

Sistemas de Gestión Empresarial

UNIDAD 06 - VISIÓN GENERAL DE ODOO ERP

INDICE

- 6.1 - Conceptos básicos.
- 6.2 - ORM (Object Relational Mapping).
- 6.3 - Modelos.
- 6.4 - Interfaces de usuario.

6.1 – Conceptos básicos.

6.1 - CONCEPTOS BÁSICOS

- Odoo ERP está desarrollado en Python.
- Usa BD PostgreSQL.
- Usa XML y Qweb para generar páginas HTML.



6.1 - CONCEPTOS BÁSICOS

- Existen cuatro conceptos básicos en la estructura.
 - ❖ *Modelo*: un objeto que contiene su propia descripción y sus relaciones con otros objetos.
 - ❖ *Vista*: mostrado en la pantalla.
 - ❖ *Menú*: elecciones posibles a partir de la ubicación del usuario en la vista.
 - ❖ *Acción*: lo que se ejecuta al pulsar una opción de menú o botón.

6.1 - CONCEPTOS BÁSICOS

- *Vistas, menús y acciones* se construyen en archivos XML.
- La *herencia* es la propiedad por la que un objeto deriva de otro extendiendo su funcionalidad.
- Odoo cuenta con un potente mecanismo de herencia que le permite añadir capas con modificaciones por encima de los objetos.

6.1 - CONCEPTOS BÁSICOS

- La *herencia* es aplicable tanto a los *modelos* como *vistas* y, en general, a la lógica de negocio.
- El procedimiento estándar para adaptar módulos de Odoo a nuestras necesidades consiste en crear nuevos objetos agregando las modificaciones

6.2 - ORM (Object Relational Mapping)

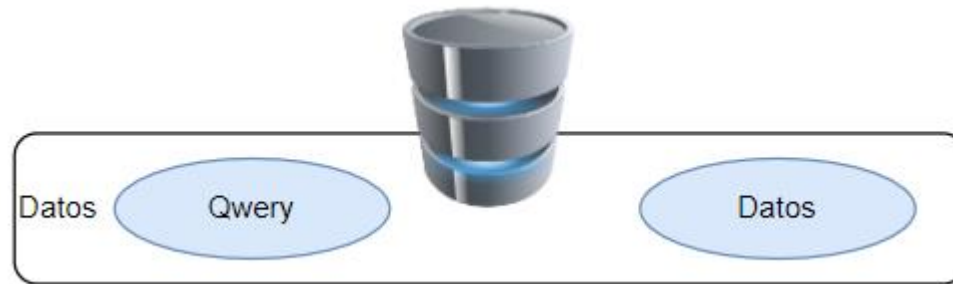
6.2 - ORM (Object Relational Mapping)

- La arquitectura de Odoo se estructura en tres niveles:
 - ❖ La capa de datos
 - ❖ La capa lógica
 - ❖ La capa de presentación

6.2 - ORM (Object Relational Mapping)

- **La capa de datos**

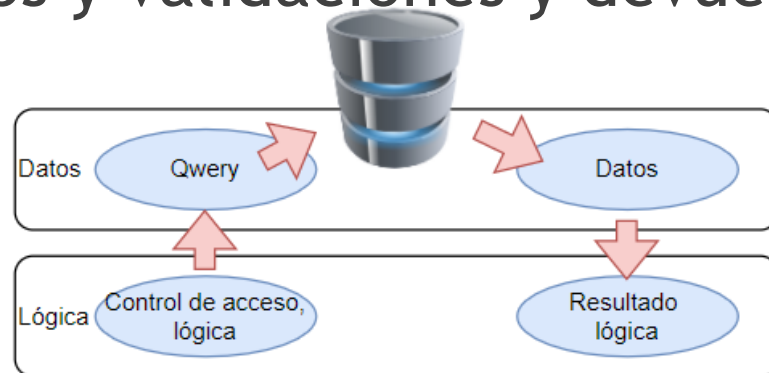
- ❖ Es el nivel más bajo.
- ❖ Almacenamiento y persistencia mediante PostgreSQL.
- ❖ Archivo binarios (documentos, imágenes, etc) en directorio *.filestore*.



6.2 - ORM (Object Relational Mapping)

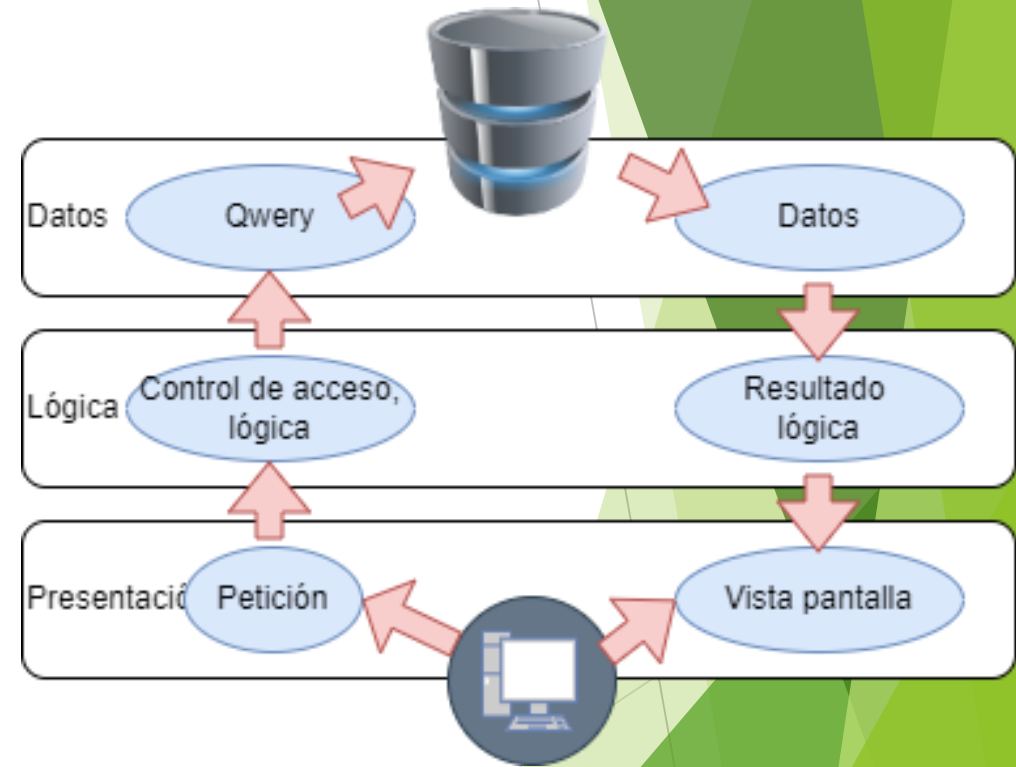
■ La capa de lógica

- ❖ Reside en el servidor de Odoo.
- ❖ Se encarga de la interacción con la capa de datos.
- ❖ En ella se encuentra el motor de asignación relacional de objetos (ORM).
- ❖ Realiza los cálculos y validaciones y devuelve los datos al cliente.



6.2 - ORM (Object Relational Mapping)

- **La capa de presentación**
 - ❖ Muestra los datos.
 - ❖ Permite la interacción con el usuario.
 - ❖ Interactúa con la API de ORM (RPC).
 - ❖ Odoo usa XML, JavaScript y Qweb.



6.2 - ORM (Object Relational Mapping)

- **ORM o Motor de asignación de objetos**
 - ❖ *Framework* para enlazar (mapear) la estructura de la BD con la de las entidades de la aplicación.
 - ❖ Simplifica el trabajo del programador.
 - ❖ Las consultas CRUD (Create, Read, Update, Delete) como objetos gestionables mediante lenguajes de programación.

6.2 - ORM (Object Relational Mapping)

- **ORM o Motor de asignación de objetos**
 - ❖ Evitamos escribir SQL.
 - ❖ Reducción de código de acceso a BD.
 - ❖ Puede generar código basura.
 - ❖ La modificación del modelo (p.e. añadir un campo) no implica la revisión de todas las sentencias INSERT, SELECT y UPDATE, se encarga el ORM.

6.2 - ORM (Object Relational Mapping)

- **ORM o Motor de asignación de objetos**
 - ❖ En algunos ORM hay que modificar archivos XML de vista.
 - ❖ 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 - ORM (Object Relational Mapping)

- **ORM o Motor de asignación de objetos**
 - ❖ Gracias a ORM el programador se abstrae de la BD.
 - ❖ En caso necesario, se podría cambiar el motor de BD sin mucha dificultad.

6.2 - ORM (Object Relational Mapping)

- ❖ Para generar un presupuesto en Odoo, se instancia un objeto presupuesto estableciendo su número de orden mediante el constructor.
- ❖ Las líneas de detalle se obtienen de la propiedad líneas del objeto presupuesto. Será un registro de la BD que mapea. Cada dato de un producto estará en la propiedad del modelo correspondiente y será un campo de la BD.
- ❖ De esta manera no se necesitan consultas, relaciones, claves foráneas...

