

Práctica 4 - Desarrollo de módulos de Odoo: Modelo y Vista

Sistemas de Gestión Empresarial

Cristian Fernández

15 de diciembre de 2025

Índice

1. Introducción	2
2. Actividad 01 - Lista de tareas	3
3. Actividad 02 - Biblioteca de cómics	10
4. Actividad 03 - Hospital	12
5. Actividad 04 - Instituto	14

1. Introducción

El objetivo de esta práctica es aplicar los conocimientos adquiridos en ejercicios anteriores sobre modelos, relaciones y vistas en el desarrollo de aplicaciones. Para ello, nos ocuparemos de la implementación de cuatro actividades que abarcan diferentes contextos, como la gestión de tareas, bibliotecas de cómics, pacientes y médicos, y ciclos formativos.

Habrá que tener en cuenta la revisión de los ejemplos proporcionados en el siguiente repositorio. <https://github.com/sergarb1/OdooModulosEjemplos>

Una vez que verifiquemos y probemos el funcionamiento de cada ejemplo en nuestro contenedor de Odoo debemos:

- Revisar los manifests.
- Revisar los modelos de datos.
- Revisar las vistas.

Entender y probar estos ejemplos nos facilitará la comprensión sobre la arquitectura de Odoo y su sistema de modular y, por tanto, la realización la práctica.

Se reutilizará el repositorio creado en las prácticas anteriores, con objeto de tener todo el trabajo bien organizado y disponible para una posible visualización en cualquier momento.

Cada actividad tendrá una explicación breve del proceso de creación y se demostrará su funcionamiento mediante capturas. Si fuese necesario también se comentarán los problemas encontrados y las soluciones empleadas.

2. Actividad 01 - Lista de tareas

En esta primera actividad modificaremos el módulo de la práctica anterior ‘`lista_tareas`’ para que:

- Tenga una nueva vista para ver las tareas en formato **Kanban**.
- Modificar las tareas para que tengan una fecha asignada (implementado en la práctica anterior) y crear una nueva vista para visualizarlas en un formato **Calendario** según esa fecha.

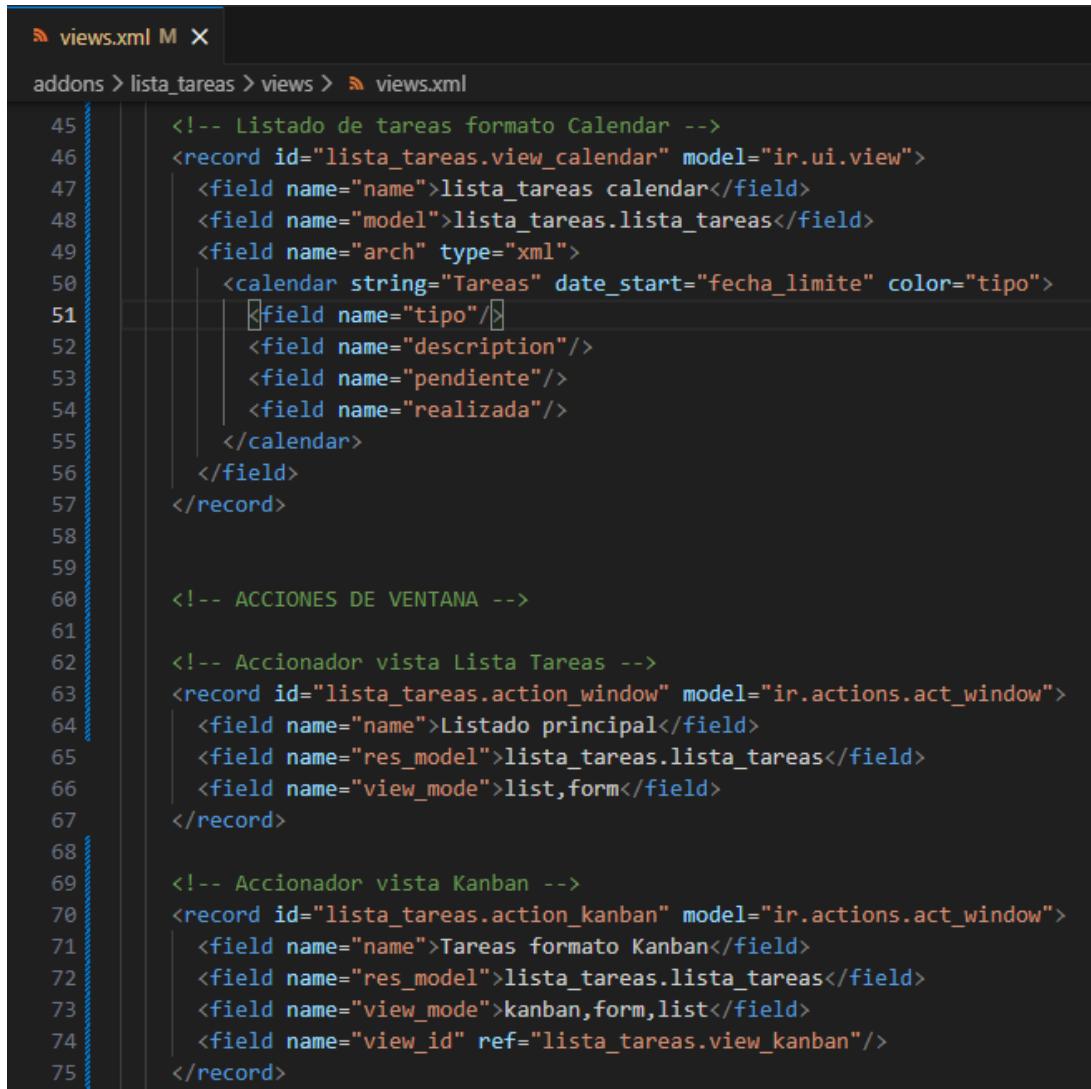
Las primera tanda de capturas servirán para mostrar el código creado/modificado e intentarán ser lo suficientemente autoexplicativas para una correcta comprensión del mismo.

