de QWeb son fragmentos HTML reutilizables que se pueden insertar en otras plantillas.

Con el modo desarrollador activo, en Ajustes/Configuración/Técnico/Informes se obtiene una relación de los informes estándar.



6.6.

Serialización e identificadores externos

Los objetos pueden ser serializables. Llamamos serialización al proceso a través del cual un objeto se convierte en una secuencia de bytes para almacenarlo o transmitirlo a la memoria, a una base de datos o a un archivo.

Las diversas interfaces de usuario vistas hasta ahora se almacenan en registros internos de Odoo que utiliza este proceso para exportar e importar datos. Cada registro de la tabla de Odoo tiene un identificador único llamado "id" que se asigna secuencialmente y de forma automática cuando se crea. La etiqueta se compone del nombre del módulo y del nombre del identificador separado por puntos.

Para poder hacer referencia al registro relacionado con un id que aún no está creado, se utilizan los identificadores externos, que asignan identificadores con nombre a los registros, mediante una tabla que contiene y relaciona los id externos con nombre, con los id numéricos de la BBDD. Se trata del modelo `ir.model.data`.

En la opción Ajustes/Técnico/Secuencias e identificadores/identificadores externos, se puede crear:

The screenshot shows the 'Identificadores externos / Nuevo' form in Odoo. The sidebar on the left contains the following fields:

- Módulo: [text input]
- Identificador externo: [text input]
- No actualizable: ☐
- Última actualización el: [text input]
- Creado el: [text input]

The main area on the right contains the following fields:

- Nombre mostrado: [text input]
- Nombre del modelo: [text input]
- ID de registro: [text input]
- Registro: [text input]

Imagen 13. Nuevo identificador externo.

- > Los campos Módulo e Identificador externo enlazados separados por un punto deben coincidir con el nombre de la vista.
- > Nombre del modelo se obtiene de la url (`ir.ui.view`).
- > El ID de registro se obtiene de la url.

Agregar estos datos actualizará el campo ID externo de la vista utilizada para el informe. Los identificadores externos también son importantes durante la importación de datos ya que evitan la duplicidad actualizando la información en caso de importaciones repetidas.



6.7.

Secuencias o contadores

Una secuencia en Odoo es un contador secuencial automático. Los campos de secuencia permiten numerar cualquier entidad. Las secuencias en Odoo generan automáticamente un número consecutivo.

Las secuencias se componen de un prefijo, un elemento de relleno, un contador o número consecutivo y un sufijo.

Secuencias / Sales Order

EDITAR **CREAR** Acción

Nombre	Sales Order	Código de secuencia	sale.order
Implementación	Estándar	Activo	<input checked="" type="checkbox"/>

Secuencia

Prefijo	S	Tamaño de secuencia	5
Sufijo		Paso	1
Utilizar subsecuencias por date range	<input type="checkbox"/>	Próximo número	2

Leyenda (para prefijo, sufijo)

Año actual con el siglo: %(year)s	Día del año: %(doy)s	Hora 00->24: %(h24)s
Año actual sin siglo: %(y)s	Semana del año: %(woy)s	Hora 00->12: %(l12)s
Mes: %(month)s	Día de la semana (0: Lunes): %(weekday)s	Minutos: %(min)s
Día: %(day)s		Segundos: %(sec)s

Imagen 14. Secuencia.

Odoo crea secuencias predeterminadas para casi todos los formularios de Odoo: presupuestos, pedidos, albaranes, asientos contables... que se pueden modificar para adaptarlas a las necesidades de la empresa.