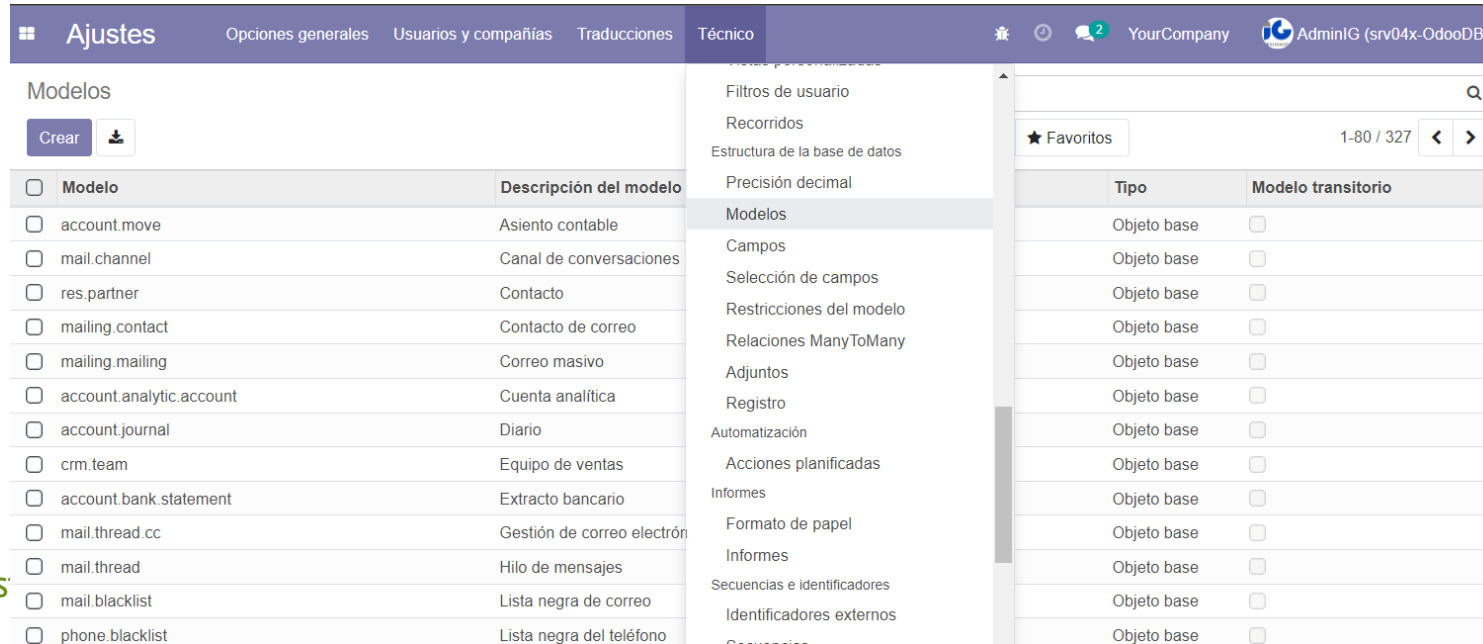
6.3 – MODELOS

6.3 - MODELOS

- En Odoo, cualquier dato es accesible a través de *modelos*, los cuales constituyen las tablas de la BBDD.
- Cada *modelo* es un objeto que contiene información en campos, de relación con otros objetos, así como la lógica de negocio.

6.3 - MODELOS

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



The screenshot displays the Odoo 'Ajustes' (Settings) interface, specifically the 'Técnico' (Technical) tab. The 'Modelos' (Models) section is active, showing a list of models and their descriptions. The interface includes a sidebar with a list of models and their descriptions, a central menu with various technical options, and a main table of models with columns for 'Tipo' (Type) and 'Modelo transitorio' (Transient Model).

Modelo	Descripción del modelo
<input type="checkbox"/> account.move	Asiento contable
<input type="checkbox"/> mail.channel	Canal de conversaciones
<input type="checkbox"/> res.partner	Contacto
<input type="checkbox"/> mailing.contact	Contacto de correo
<input type="checkbox"/> mailing.mailing	Correo masivo
<input type="checkbox"/> account.analytic.account	Cuenta analítica
<input type="checkbox"/> account.journal	Diario
<input type="checkbox"/> crm.team	Equipo de ventas
<input type="checkbox"/> account.bank.statement	Extracto bancario
<input type="checkbox"/> mail.thread.cc	Gestión de correo electrónico
<input type="checkbox"/> mail.thread	Hilo de mensajes
<input type="checkbox"/> mail.blacklist	Lista negra de correo
<input type="checkbox"/> phone.blacklist	Lista negra del teléfono

Tipo	Modelo transitorio
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>
Objeto base	<input type="checkbox"/>

6.3 - MODELOS

- Los modelos se pueden editar o crear desde cero.
- La pantalla que describe un modelo incluye campos, módulos en los que se usa, permisos de acceso y vistas asociadas entre otras.

The screenshot shows the 'Modelos / Contacto' configuration page in Odoo. The top navigation bar includes 'Ajustes' and various settings categories. The main content area is divided into sections for model description, fields, and permissions. The 'Campos' (Fields) tab is active, displaying a table of fields for the 'Contacto' model.

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
active	Activo	booleano	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
active_lang_count	Contador de idiomas activos	entero	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
activity_date_deadline	Siguiente plazo de 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

6.3 - MODELOS

- Son clases Python que corresponden a tablas Postgres.
- Todos extienden de la clase *models.Model*.
- Es el conjunto de campos y atributos que definen una entidad (cómo se almacenan, organizan y manipulan los datos).

6.3 - MODELOS

- Se nombran con un prefijo de la aplicación a la que pertenecen (en singular, aunque hay excepciones heredadas, p.e. res.users)

<input type="checkbox"/> Modelo	Descripción del modelo
<input type="checkbox"/> account.move	Asiento contable
<input type="checkbox"/> mail.channel	Canal de conversaciones
<input type="checkbox"/> res.partner	Contacto
<input type="checkbox"/> mailing.contact	Contacto de correo
<input type="checkbox"/> mailing.mailing	Correo masivo
<input type="checkbox"/> account.analytic.account	Cuenta analítica
<input type="checkbox"/> account.journal	Diario
<input type="checkbox"/> crm.team	Equipo de ventas
<input type="checkbox"/> account.bank.statement	Extracto bancario
<input type="checkbox"/> mail.thread.cc	Gestión de correo electrónico CC
<input type="checkbox"/> mail.thread	Hilo de mensajes
<input type="checkbox"/> mail.blacklist	Lista negra de correo
<input type="checkbox"/> phone.blacklist	Lista negra del teléfono

6.3 - MODELOS

- Existen tres tipos de modelos:
 - Modelo (*Model*). Estándar más común.
 - Transitorios (*TransientModel*). Basados en la clase `models.TransientModel` y utilizados para almacenar datos de forma temporal, se eliminan periódicamente.
 - Abstractos (*AbstractModel*). Basados en la clase `models.AbstractModel` y utilizados para definir clases abstractas que pueden ser heredadas por múltiples modelos.

6.3 - MODELOS

- En Ajustes / Técnico / Estructura de la base de datos / Modelos tenemos la lista de todos los modelos disponibles en la BBDD.
- Los modelos contienen campos con datos y atributos.

6.3 - MODELOS

- Se puede considerar los siguientes **tipos de campos principales**:
 - ❖ **Campos básicos**: numérico o texto (Integer, Boolean, ...)
 - ❖ **Campos reservados**: Creados por Odoo para uso por el sistema (id, create_date, create_uid, ...)
 - ❖ **Campos avanzados**: Binary, Html, Image, ...
 - ❖ **Campos calculados**: Su valor depende de una función y no de la BBDD. Declarado de forma estándar pero con argumento *compute* que referencia la función de cálculo definida en otra parte del código.
 - ❖ **Campos relacionales**: relaciones entre modelos. **Tres tipos.**

6.3 - MODELOS

- **Campos relacionales**
 - ❖ **Many2one:** Muchos a uno. Unidireccional. Selecciona un registro de otra tabla.
 - (varios empleados pertenecen a un departamento)
 - ❖ **Many2many:** Uno a muchos. Bidireccional.
 - (los clientes pueden comprar varios productos y cada producto puede ser adquirido por varios clientes)
 - ❖ **One2many:** Uno a muchos. Unidireccional. Uno o varios registros de otra tabla.
 - (un cliente ha efectuado varios pedidos)
- Un campo One2many debe tener un Many2one relacionado con él.

6.3 - MODELOS

- 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.

6.3 - MODELOS

- No hay una relación de las principales clases de Odoo. La mejor forma de conocerlas es usando pgAdmin.
- Lista de los modelos más utilizados



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 estado de instalación
ir.ui.view	ir_ui_view	Tipos de vistas
ir.ui.menú	ir_ui_menú	Elementos de menú
ir.actions.act_windows	ir_act_windows_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.access	ir_model_Access	Permisos de usuario
ir.sequence	ir_sequence	Campos que utilizan secuencias
ir.config_parameter	ir_config_parameter	Parámetros del sistema

6.3 - MODELOS

pgAdmin File Object Tools Help

Browser

- ir_act_window_view
- ir_actions
- ir_actions_todo
- ir_attachment
- ir_config_parameter
- ir_cron
- ir_default
- ir_demo
- ir_demo_failure
- ir_demo_failure_wizard
- ir_exports
- ir_exports_line
- ir_filters
- ir_logging
- ir_mail_server
- ir_model**
- ir_model_access
- ir_model_constraint
- ir_model_data
- ir_model_fields
- ir_model_fields_group_rel
- ir_model_fields_selection
- ir_model_relation
- ir_module_category
- ir_module_module
- ir_module_module_dependency
- ir_module_module_exclusion
- ir_property
- ir_rule
- ir_sequence
- ir_sequence_date_range

Query Editor

public.ir_model/srv04x-OdooDB/postgres@Ubuntu PostgreSQL 12

Query History

Data Output

	id	name	model	order	info	state	transient	create_uid
	[PK] Integer	character varying	character varying	character varying	text	character varying	boolean	integer
1		1 Base	base	id		base	false	
2		2 Unknown	_unknown	id		base	false	
3		3 Models	ir.model	is_mail_thread DESC, n...	Main s...	base	false	
4		4 Fields	ir.model.fields	name	Main s...	base	false	
5		5 Fields Selection	ir.model.fields.selection	sequence, id	Main s...	base	false	
6		6 Model Constraint	ir.model.constraint	id		base	false	
7		7 Relation Model	ir.model.relation	id		base	false	
8		8 Model Access	ir.model.access	model_id,group_id,nam...	Main s...	base	false	
9		9 Model Data	ir.model.data	module,model,name	Holds ...	base	false	
10		10 Create Menu Wizard	wizard.ir.model.menu.c...	id	Model ...	base	true	
11		11 Sequence	ir.sequence	name	Seque...	base	false	
12		12 Sequence Date Range	ir.sequence.date_range	id	Main s...	base	false	
13		13 Menu	ir.ui.menu	sequence,id	Main s...	base	false	

Explain Messages Notifications

Successfully run. Total query runtime: 175 msec.
327 rows affected.

6.3 - MODELOS

ACTIVIDADES PROPUESTAS.

- a) Modifica el modelo producto (*product.product*) añadiendo un campo “procedencia” que contenga uno de los continentes.