Echémosle un vistazo a la vista principal del módulo.

The screenshot shows a code editor with a dark theme, displaying an XML file named 'views.xml'. The file is located in the 'addons > lista_tareas > views' directory. The code defines two records: one for a main list view and one for a Kanban view.

```
<odoo>
    <data>
        <!-- DEFINICIÓN DE VISTAS -->
        <!-- Listado de tareas principal -->
        <record model="ir.ui.view" id="lista_tareas.list">
            <field name="name">lista_tareas list</field>
            <field name="model">lista_tareas.lista_tareas</field>
            <field name="arch" type="xml">
                <list>
                    <field name="tipo"/>
                    <field name="fecha_limite"/>
                    <field name="pendiente"/>
                    <field name="realizada"/>
                    <field name="fallida"/>
                </list>
            </field>
        </record>
        <!-- Listado de tareas formato Kanban -->
        <record id="lista_tareas.view_kanban" model="ir.ui.view">
            <field name="name">lista_tareas kanban</field>
            <field name="model">lista_tareas.lista_tareas</field>
            <field name="arch" type="xml">
                <kanban class="o_kanban_mobile" sample="1" quick_create="false">
                    <progressbar field="activity_state"
                        colors='{"planned": "success", "today": "warning", "overdue": "danger"}'/>
                    <templates>
                        <t t-name="card">
                            <div class="d-flex mb-2" style="justify-content: space-between;">
                                <field name="tipo" class="fw-bolder fs-5" />
                                <field name="fecha_limite" class="ms-1 text-muted fs-5"/>
                            </div>
                            <footer>
                                <div class="d-flex text-muted">
                                    <field name="description"/>
                                </div>
                            </footer>
                        </t>
                    </templates>
                </kanban>
            </field>
        </record>
    </data>
</odoo>
```

Figura 1: Vista listado principal y la nueva adición, la vista Kanban



The screenshot shows a code editor window with the file 'views.xml' open. The file is located in the directory 'addons > lista_tareas > views'. The code itself defines several XML records for a 'lista_tareas' model.

```
45  <!-- Listado de tareas formato Calendar -->
46  <record id="lista_tareas.view_calendar" model="ir.ui.view">
47    <field name="name">lista_tareas calendar</field>
48    <field name="model">lista_tareas.lista_tareas</field>
49    <field name="arch" type="xml">
50      <calendar string="Tareas" date_start="fecha_limite" color="tipo">
51        <field name="tipo"/>
52        <field name="description"/>
53        <field name="pendiente"/>
54        <field name="realizada"/>
55      </calendar>
56    </field>
57  </record>
58
59
60  <!-- ACCIONES DE VENTANA -->
61
62  <!-- Accionador vista Lista Tareas -->
63  <record id="lista_tareas.action_window" model="ir.actions.act_window">
64    <field name="name">Listado principal</field>
65    <field name="res_model">lista_tareas.lista_tareas</field>
66    <field name="view_mode">list,form</field>
67  </record>
68
69  <!-- Accionador vista Kanban -->
70  <record id="lista_tareas.action_kanban" model="ir.actions.act_window">
71    <field name="name">Tareas formato Kanban</field>
72    <field name="res_model">lista_tareas.lista_tareas</field>
73    <field name="view_mode">kanban,form,list</field>
74    <field name="view_id" ref="lista_tareas.view_kanban"/>
75  </record>
```