El modelo ir.sequence tiene asociadas las siguientes vistas:

Modelos / Secuencia

GUARDAR **DESCARTAR**

Descripción del modelo	Secuencia	Tipo	En las aplicaciones	Objeto base	base
Modelo	ir.sequence				
Pedido	name				
Modelo transitorio	<input type="checkbox"/>				
Hilo de mensajes	<input type="checkbox"/>				
Actividad de correo	<input type="checkbox"/>				
Lista negra de correo	<input type="checkbox"/>				

Campos	Permisos de acceso	Reglas de registro	Notas	Vistas	Formularios de página web
Nombre de la vista	Tipo de vista	Modelo	ID externo	Vista heredada	
ir.sequence form	Formulario	ir.sequence	base.sequence_view		
ir.sequence search	Búsqueda	ir.sequence	base.view_sequence_search		
ir.sequence tree	Árbol	ir.sequence	base.sequence_view_tree		

Imagen 15. Modelo ir.sequence.



Abrir: Vistas

Nombre de la vista	ir.sequence form	Campo hijo
Tipo de vista	Formulario	Vista heredada
Clave		Modo de herencia de la vista
Modelo	ir.sequence	Vista base
Secuencia	16	ir.sequence form
Activo	<input checked="" type="checkbox"/>	base.sequence_view

Este campo es el mismo que el campo 'arch' sin traducción

- o Campo: arch_base
- o Objeto: ir.ui.view
- o Tipo: text
- o Widget: ace
- o Modificadores: {'required': true}

Estructura

Permisos de acceso

Vistas heredadas

Editar traducciones

```

<?xml version="1.0"?>
<form string="Sequences">
  <sheet>
    <group>
      <group>
        <field name="name"/>
        <field name="implementation"/>
      </group>
      <group>
        <field name="code"/>
        <field name="active" widget="boolean_toggle"/>
        <field name="company_id" groups="base.group_multi_company"/>
      </group>
    </group>
    <notebook>
      <page string="Sequences" name="sequence">
        <group>
          <group>
            <field name="prefix"/>
            <field name="suffix"/>
            <field name="use_date_range"/>
          </group>
          <group>
            <field name="padding"/>
            <field name="number_increment"/>
            <field name="number_next_actual" string="Next Number" attrs="{ 'invisible': [ ('use_date_range', '=', True)] }"/>
          </group>
        </group>
        <field name="date_range_ids" attrs="{ 'invisible': [ ('use_date_range', '=', False)] }">
          <tree string="Sequences" editable="top">
            <field name="date_from"/>
            <field name="date_to"/>
            <field name="number_next_actual" string="Next Number"/>
          </tree>
        </field>
      </page>
    </notebook>
  </sheet>
</form>

```

Imagen 16. Vista Formulario.



6.8.

Documentos y plantillas

Odoo pone a disposición de sus usuarios plantillas de documentos comerciales y de correo electrónico editables y parametrizables según las preferencias de la empresa, a las que podemos acceder desde el menú Ajustes/Técnico/Plantillas.

Plantillas de correo electrónico / Factura: Enviar por correo electrónico

Editar Crear

Acción

Nombre: Factura: Enviar por correo electrónico

Aplica a: Asiento contable

Contenido Configuración del correo electrónico Configuración avanzada Generador dinámico de expresiones con marcadores de posición

Asunto: {{ object.company_id.name }} Factura (Ref {{ object.name or 'n/a' }})

Dear Brandon Freeman (Account Interior):

Here is your invoice INV/2021/05/0005 (with reference: SUB002) amounting in \$ 143,750.00 from YourCompany. This invoice is already paid.

Do not hesitate to contact us if you have any questions.

Mitchell Admin

Imagen 17. Ejemplo plantilla.

Es necesario especificar el objeto del cual se debe obtener la información para ser incorporada al texto mediante el uso de campos personalizados, como en la imagen 17 que hemos visto.

Recuerde que una vez configurado, se debe vincular una acción del servidor que indica como y cuando usar la plantilla.

Acciones de servidor / Nuevo

Guardar Descartar

Crear acción contextual

Nombre de acción: Por ejemplo, Actualizar la cantidad de pedido

Modelo: Pedido Acción a realizar: Actualizar el registro

Datos para escribir Seguridad

Configure el modelo que va a crear antes de seleccionar los valores

Campo	Tipo de evaluación	Registro	Valor
Agregar línea			

Imagen 18. Acción de servidor.



 www.universae.com