6.4 – INTERFACES DE USUARIO

6.4 - INTERFACES DE USUARIO

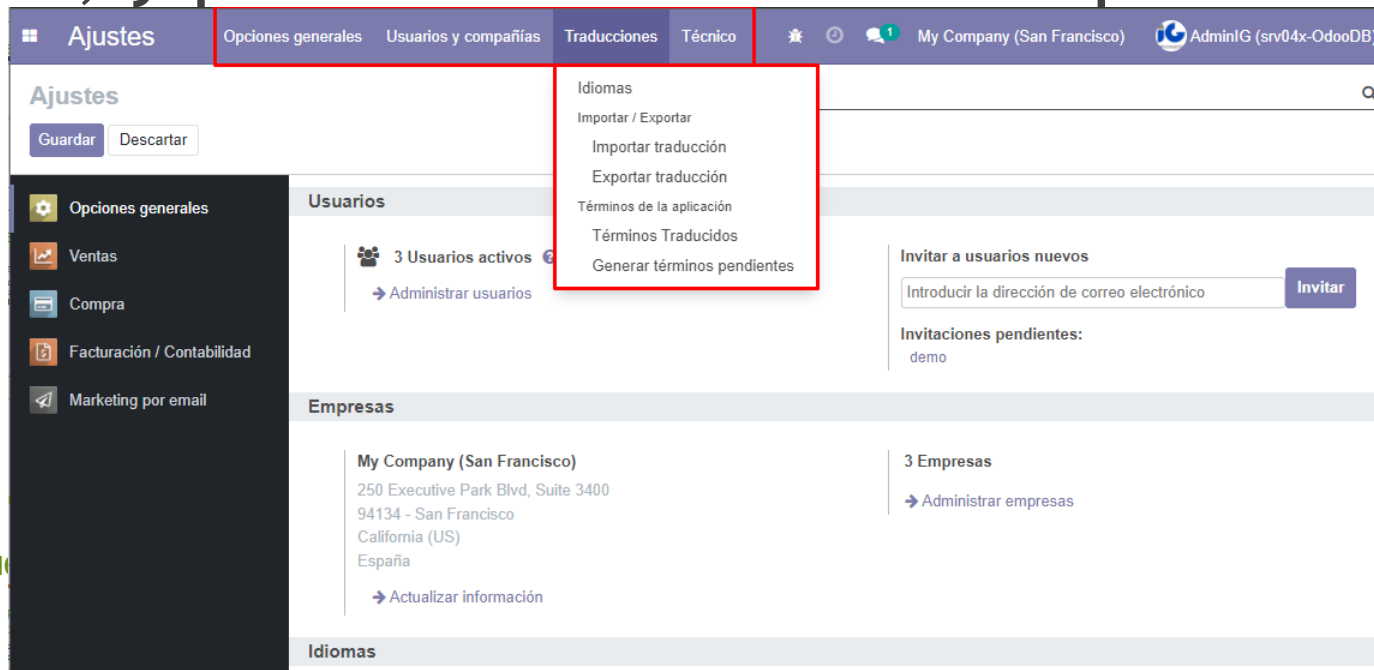
MENÚ Y OPCIONES DE MENÚ

- Un *menú* es un conjunto de opciones que se muestran en cualquier interfaz de Odoo.
- Las *opciones de menú* son los submenús de un menú.

6.4 - INTERFACES DE USUARIO

MENÚ Y OPCIONES DE MENÚ

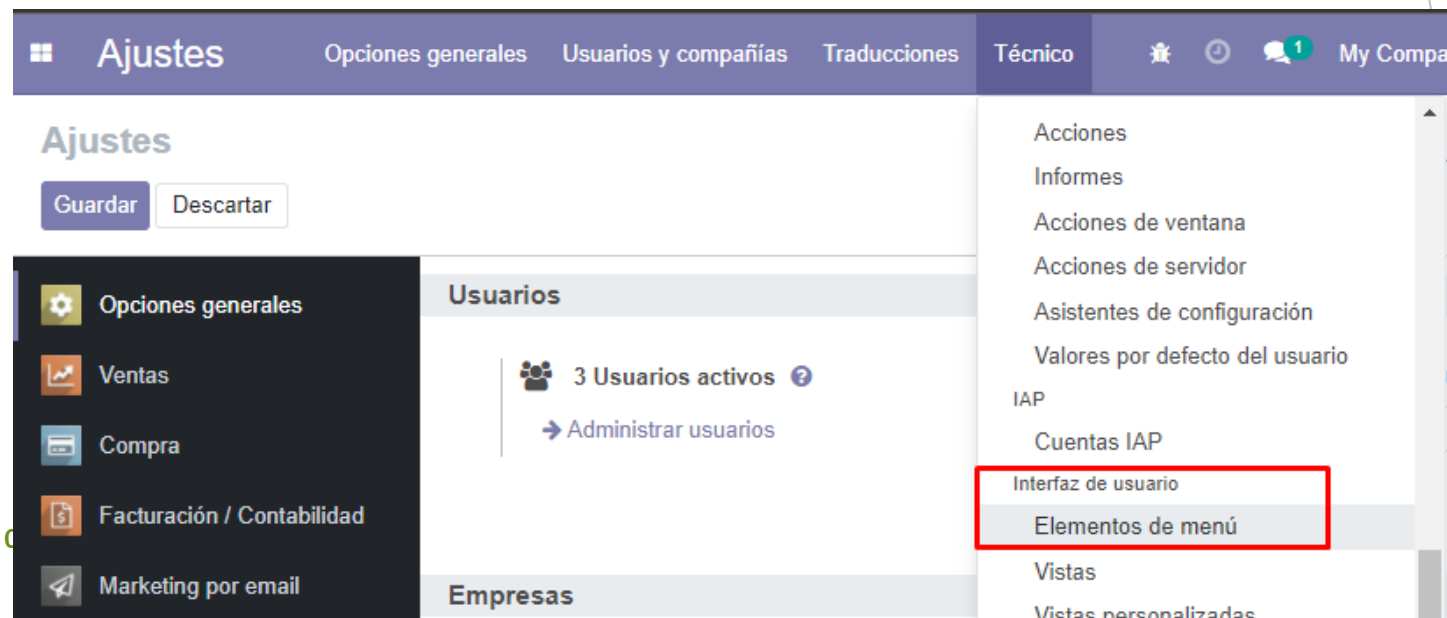
- Las *opciones de menú*, *las vistas*, *los menús* están internamente asignados a usuarios y grupos, y por lo tanto visibles solo por ellos.



6.4 - INTERFACES DE USUARIO

MENÚ Y OPCIONES DE MENÚ

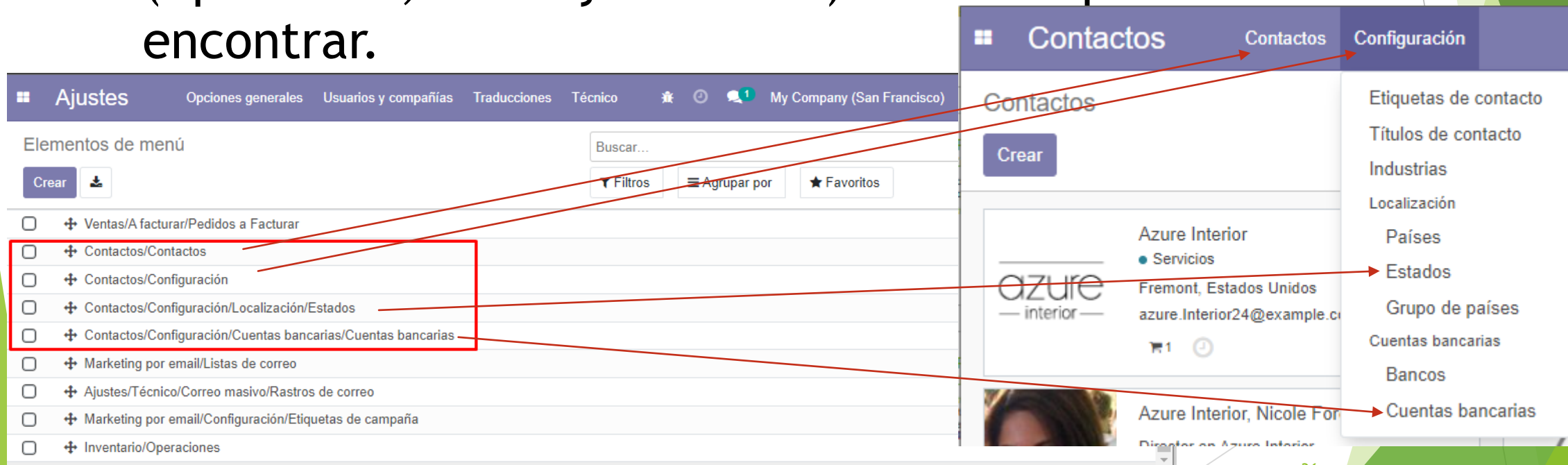
- Se almacenan en el modelo *ir.ui.menú*
- Las opciones de menú disponibles se encuentran en *Ajustes / Técnico / Interfaz de usuario / Elementos de menú*



6.4 - INTERFACES DE USUARIO

MENÚ Y OPCIONES DE MENÚ

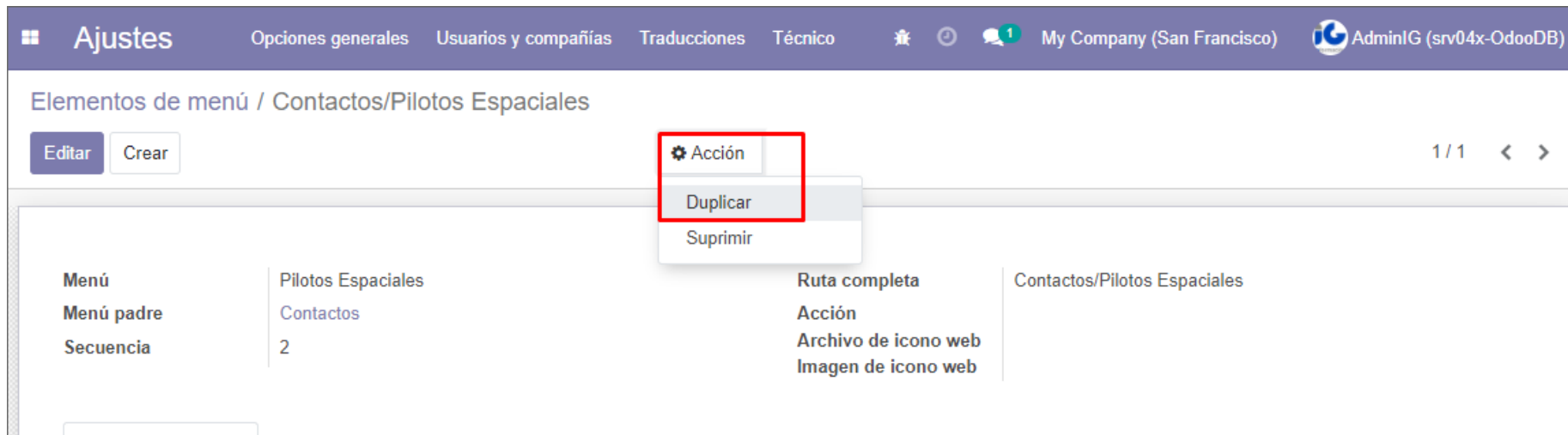
- Los nombres responden a una jerarquía de árbol (Aplicación, menú y submenú) donde se pueden encontrar.