Figura 2: Segunda adiciónsn (vista Calendar) y accionadores listado principal y Kanban

The screenshot shows a code editor window with the following details:

- Title Bar:** Shows the file name "views.xml" and the tabs "views" and "views.xml".
- Path:** addons > lista_tareas > views > views.xml
- Code Content:** The XML code defines an action for a calendar view and menu items for the main menu.

```
<!-- Accionador vista Calendar -->
<record id="lista_tareas.action_calendar" model="ir.actions.act_window">
    <field name="name">Tareas formato Calendar</field>
    <field name="res_model">lista_tareas.lista_tareas</field>
    <field name="view_mode">calendar,form,list</field>
    <field name="view_id" ref="lista_tareas.view_calendar"/>
</record>

<!-- ELEMENTOS MENÚ CON ACCIONADORES -->

<!-- Principal -->
<menuitem name="Lista-Tareas" id="lista_tareas.menu_root"
    action="lista_tareas.action_window"/>

<!-- Adyacentes -->
<menuitem name="Formato Kanban" id="lista_tareas.menu_kanban"
    parent="lista_tareas.menu_root" action="lista_tareas.action_kanban"/>

<menuitem name="Formato Calendario" id="lista_tareas.menu_calendario"
    parent="lista_tareas.menu_root" action="lista_tareas.action_calendar"/>

</data>
</odoor>
```

Figura 3: Accionador Calendar y elementos de menú

Y este es todo el código generado para esta actividad. Ambas vistas fueron añadidas en el archivo xml principal *views.xml* por pura simplificación de código.

A continuación unas capturas del módulo *lista_tareas* en funcionamiento.

<input type="checkbox"/> Tipo	Fecha límite	Pendiente	Realizada	Fallida
<input type="checkbox"/> Cita médica	12/18/2025 17:00:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Hacer compra	12/16/2025 11:00:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Sacar al perro	12/15/2025 21:00:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Ir a la peluquería	12/17/2025 13:00:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Ir a la gestoría	12/19/2025 16:00:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/> Compras navideñas	12/20/2025 18:00:00	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Figura 4: Vista principal

Categoría	Tarea	Fecha límite
Cita médica	Vacuna contra la gripe	12/18/2025 17:00:00
	El cónido debe evacuar satisfactoriamente antes de finalizar el día.	12/15/2025 21:00:00
Hacer compra	Leche, patatas, huevos.	12/16/2025 11:00:00
	Hay que arreglar este desastre.	12/17/2025 13:00:00
Ir a la gestoría	Gestionar asuntos.	12/19/2025 16:00:00
	Urgente, no olvidarse!	12/20/2025 18:00:00
Compras navideñas		