6.4 - INTERFACES DE USUARIO

MENÚ Y OPCIONES DE MENÚ

- Se pueden crear nuevos (se aconseja duplicar) o modificar entrando en ellos.



6.4 - INTERFACES DE USUARIO

MENÚ Y OPCIONES DE MENÚ

- Si un elemento no está asignado a un grupo/usuario, será visible por todos.

Ajustes Opciones generales Usuarios y compañías Traducciones Técnico My Company (San Francisco) AdminIG (srv04x-OdooDB)

Elementos de menú / Ventas/Productos/Productos

Guardar Descartar 36 / 80 < >

Menú	Productos	Ruta completa	Ventas/Productos/Productos
Menú padre	Ventas/Productos	Acción	ir.actions.act_window Productos
Secuencia	1	Archivo de icono web	
		Imagen de icono web	Suba su archivo

Permisos de acceso Submenús

Nombre del grupo

Agregar línea

6.4 - INTERFACES DE USUARIO

MENÚ Y OPCIONES DE MENÚ

- Están relacionadas con las *vistas*.
- Editando y opción *Acción*

The screenshot displays the 'Ajustes' (Settings) section of the AdminIG application. The breadcrumb trail is 'Elementos de menú / Ventas/Productos/Productos'. The interface includes a top navigation bar with links like 'Opciones generales', 'Usuarios y compañías', 'Traducciones', and 'Técnico'. The main content area is divided into two tabs: 'Permisos de acceso' and 'Submenús'. The 'Submenús' tab is active, showing a form for configuring a menu item. The form includes fields for 'Menú' (set to 'Productos'), 'Menú padre' (set to 'Ventas/Productos'), and 'Secuencia' (set to '1'). The 'Ruta completa' field is set to 'Ventas/Productos/Productos'. The 'Acción' field is set to 'ir.actions.act_window', and the 'Archivo de icono web' field is set to 'Productos'. A red box highlights the 'Acción' field. Below the form, there is a 'Submenús' section with a 'Nombre del grupo' field and an 'Agregar línea' button.

Ajustes Opciones generales Usuarios y compañías Traducciones Técnico My Company (San Francisco) AdminIG (srv04x-OdooDB)

Elementos de menú / Ventas/Productos/Productos

Guardar Descartar 36 / 80

Menú Productos

Menú padre Ventas/Productos

Secuencia 1

Ruta completa Ventas/Productos/Productos

Acción ir.actions.act_window Productos

Archivo de icono web

Imagen de icono web Suba su archivo

Permisos de acceso Submenús

Nombre del grupo

Agregar línea

6.4 - INTERFACES DE USUARIO

MENÚ Y OPCIONES DE MENÚ

- *Vistas* relacionadas con el menú productos
- Editando y opción

Acción

Abrir: Acción

Nombre de acción: Productos

ID externo: sale.product_template_action

Objeto: product.template

Uso de la acción:

Tipo de acción: ir.actions.act_window

Ventana del destino: Ventana actual

Opciones Generales Seguridad

Vistas

Modo de vista: kanban,tree,form,activity

Ref. de la vista: Product.template.product.kanban

Ref. vista de búsqueda: product.template.search

Filtros

Valor del dominio:

Valor de contexto: {"search_default_filter_to_sell":1, "sale_m...

Límite: 80

Filtro: ☐

Ayuda

Crear un nuevo producto

Debe definir un producto para todo lo que vende o compra, ya sea un producto almacenable, un consumible o un servicio.

Guardar Descartar

6.4 - INTERFACES DE USUARIO

ACTIVIDADES PROPUESTAS.

- b) Duplica una opción de menú para que salga en otro elemento de menú distinto al suyo. Asígnala a un grupo concreto y comprueba las consecuencias de ello.



6.4 - INTERFACES DE USUARIO

VISTAS

- Una *Vista* es un diseño XML que determina cómo verá el usuario los registros de un modelo.
- Existen tres vistas básicas: Kanban, Árbol y Formulario.
- También hay otras vistas como , de búsqueda, de calendario, de Gantt, de Gráfico, de Actividad, etc.

6.4 - INTERFACES DE USUARIO

VISTAS

- Estructura básica común de una vista

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

6.4 - INTERFACES DE USUARIO

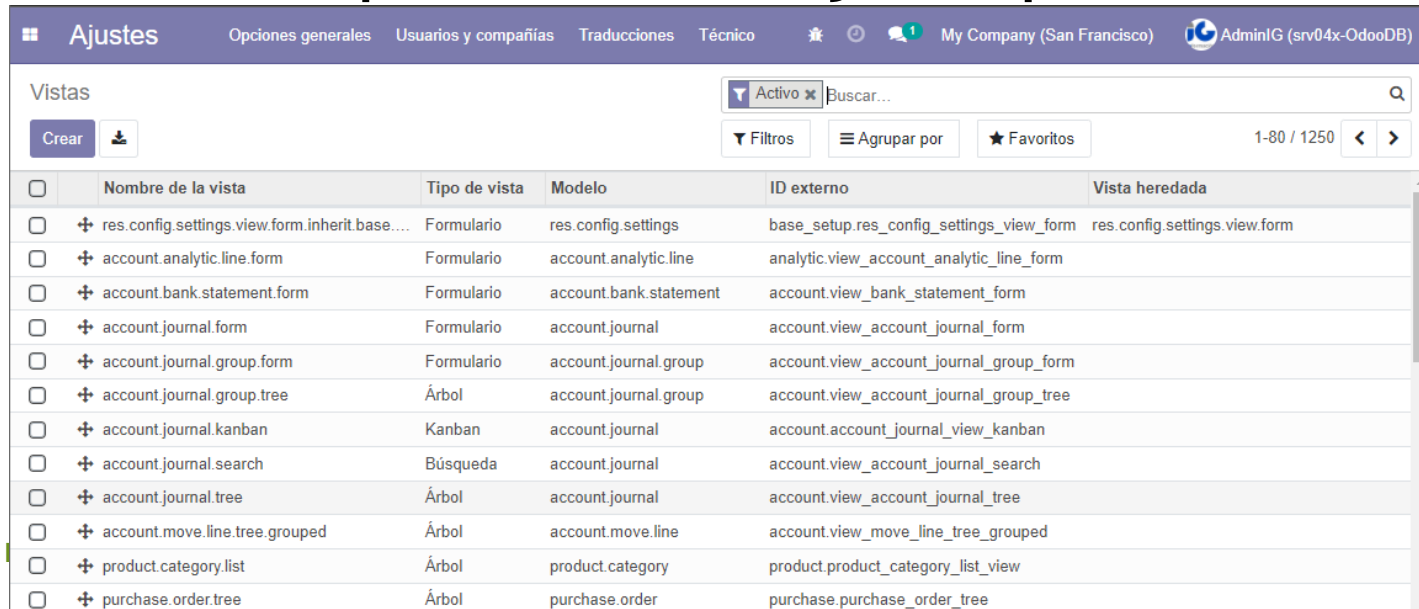
VISTAS

- El único campo obligatorio es el “name” que identifica la vista.
- Todas las pantallas de Odoo están constituidas 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 o más consultas a la BBDD.

6.4 - INTERFACES DE USUARIO

VISTAS

- En *Ajustes* / *Técnico* / *Interfaz de usuario* / *Vistas* se obtiene las vistas disponibles.
- Los nombres responden a una jerarquía de árbol.



<input type="checkbox"/>	Nombre de la vista	Tipo de vista	Modelo	ID externo	Vista heredada
<input type="checkbox"/>	+ res.config.settings.view.form.inherit base....	Formulario	res.config.settings	base_setup.res_config_settings_view_form	res.config.settings.view.form
<input type="checkbox"/>	+ account.analytic.line.form	Formulario	account.analytic.line	analytic.view_account_analytic_line_form	
<input type="checkbox"/>	+ account.bank.statement.form	Formulario	account.bank.statement	account.view_bank_statement_form	
<input type="checkbox"/>	+ account.journal.form	Formulario	account.journal	account.view_account_journal_form	
<input type="checkbox"/>	+ account.journal.group.form	Formulario	account.journal.group	account.view_account_journal_group_form	
<input type="checkbox"/>	+ account.journal.group.tree	Árbol	account.journal.group	account.view_account_journal_group_tree	
<input type="checkbox"/>	+ account.journal.kanban	Kanban	account.journal	account.account_journal_view_kanban	
<input type="checkbox"/>	+ account.journal.search	Búsqueda	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"/>	+ product.category.list	Árbol	product.category	product.product_category_list_view	
<input type="checkbox"/>	+ purchase.order.tree	Árbol	purchase.order	purchase.purchase_order_tree	

6.4 - INTERFACES DE USUARIO

VISTAS

- Los nombres tienen el formato *modelo.nombre_descriptivo.extensión*, siendo extensión el tipo de vista (kanban, tree, search, ...)
- Pulsando sobre ella accedemos a composición y configuración.