Figura 5: Vista Kanban

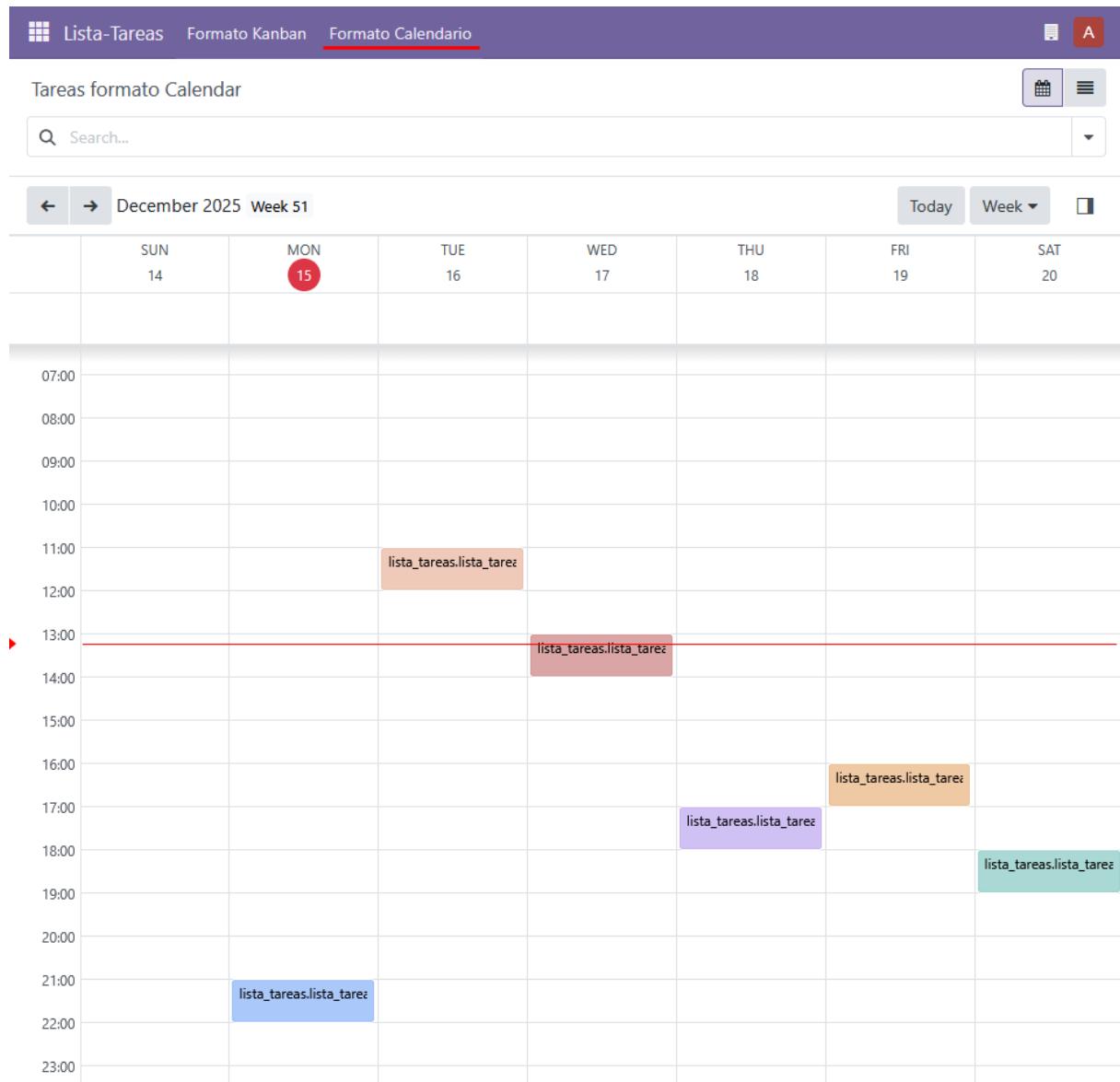


Figura 6: Vista Calendar parte 1

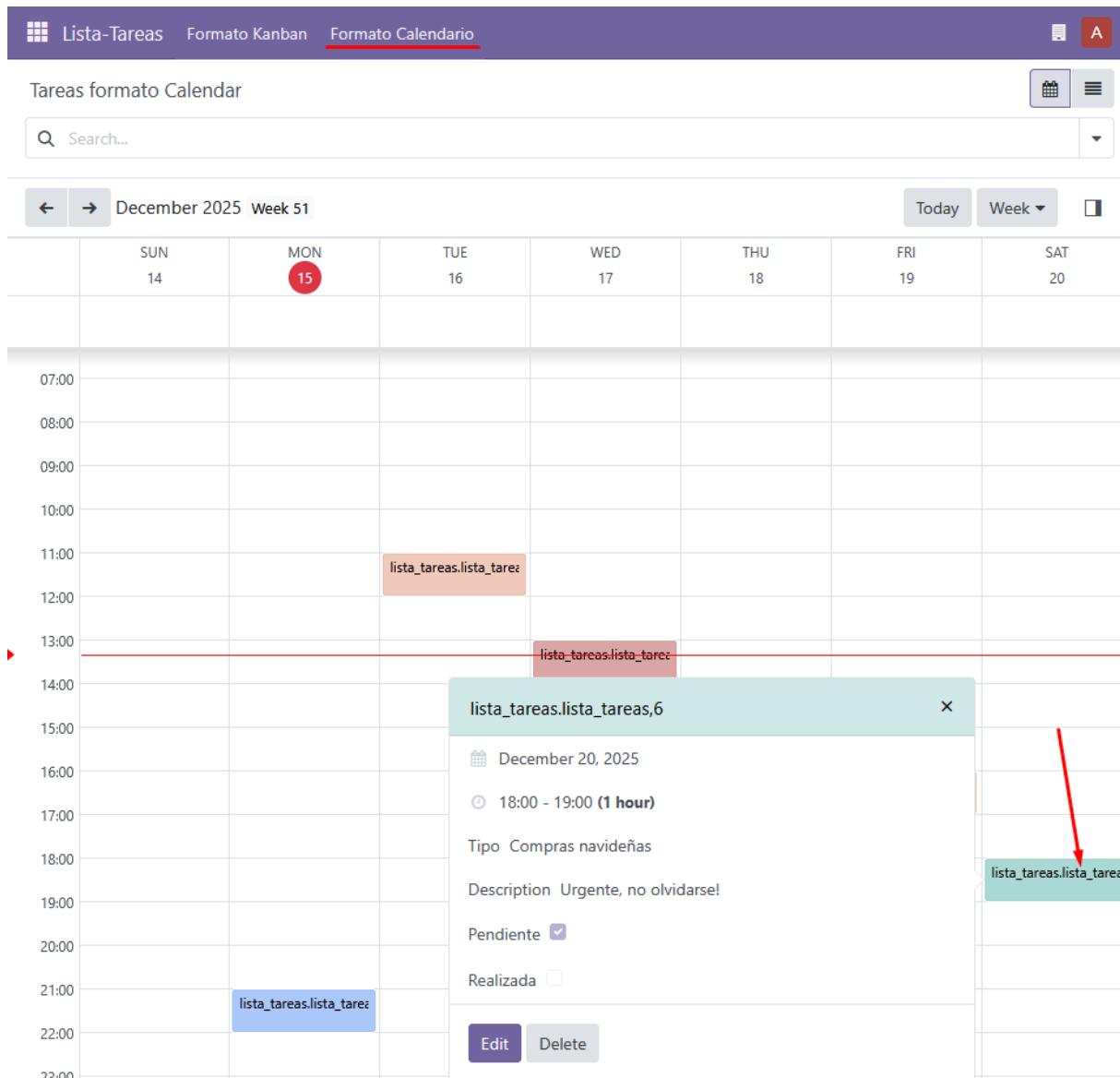


Figura 7: Vista Calendar parte 2 (clic en tarea)

3. Actividad 02 - Biblioteca de cómics

En esta segunda actividad ampliaremos el módulo de prueba ‘EJ03-ComicsSimple’ con objeto de:

- Incluir la posibilidad de gestionar socios, almacenando su nombre, apellido e identificador.
- Introducir al opción de que existan ejemplares de cómics para préstamo.
 - El modelo actual de cómic servirá como referencia de la información del cómic.
 - Se creará un nuevo modelo para gestionar los ejemplares prestables de cada cómic.
- Estos ejemplares de préstamo deben registrar:
 - A quién están prestados (socio).
 - La fecha de inicio y fecha de fin del préstamo.
 - No es necesario mantener un histórico de préstamos, sólo quién tiene el cómic actualmente, cuándo se prestó y la fecha prevista de devolución.
- Restricciones:
 - La fecha de préstamo no puede ser posterior al día actual.
 - La fecha prevista de devolución no puede ser anterior al día actual.

A continuación se mostrarán las capturas del código realizado y las capturas con el módulo en funcionamiento.

Actividad 02...

4. Actividad 03 - Hospital

En esta tercera actividad crearemos un módulo de cero para gestionar los pacientes y médicos de un hospital:

- **Modelo Paciente:**

- Nombre y apellidos del paciente.
- Sintomatología.