Ajustes Opciones generales Usuarios y compañías Traducciones Técnico My Company (San Francisco) AdminiG (srv04x-OdooDB)

Vistas / account.journal.search

Editar Crear Acción 8 / 80

Nombre de la vista	account.journal.search	Campo hijo	
Tipo de vista	Búsqueda	Vista heredada	
Modelo	account.journal	Modo de herencia de la vista	Vista base
Secuencia	1	Datos del modelo	account.journal.search
Activo	<input checked="" type="checkbox"/>	ID externo	account.view_account_journal_search

Estructura Permisos de acceso Vistas heredadas

Editar traducciones

```
<?xml version="1.0"?>
<search string="Search Account Journal">
  <field name="name" string="Journal" filter_domain="['|', ('name', 'ilike', self), ('code', 'ilike', self)]"/>
  <filter name="dashboard" string="Favorites" domain="[('show_on_dashboard', '=', True)]"/>
  <separator/>
  <filter name="sales" string="Sales" domain="[('type', '=', 'sale')]/>
  <filter name="purchases" string="Purchases" domain="[('type', '=', 'purchase')]/>
  <filter name="liquidity" string="Liquidity" domain="[('type', '=', 'cash'), ('type', '=', 'bank')]/>
  <filter name="miscellaneous" string="Miscellaneous" domain="[('type', 'not in', ['sale', 'purchase', 'cash', 'bank'])]/>
  <separator/>
  <filter name="inactive" string="Archived" domain="[('active', '=', False)]"/>
</search>
```

6.4 - INTERFACES DE USUARIO

VISTAS

- En la opción *Herramientas de desarrollo* se pueden editar las vistas para acceder al código xml.

Ajustes Opciones generales Usuarios y compañías Traducciones Técnico

Vistas / account.journal.search

Editar Crear

Nombre de la vista account.journal.search

Tipo de vista Búsqueda

Modelo account.journal

Secuencia 1

Activo ☒

Estructura Permisos de acceso Vistas heredadas

```
<?xml version="1.0"?>
<search string="Search Account Journal">
  <field name="name" string="Journal">
    <filter name="dashboard" string="Dashboard">
      <separator/>
    <filter name="sales" string="Sales">
      <filter name="purchases" string="Purchases">
        <filter name="liquidity" string="Liquidity">
          <filter name="miscellaneous" string="Miscellaneous">
            <separator/>
          <filter name="inactive" string="Inactive">
            <separator/>
          <separator/>
        <separator/>
      <separator/>
    <separator/>
  </field>
</search>
```

Correr los tests JS

Ejecutar pruebas de JS Mobile

Haga clic en Prueba en todas partes

Abrir vista

Comenzar recorrido

Editar Acción

Ver Campos

Gestionar Filtros

Traducción técnica

View Access Rights

View Record Rules

Establecer Predeterminados

Ver metadatos

Administrar mensajes

Administrar archivos adjuntos

Obtener Campos de Vista

Editar Vista: Formulario

Editar la vista del panel de control

Activar depuración de activos

Activar pruebas Depuración de activos

Regenerar paquetes de activos

Convertirse en superusuario

Abandonar las herramientas de desarrollo

6.4 - INTERFACES DE USUARIO

VISTAS

- Mediante *herencia de vistas* se pueden obtener nuevas vistas en base a vistas estándar.
- Se utiliza el *lenguaje xpath* para indicar las modificaciones, usando las marcas:

<xpath>

.../...

</xpath>

6.4 - INTERFACES DE USUARIO

VISTAS

- Por ejemplo, en la pantalla de la vista *product.template.product.form* podemos ver de dónde hereda (vista base) y las vistas heredadas (en su pestaña)

Nombre de la vista: product.template.product.form
Tipo de vista: Formulario
Modelo: product.template
Secuencia: 8
Activo: ☒

Campo hijo:
Vista heredada: product.template.common.form
Modo de herencia de la vista: Vista base
Datos del modelo: product.template.product.form
ID externo: product.product_template_only_form_view

Estructura Permisos de acceso Vistas heredadas

ID	Secuencia	Nombre de la vista ▼	ID externo
997	16	product.template.purchase.button.inherit	purchase.view_product_template_purchase_buttons_from
827	16	product.template.sale.order.button	sale.product_template_form_view_sale_order_button
1.099	16	product.template_procurement	stock.product_template_form_view_procurement_button

6.4 - INTERFACES DE USUARIO

VISTAS

- En el detalle de una *vista heredada* , por ejemplo el de *product.template_procurement*, podemos observar:

Abrir: Vistas que heredan de ésta

Nombre de la vista	product.template_procurement
Tipo de vista	Formulario
Modelo	product.template
Secuencia	16
Activo	<input checked="" type="checkbox"/>
Campo hijo	
Vista heredada	product.template.product.form
Modo de herencia de la vista	Vista de extensión
Datos del modelo	product.template_procurement
ID externo	stock.product_template_form_view_procurement_button

De qué vista hereda

Cómo hereda: Vista o Extensión

Estructura Permisos de acceso Vistas heredadas

```
<?xml version="1.0"?>
<data>
  <header position="inside">
    <button string="Update Quantity" type="object" groups="stock.group_stock_manager" name="action_update_quantity" context="{ 'default_product_tmpl_id': id }" groups="stock.group_stock_manager" />
    <button string="Replenish" type="action" name="366" context="{ 'default_product_tmpl_id': id }" groups="stock.group_stock_manager" />
  </header>
  <div name="button_box" position="inside">
    <button type="object" name="action_open_quants" attrs="{ 'invisible': [ ( 'type', '!=', 'product' ) ] }" class="oe_stat_button" icon="statinfo" />
  </div>
</data>
```

Abrir: Vistas que heredan de ésta

```
<field name="virtual_available" widget="statinfo" nolabel="1" class="mr4"/>
<field name="uom_name"/>
</span>
<span class="o_stat_text">Forecasted</span>
</div>
</button>
<button string="Product Moves" type="object" name="action_view_stock_move_lines" attrs="{ 'invisible': [ ( 'type', '!=', 'product' ) ] }" />
<div class="o_field_widget o_stat_info mr4">
  <span class="o_stat_text">Min:</span>
  <span class="o_stat_text">Max:</span>
</div>
<div class="o_field_widget o_stat_info">
  <span class="o_stat_value"><field name="reordering_min_qty"/></span>
  <span class="o_stat_value"><field name="reordering_max_qty"/></span>
</div>
</button>
<button type="object" name="action_view_orderpoints" attrs="{ 'invisible': [ ( 'type', '!=', 'product' ) ] }" />
  <field name="nbr_reordering_rules" widget="statinfo"/>
</button>
<button string="Lot/Serial Numbers" type="object" name="action_open_product_lot" attrs="{ 'invisible': [ ( 'track_id', '!=', 'product' ) ] }" />
<button string="Putaway Rules" type="object" name="action_view_related_putaway_rules" class="oe_stat_button" icon="statinfo" />
</div>
<!-- change attrs of fields added in view_template_property_form
to restrict the display for templates -->
<xpath expr="//group[@name='groupLotsAndWeight']" position="attributes">
  <attribute name="attrs">{'invisible': [ ( 'type', 'not in', [ 'product', 'consumable' ] ) ]}</attribute>
</xpath>
<xpath expr="//label[@for='weight']" position="before">
  <field name="responsible_id" domain="[ ( 'share', '=', False ) ]"/>
</xpath>
</data>
```

6.4 - INTERFACES DE USUARIO

ACTIVIDADES PROPUESTAS.

- c) Averigua las diferencias de vista base y vista de extensión en el modo de herencia de vista. Crea una vista heredada de la vista árbol de productos, quitando la imagen y añadiendo el precio de coste.



6.4 - INTERFACES DE USUARIO

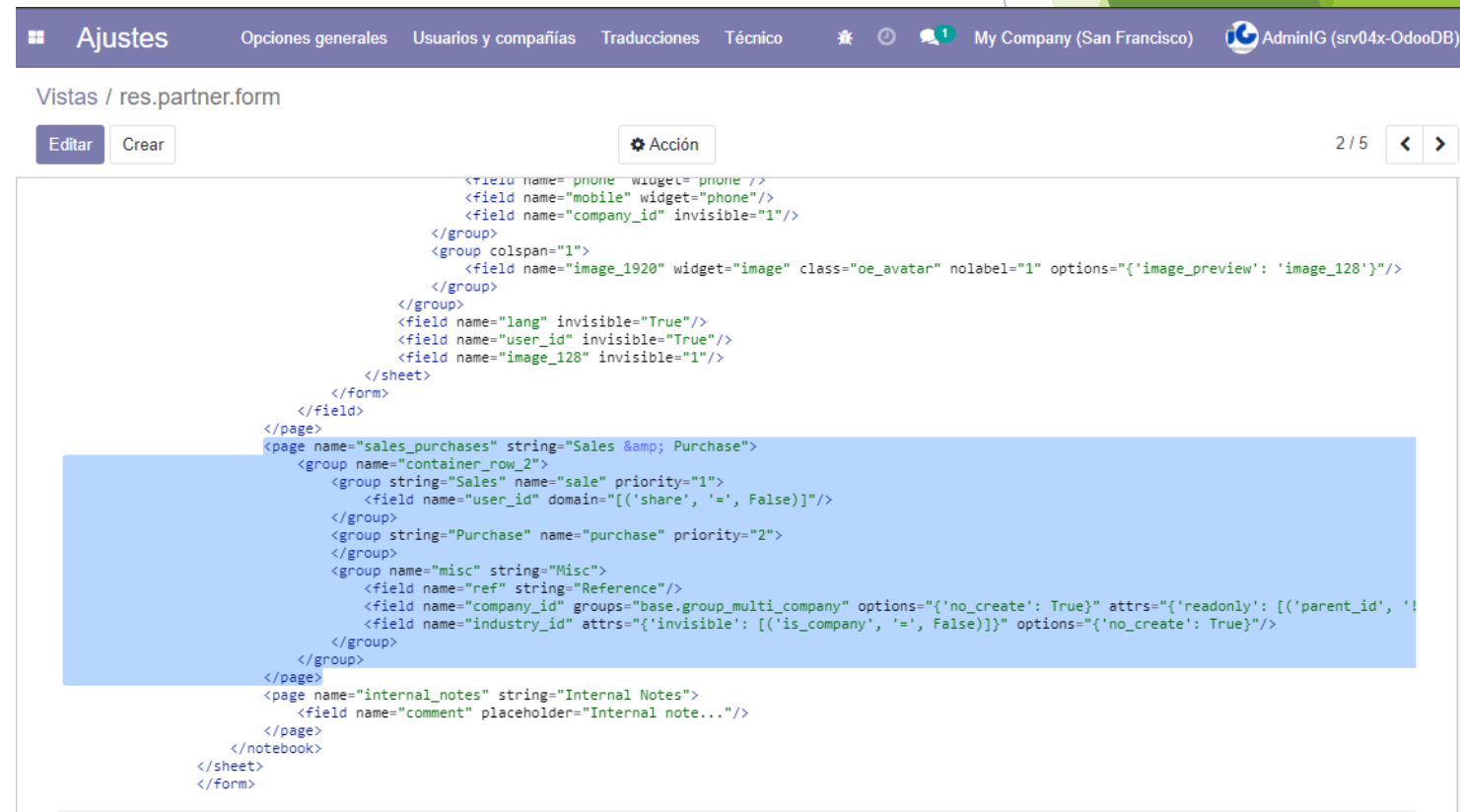
VISTAS

- Si un modelo tiene asociado distintos tipos de vista se usará por defecto el indicado con menor valor en el campo *Secuencia*.
- Salvo que la *Acción de ventana* asociada al elemento de menú sí especifique la vista, la cual tiene preferencia sobre la *Secuencia*.

6.4 - INTERFACES DE USUARIO

VISTAS

- Las *vistas complejas* son aquellas que tienen más de una solapa, es decir varias páginas.
- El código contenido entre `<page></page>` dibuja cada pestaña.



The screenshot shows the Odoo Vistas editor interface. At the top, there is a navigation bar with tabs: "Ajustes", "Opciones generales", "Usuarios y compañías", "Traducciones", and "Técnico". To the right of the tabs, it says "My Company (San Francisco)" and "AdminIG (srv04x-OdooDB)". Below the navigation bar, the title "Vistas / res.partner.form" is displayed. There are buttons for "Editar", "Crear", and "Acción". On the right side, there is a "2 / 5" indicator and navigation arrows. The main area displays the XML code for the form view. A blue highlight is placed over a specific section of the code, which is a page named "sales_purchases".

```
<field name="phone" widget="phone"/>
<field name="mobile" widget="phone"/>
<field name="company_id" invisible="1"/>
</group>
<group colspan="1">
  <field name="image_1920" widget="image" class="oe_avatar" nolabel="1" options="{ 'image_preview': 'image_128' }"/>
</group>
</group>
<field name="lang" invisible="True"/>
<field name="user_id" invisible="True"/>
<field name="image_128" invisible="1"/>
</sheet>
</form>
</field>
</page>
<page name="sales_purchases" string="Sales & Purchase">
  <group name="container_row_2">
    <group string="Sales" name="sale" priority="1">
      <field name="user_id" domain="[('share', '=', False)]"/>
    </group>
    <group string="Purchase" name="purchase" priority="2">
    </group>
    <group name="misc" string="Misc">
      <field name="ref" string="Reference"/>
      <field name="company_id" groups="base.group_multi_company" options="{ 'no_create': True }" attrs="{ 'readonly': [('parent_id', '!0)]"/>
      <field name="industry_id" attrs="{ 'invisible': [('is_company', '=', False)] }" options="{ 'no_create': True }"/>
    </group>
  </group>
</page>
<page name="internal_notes" string="Internal Notes">
  <field name="comment" placeholder="Internal note..." />
</page>
</notebook>
</sheet>
</form>
```

6.4 - INTERFACES DE USUARIO

ACTIVIDADES PROPUESTAS.

- d) Cambia el orden de las columnas que muestra una vista de lista cualquiera. Modifícala mediante la inclusión de una columna con el campo nuevo añadido al modelo de la actividad propuesta **6.4 a)**.



6.4 - INTERFACES DE USUARIO

ACCIONES DE VENTANA

- Las *acciones* son los métodos desde los que se van a abrir las vistas.
- Una *acción de ventana* o *acción de menú* contiene las instrucciones para construir la interfaz que se mostrará al pulsar un menú o botón.

6.4 - INTERFACES DE USUARIO

ACCIONES DE VENTANA

- Son archivos XML que contienen las instrucciones para construir la nueva vista.
- Las acciones de ventana se almacenan en el modelo *ir.actions.act_window* y también disponen de un procedimiento de herencia.
- Aparecen relacionadas con los *elementos de menú* y los botones de *vista*.

6.4 - INTERFACES DE USUARIO

ACCIONES DE VENTANA

- Los botones y *menús* tienen asociada una acción: lo que debe ocurrir al activarlos
- Forma parte del proceso de creación del *menú* o botón e indica el *modelo* y la *vista* con las que trabaja.
- Los tipos de acciones más importantes son:
 - ❖ Window: abre vista en nueva ventana.
 - ❖ Report: imprime un informe.
 - ❖ Wizard: ejecuta un asistente para un trabajo o proceso.

6.4 - INTERFACES DE USUARIO

ACCIONES DE VENTANA

- Para crear un acción de ventana que abra una vista, al menos se debe introducir:
 - ❖ Nombre, tipo, nombre de vista y tipo de vista.

Abrir: Acción

Nombre de acción	Facturas	Uso de la acción	
ID externo	account.action_move_out_invoice_type	Tipo de acción	ir.actions.act_window
Objeto	account.move	Ventana del destino	Ventana actual

Opciones Generales Seguridad

Vistas	Filtros
Modo de vista	Valor del dominio
Ref. de la vista	Valor de contexto
Ref. vista de búsqueda	Límite
	Filtro

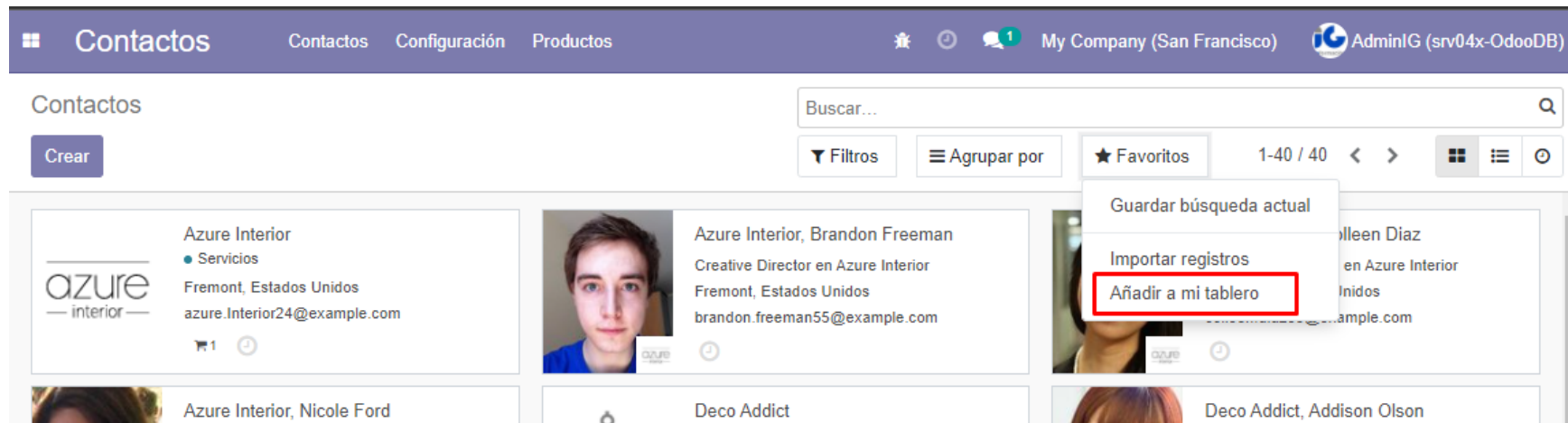
tree,kanban,form	[('move_type', '=', 'out_invoice')]
account.out.invoice.tree	{'default_move_type': 'out_invoice'}
account.invoice.select	80 58
	<input type="checkbox"/>

6.4 - INTERFACES DE USUARIO

TABLEROS



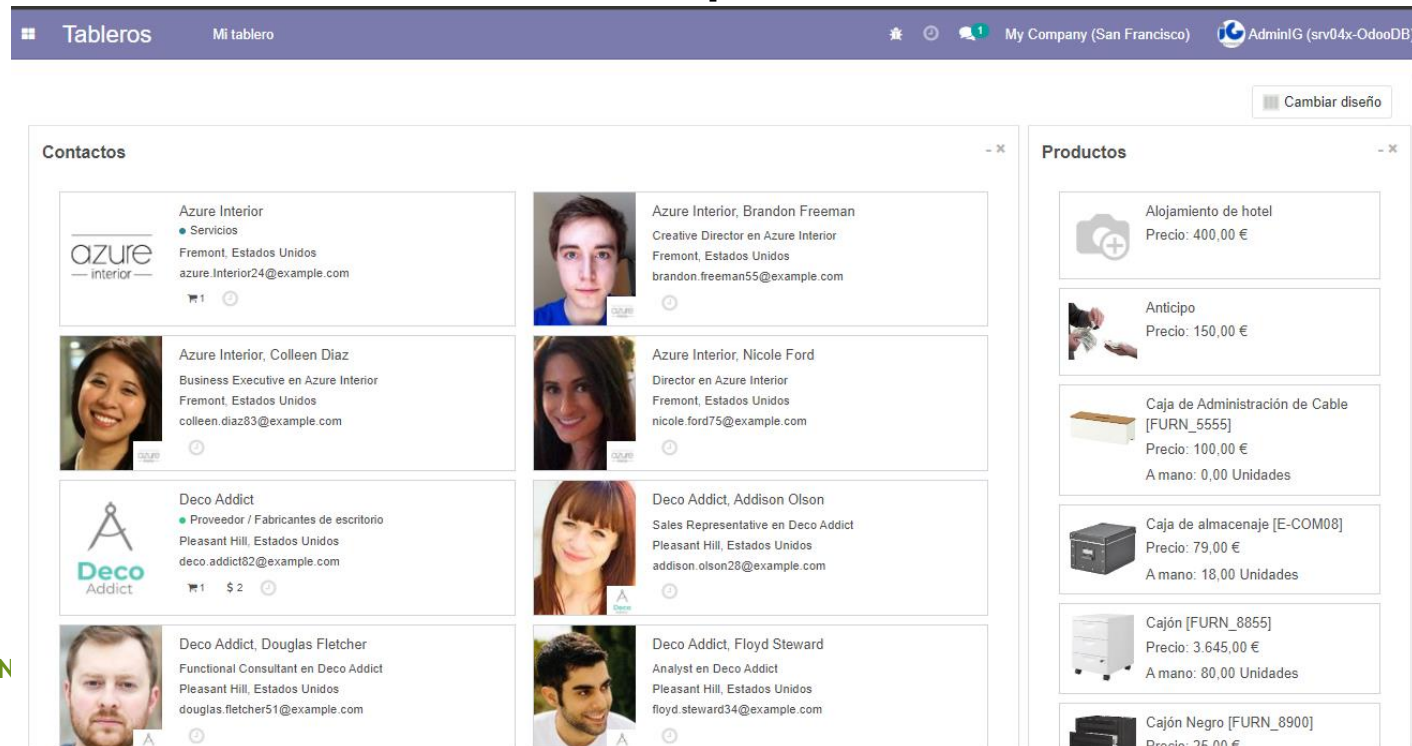
- Es un tipo especial de vista que contiene otras vistas.
- Es un cuadro de mandos que se puede diseñar con XML o su propio módulo llamado *Tableros*.