- **Modelo Médico:**

- Nombre y apellidos del médico.
- Número de colegiado.

- **Modelo Consulta:**

- Registro de cada vez que un médico atiende a un paciente, incluyendo el diagnóstico.

- Relaciones:

- Un paciente puede haber sido atendido por varios médicos.
- Un médico puede haber atendido a varios pacientes.

- Implementar los modelos y las vistas que se consideren oportunos.

A continuación se mostrarán las capturas del código realizado y las capturas con el módulo en funcionamiento.

Actividad 03...

5. Actividad 04 - Instituto

En esta cuarta y última actividad desarrollaremos un módulo que represente los estudios de ciclos formativos en un instituto:

- **Modelo Ciclo Formativo:**

- Representa un ciclo formativo en el instituto.
- Un ciclo puede tener uno o más módulos asociados.

- **Modelo Módulo,** que relacionará con:

- El ciclo formativo al que pertenece.
- Los alumnos matriculados.
- El profesor que lo imparte.

- **Modelo Alumno:**

- Relacionado con los módulos en los que está matriculado.

- **Modelo Profesor:**

- Relacionado con los módulos que imparte.
- Se deberán implementar las relaciones necesarias y las vistas adecuadas para los cuatro modelos.
- Por último, se creará el rol de usuario *Director* (todos los permisos) y el rol *Profesor* (permiso de sólo lectura).

A continuación se mostrarán las capturas del código realizado y las capturas con el módulo en funcionamiento.

Actividad 04...