6.4 - INTERFACES DE USUARIO

TABLEROS

- Se trata de un resumen gráfico de procesos empresariales en una sola pantalla.



6.4 - INTERFACES DE USUARIO

ACTIVIDADES PROPUESTAS.

- e) Añade a Mi tablero tres vistas de tres aplicaciones diferentes.



6.4 - INTERFACES DE USUARIO

INFORMES Qweb

- Un informe es una vista que muestra información fija.
- Cada informe tiene asociado el nombre de una plantilla XML con su diseño.

The screenshot shows the 'Facturas' (Invoices) report configuration page in Odoo. The interface includes a top navigation bar with 'Ajustes' (Settings) and various options like 'Opciones generales', 'Usuarios y compañías', 'Traducciones', and 'Técnico'. The main content area is titled 'Informes / Facturas' and contains a table with report details. The table has columns for 'Nombre', 'Tipo de informe', 'Formato de papel', 'Nombre del modelo', 'Nombre de plantilla', and 'Nombre del informe impreso'. The 'Nombre de plantilla' column contains the value 'account.report_invoice_with_payments', which is highlighted with a red box. The 'Nombre del informe impreso' column contains the value '(object._get_report_base_filename())'. The 'Formato de papel' column contains the value 'PDF'. The 'Nombre' column contains the value 'Facturas'. The 'Tipo de informe' column contains the value 'Facturas'. The 'Formato de papel' column contains the value 'PDF'. The 'Nombre del modelo' column contains the value 'account.move'. The 'Nombre de plantilla' column contains the value 'account.report_invoice_with_payments'. The 'Nombre del informe impreso' column contains the value '(object._get_report_base_filename())'. The interface also includes a sidebar with 'Seguridad' and 'Propiedades avanzadas' tabs, and a bottom section with 'Nombre del grupo', 'Contabilidad / Facturación', and 'Técnico / Mostrar funciones de contabilidad: solo lectura'.

Nombre	Tipo de informe	Formato de papel	Nombre del modelo	Nombre de plantilla	Nombre del informe impreso
Facturas	Facturas	PDF	account.move	account.report_invoice_with_payments	(object._get_report_base_filename())

6.4 - INTERFACES DE USUARIO

INFORMES Qweb

- Odoo cuenta con el lenguaje de plantillas *Qweb* que permite realizar modificaciones en informes o crear nuevos.
- La vista de tipo Qweb que los contiene por defecto es *web.html_container*.
- Los nombres internos de todos los informes empiezan por *report_*

6.4 - INTERFACES DE USUARIO

INFORMES Qweb

- En *Ajustes / Técnico / Acciones / Informes* tenemos la relación de informes donde podremos consultarlos.

<input type="checkbox"/>	Solicitud de presupuesto	purchase.order	ir.actions.report	purchase.report_purchasequotation PDF
<input type="checkbox"/>	Operaciones de albarán	stock.picking	ir.actions.report	stock.report_picking PDF
<input type="checkbox"/>	Albarán	stock.picking	ir.actions.report	stock.report_deliveryslip PDF
<input type="checkbox"/>	Hoja de recuento	stock.inventory	ir.actions.report	stock.report_inventory PDF
<input type="checkbox"/>	Código de barras del paquete con conteni...	stock.quant.package	ir.actions.report	stock.report_package_barcode PDF
<input type="checkbox"/>	Código de barras del paquete (PDF)	stock.quant.package	ir.actions.report	stock.report_package_barcode_small PDF

6.4 - INTERFACES DE USUARIO

INFORMES Qweb

- En la opción *Vistas Qweb* se obtiene la relación de vistas que utiliza.

The image shows two screenshots of the Odoo user interface. The top screenshot shows the 'Ajustes' (Settings) menu with 'Vistas QWeb' highlighted. A red arrow points from this option to the bottom screenshot. The bottom screenshot shows the 'Vistas QWeb' view, which is a table listing the QWeb views used by the system.

Nombre	Nombre del modelo
Hoja de recuento	stock.inventory
Tipo de informe	stock.report_inventory
Formato de papel	'Inventario- %s' % (object.name)

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

6.4 - INTERFACES DE USUARIO

INFORMES Qweb

- En la solapa *Estructura*, el código HTML que forma la plantilla Qweb.
- El código HTML es reutilizable en otras plantillas.

The screenshot shows the 'Ajustes' (Settings) interface for a Qweb report. The breadcrumb trail is 'Informes / Hoja de recuento / Vistas / report_inventory'. The interface has a top navigation bar with 'Ajustes', 'Opciones generales', 'Usuarios y compañías', 'Traducciones', and 'Técnico'. The user is logged in as 'My Company (San Francisco)' with 'AdminIG (srv04x-OdoDB)'. Below the breadcrumb, there are buttons for 'Editar', 'Crear', and 'Acción'. The main content area is divided into two sections. The left section contains metadata for the report: 'Nombre de la vista' (report_inventory), 'Tipo de vista' (QWeb), 'Modelo' (16), 'Secuencia' (16), and 'Activo' (checked). The right section contains 'Campo hijo' (Vista heredada), 'Modo de herencia de la vista' (Vista base), 'Datos del modelo' (report_inventory), and 'ID externo' (stock.report_inventory). Below this, there are three tabs: 'Estructura' (selected), 'Permisos de acceso', and 'Vistas heredadas'. The 'Estructura' tab displays XML code for the report template. The code starts with a root element 'stock.report_inventory' and includes a 'web.html_container' and 'web.external_layout' sections. The 'web.html_container' section contains a 'page' div with a 'table' element. The 'web.external_layout' section contains a 'table' element with a 'thead' and 'tbody'. The 'tbody' contains three rows: 'Location', 'Product', and 'Lot/Serial Number'. The 'table' element is styled with 'table-sm' and 'table-bordered' classes. The code ends with a closing tag for the root element. The page number '66' is visible in the bottom right corner.

```
<?xml version="1.0"?>
<t t-name="stock.report_inventory">
  <t t-call="web.html_container">
    <t t-foreach="docs" t-as="o">
      <t t-call="web.external_layout">
        <div class="page">
          <br/>
          <h2 t-field="o.name"/>
          <div class="row mt32 mb32" id="informations">
            <div t-if="o.date" class="col-auto mw-100 mb-2">
              <strong>Date:</strong>
              <p class="m-0" t-field="o.date"/>
            </div>
          </div>
          <t t-set="locations" t-value="o.line_ids.mapped('location_id')"/>
          <table class="table table-sm">
            <thead>
              <tr>
                <th groups="stock.group_stock_multi_locations"><strong>Location</strong></th>
                <th><strong>Product</strong></th>
                <th groups="stock.group_production_lot"><strong>Lot/Serial Number</strong></th>
                <th groups="stock.group_tracking_lot"><strong>Package</strong></th>
              </tr>
            </thead>
          </table>
        </div>
      </t>
    </t>
  </t>
</t>
```

6.4 - INTERFACES DE USUARIO

SERIALIZACIÓN e identificadores externos

- Se llama *serialización* al proceso de convertir un objeto en una secuencia de bytes para almacenarlo o transmitirlo a la memoria, una BBDD o un archivo.
- Las interfaces de usuario vistas hasta ahora se encuentran en registros de la BD de Odoo que usan la *serialización* para exportar e importar datos .
- Cada registro de una tabla de Odoo tiene un identificador único llamado “id”.

6.4 - INTERFACES DE USUARIO

SERIALIZACIÓN e identificadores externos

- Para poder hacer referencia al registro relacionado con un id que aún no está creado, se utilizan los *identificadores externos*.
- Los *identificadores externos* asignan identificadores con nombre a los registros,
- Usan el modelo *ir.model.data*, donde se tiene la relación de los id externos con nombre, con los id numéricos de la BBDD.

6.4 - INTERFACES DE USUARIO

SERIALIZACIÓN e identificadores externos

- En *Ajustes / Técnico / Secuencias* e *Identificadores* externos tenemos entre otras la opción de crear.
- *Módulo* e *Identificador externo* concatenados con un punto deben coincidir con el nombre de la vista.
- *Nombre del modelo* se obtiene de la url (ir.ui.view)

Ajustes Opciones generales Usuarios y compañías Traducciones Técnico

My Company (San Francisco) AdminIG (srv04x-OdooDB)

Identificadores externos / Nuevo

Guardar Descartar

Módulo

Identificador externo

No actualizable ☐

Última actualización el

Creado el

Nombre mostrado

Nombre del modelo

ID de registro

Registro

6.4 - INTERFACES DE USUARIO

SERIALIZACIÓN e identificadores externos

- Añadiendo estos datos, se actualizará el campo id externo de la vista utilizada para el informe.
- También se usan para los procesos de importación de datos, evitando duplicidades.

6.4 - INTERFACES DE USUARIO

SECUENCIAS O CONTADORES

- Se trata de un contador secuencial automático.
- Permite numerar cualquier entidad.
- Se componen de prefijo, relleno, contador y sufijo.

Ajustes Opciones generales Usuarios y compañías Traducciones Técnico My Company (San Francisco) AdminIG (srv04x-OdooDB)

Secuencias / IGFormación Secuencia de albarán

Editar Crear Acción 28 / 28

Nombre	IGFormación Secuencia de albarán	Código de secuencia	Activo
Implementación	Estándar	Compañía	My Company (San Francisco)

Secuencia

Prefijo	IGF/PICK/	Tamaño de secuencia	5
Sufijo		Paso	1
Utilizar subsecuencias por date_range	<input type="checkbox"/>	Próximo número	1

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: %(h12)s
Mes: %(month)s	Día de la semana (0: Lunes): %(weekday)s	Minutos: %(min)s
Día: %(day)s		Segundos: %(sec)s

6.4 - INTERFACES DE USUARIO

SECUENCIAS O CONTADORES

- Odoo crea secuencias para casi todos los formularios y se pueden modificar según necesidad de la empresa.
- El modelo `ir.sequence` tiene asociadas las siguientes vistas.

Ajustes Opciones generales Usuarios y compañías Traducciones Técnico My Company (San Francisco) AdminIG (srv04x-OdooDB)

Modelos / Secuencia

Editar Crear Imprimir Acción 2 / 2 < >

Descripción del modelo	Secuencia	Tipo	Objeto base
Modelo	ir.sequence	En las aplicaciones	base
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**

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	

6.4 - INTERFACES DE USUARIO

SECUENCIAS O CONTADORES

- Haciendo clic en la vista formulario vemos su estructura

Abrir: Vistas

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

Estructura de la vista

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

- Field: arch_base
- Objeto: ir.ui.view
- Tipo: text
- Widget: ace
- Modificadores: {"required":true}

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="Sequence" name="sequence">
        <group>
          <group>
            <field name="prefix"/>
            <field name="suffix"/>
            <field name="use_date_range"/>
          </group>
        </group>
      </page>
    </notebook>
  </sheet>
</form>
```

Cerrar

6.4 - INTERFACES DE USUARIO

SECUENCIAS O CONTADORES

- Una secuencia puede estar asociada al nombre del registro o formar parte de su información interna.

Por ejemplo:

`<field name="prefix">JJ</field>`

`<field name="padding">4</field>`

- Generará un contador que empezará por JJ0001 (un prefijo JJ que aparecerá en todos los nombres y un número secuencial de 4 dígitos.

6.4 - INTERFACES DE USUARIO

DOCUMENTOS Y PLANTILLAS

- Odoo dispone de plantillas y documentos comerciales editables según las preferencias de la empresa.
- Se accede a través de *Ajustes / Técnico / Plantillas*

Ajustes						
Opciones generales Usuarios y compañías Traducciones Técnico						
My Company (San Francisco) AdminIG (srv04x-OdooDB)						
Plantillas						
Buscar...						
Filtros Agrupar por Favoritos 1-14 / 14						
<input type="checkbox"/>	Nombre	Aplica a	Asunto	Desde	Para (correos electrónicos)	Fichero del informe
<input type="checkbox"/>	Auth Signup: Odoo Account C...	Usuarios	Bienvenido a \${object.compan...	"\${object.company_id.name ...	\${object.email_formatted safe}	
<input type="checkbox"/>	Auth Signup: Odoo Connection	Usuarios	\${object.create_uid.name} de ...	"\${object.company_id.name ...	\${object.email_formatted safe}	
<input type="checkbox"/>	Auth Signup: Reset Password	Usuarios	Restablecer contraseña	"\${object.company_id.name ...	\${object.email_formatted safe}	
<input type="checkbox"/>	Auth Signup: Unregistered Us...	Usuarios	Recordatorio para usuarios no...	\${object.company_id.partner_i...	\${object.email_formatted safe}	
<input type="checkbox"/>	Delivery: Send by Email	Albarán	\${object.company_id.name} O...			\${object.partner_id.... \${((object.name or "").replace("f...
<input type="checkbox"/>	Invoice: Send by email	Asiento contable	\${object.company_id.name} F...	\${((object.invoice_user_id.emai...		\${object.partner_id.... Factura_\${((object.name or "").repl...
<input type="checkbox"/>	Partner Mass Mail	Contacto				\${object.id}
<input type="checkbox"/>	Payment Receipt: Send by email	Pagos	\${object.company_id.name} R...			\${object.partner_id.... \${((object.name or "").replace("f...
<input type="checkbox"/>	Portal: new user	Configuración usuario portal	Su cuenta de Odoo en \${objec...			\${object.user_id.email_format...
<input type="checkbox"/>	Purchase Order: Send PO	Pedido de compra	\${object.company_id.name} P...			\${object.partner_id.... PC_\${((object.name or "").repla...
<input type="checkbox"/>	Purchase Order: Send RFQ	Pedido de compra	\${object.company_id.name} P...			\${object.partner_id.... PdO_\${((object.name or "").repl...
<input type="checkbox"/>	Purchase Order: Vendor Remi...	Pedido de compra	\${object.company_id.name} P...	\${((object.user_id.email_format...		\${object.partner_id.... PC_\${((object.name or "").repla...

6.4 - INTERFACES DE USUARIO

DOCUMENTOS Y PLANTILLAS

- Es necesario indicar el objeto del que se va a extraer la información usando campos personalizados, por ejemplo *`${object.company_id.name}`*

The screenshot shows the Odoo user interface for editing an email template. The top navigation bar includes 'Ajustes' (Settings) and various configuration options. The main header indicates the current template is 'Plantillas / Invoice: Send by email'. Below this, there are buttons for 'Editar' (Edit), 'Crear' (Create), and 'Acción' (Action). The main content area is titled 'Invoice: Send by email' and shows the template content with several tabs: 'Contenido' (Content), 'Configuración del correo electrónico' (Email configuration), 'Configuración avanzada' (Advanced configuration), and 'Generador dinámico de expresiones con marcadores de posición' (Dynamic expression generator with placeholders). The 'Contenido' tab is active, displaying the following text: 'Asunto `${object.company_id.name}` Factura (Ref `${object.name or 'n/a'}`)', 'Estimado % if object.partner_id.parent_id: `${object.partner_id.parent_id.name}`, % else: `${object.partner_id.name}`, % endif', 'Aquí está su % if object.name: factura `${object.name}` % else: factura %endif % if object.invoice_origin: (con referencia: `${object.invoice_origin}`) % endif por el monto de `${format_amount(object.amount_total, object.currency_id)}` de `${object.company_id.name}`, % if object.payment_state == 'paid': Esta factura ya está pagada. % else: Por favor remita el pago a su más pronta conveniencia. % if object.payment_reference: Utilice la siguiente referencia para su pago: `${object.payment_reference}`, % endif % endif', and 'No dude en contactarnos si tiene alguna pregunta. % if object.invoice_user_id.signature: `${object.invoice_user_id.signature | safe}` % endif'.

6.4 - INTERFACES DE USUARIO

DOCUMENTOS Y PLANTILLAS

- No hay que olvidar que una vez configurada hay que asociarle una acción de servidor que indique cómo y cuando se utilizará la plantilla.

Ajustes Opciones generales Usuarios y compañías Traducciones Técnico My Company (San Francisco) AdminIG (srv04x-OdooDB)

Acciones de servidor / Nuevo

Guardar Descartar

Crear acción contextual

Nombre de acción
Por ejemplo, Actualizar la cantidad de pedido

Modelo Actualizar el registro

Datos para escribir Seguridad

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

Mapeo de valores

- Field: fields_lines
- Objeto: ir.actions.server
- Tipo: one2many
- Contexto: {}
- Dominio: []
- Relación: ir.server.object.lines

Campo	Tipo de evaluación	Registro	Valor
Agregar línea			

RESUMEN



- ✓ Odoo ERP está desarrollado en Python y utiliza PostgreSQL.
- ✓ Con XML y Qweb genera distintas interfaces de usuario e informes.
- ✓ Los cuatro conceptos básicos, objetos o entidades de la estructura de Odoo son: Modelo, Vista, Menú y Acción.
- ✓ La herencia es la propiedad mediante la que un objeto deriva de otro y extiende su funcionalidad.

RESUMEN



- ✓ ORM realiza un mapeo entre las entidades de la base de datos.
- ✓ Con ORM el programador se puede abstraer de la base de datos y evitar la complicación de sus sentencias.
- ✓ Modelo son clases Python que se corresponden con tablas PostgreSQL.
- ✓ Existen tres tipos: Estándar, Transitorios y Abstractos.

RESUMEN



- ✓ Los principales tipos de campos de los modelos son: básicos, reservados, avanzados, calculados y relacionales.
- ✓ Los campos relacionales pueden ser Many2one, Many2many y One2many.
- ✓ Algunos de los atributos más usados para definir los campos son: string, default, required, help, readonly, compute, store.
- ✓ Las interfaces de usuario que utiliza Odoo son los menús y las vistas con sus acciones de ventana, los tableros, los informes.

RESUMEN



- ✓ Las opciones de menú que aparecen en cada vista se almacenan en el modelo *ir.ui.menu*.
- ✓ Si no están asignado a un grupo en concreto, son visibles por todo los usuarios.
- ✓ Todas las pantallas están construidas como una vista.
- ✓ Los principales tipos de vista son: Formulario, Lista, Kanban y Búsqueda.

RESUMEN



- ✓ También existen otros tipos: Qweb, Gantt, Gráfico, Calendario, Pivote, Cohorte, Tablero.
- ✓ Las Acciones son archivos XML que contienen instrucciones que se ejecutan al activar un botón o elemento de menú.
- ✓ Los Tableros (dashboard) son vistas especiales que contienen otras vistas encajadas.
- ✓ Un informe es una vista, no modificable una vez creada, que proporciona datos de las tablas obtenidos mediante aplicación de uno o varios criterios.

RESUMEN



- ✓ Qweb es un motor de plantillas utilizado también en la generación de informes.
- ✓ Una secuencia es un contador secuencial. Obedecen a un formato prefijado y se pueden modificar.
- ✓ Odoo dispone de una gran variedad de plantillas que permiten la personalización de campos.

Sistemas de Gestión Empresarial

FIN DE LA UNIDAD