

# PRÁCTICA 4 ODOO

Iván Alfaya Pérez

15 de diciembre de 2025



# Índice

1. Introducción	3
2. Actividad 01	3
3. Actividad 02	5

# 1. Introducción

El objetivo de esta práctica es aplicar los conocimientos adquiridos sobre modelos, relaciones y vistas en el desarrollo de aplicaciones. Para ello, se trabajará en la implementación de cuatro actividades que abarcan diferentes contextos, como la gestión de tareas, bibliotecas de cómics, pacientes y médicos, y ciclos formativos. Se debe entregar una documentación con los resultados obtenidos, apoyada mediante capturas de pantalla.

## 2. Actividad 01

Para esta actividad se modificará el módulo de tareas creado en la practica anterior con el fin de que disponga de una vista Kanban y otra vista de calendario.

En mi caso, solo trabajaré sobre la vista definida en el archivo `view.xml`. Lo más idóneo sería crear una vista para cada funcionalidad con el fin de mantener el código modularizado. A continuación, se muestra lo que se añadirá:

Parte de la vista de kanvan:

```
<!--vista kanvan-->
<record id="lista_tareas_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>
      <field name="tarea"/>
      <field name="prioridad"/>
      <field name="urgente"/>
      <field name="realizada"/>
      <field name="deadline"/>

      <templates>
        <t t-name="kanban-box">
          <div class="oe_kanban_card oe_kanban_global_click">
            <strong><field name="tarea"/></strong>
            <div>
              <span>Prioridad: <field name="prioridad"/></span><br/>
              <span>Urgente: <field name="urgente" widget="boolean"/></span><br/>
              <span>Realizada: <field name="realizada" widget="boolean"/></span><br/>
              <span>Fecha limite: <field name="deadline"/></span>
            </div>
          </div>
        </t>
      </templates>
    </kanban>
  </field>
</record>
```

Figura 1: Ejemplo de vista Kanban

Como se puede observar en la imagen, se añade la estructura necesaria para mostrar las tareas en formato Kanban.

A continuación, se incorpora la parte que indica a Odoo qué modelo debe utilizar y qué vistas se deben mostrar al abrir la acción correspondiente.

```

<!-- Acción para vista Kanban -->
<record id='action_tareas_kanban' model='ir.actions.act_window'>
  <field name="name">Mis tareas</field>
  <field name="res_model">lista_tareas.lista_tareas</field>
  <!-- añadir lo de calendar para intercambiar -->
  <field name="view_mode">kanban,list,form,calendar</field>
  <field name="view_id" ref="lista_tareas_kanban"/>
</record>

```

Figura 2: Iconos de intercambio de vistas

Como se puede comprobar, disponemos de varios iconos para intercambiar entre vistas. En cada sección tendremos uno; por ejemplo, cuando estemos en la vista Kanban aparecerá su icono correspondiente. Por tanto, es necesario crear uno para cada vista: lista de tareas, calendario y Kanban.

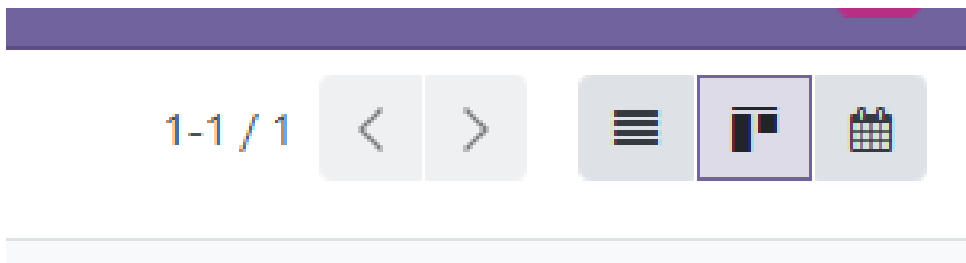


Figura 3: Iconos de intercambio entre vista normal, Kanban y calendario

Finalmente, se añade el *menu item*, que es el enlace que el usuario visualiza y utiliza para abrir la acción.

```

<!-- Menu kanban -->
<menuitem name="Ver Kanban" id="lista_tareas_menu_kanban" parent="lista_tareas_menu_opciones" action="action_tareas_kanban"/>

```

Figura 4: MenuItem de Kanban

## Ejemplo

A continuación, se muestra un ejemplo de una tarea en la vista normal.

Listado de tareas Opciones					
Nuevo Listado de tareas ⚙		Q Buscar...			
<input type="checkbox"/> Tarea	Prioridad	Urgente	Realizada	Fecha Li...	Responsables
<input type="checkbox"/> nueva_tarea	0	<input type="checkbox"/>	<input type="checkbox"/>	17/12/2025	No hay registros

Figura 5: Vista normal de una tarea

Vista Kanban:

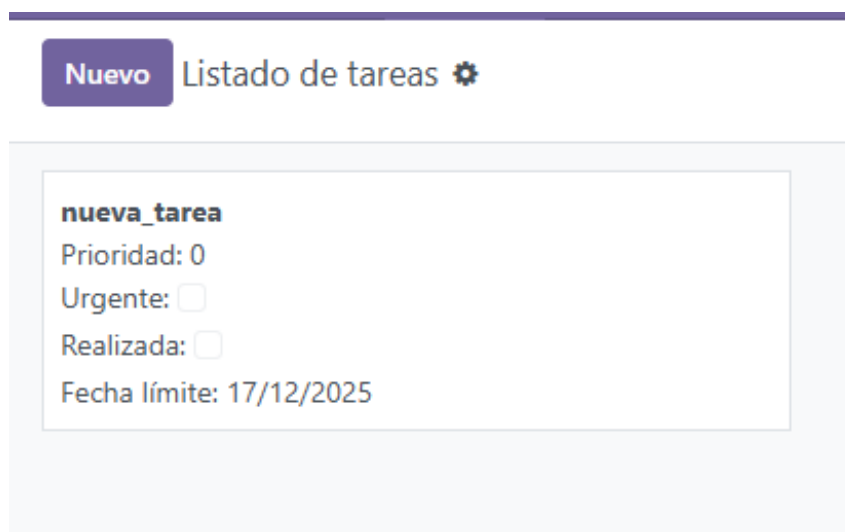


Figura 6: Vista Kanban de una tarea

Vista calendario:



Figura 7: Vista calendario de una tarea

Básicamente, para crear una vista Kanban es necesario definir, en primer lugar, la vista que indica cómo se mostrarán las tareas en este formato. Además, se debe crear la acción, que se encarga de especificar qué modelo se va a abrir, qué vistas se pueden intercambiar (Kanban, lista, formulario y calendario) y cuál será la vista inicial, en este caso la Kanban. Por último, se añade el menu item, que actúa como el enlace mediante el cual el usuario accede a dicha vista. El de calendario seguiría esta estructura.

### 3. Actividad 02

Para este ejercicio se utiliza el ejemplo del módulo de biblioteca disponible en el repositorio de GitHub que se nos facilitó: [enlace\]\(https://github.com/sergarb1/OdooModulosEjemplos\)](https://github.com/sergarb1/OdooModulosEjemplos) al repositorio. La parte escogida es EJ03-ComicsSimple.

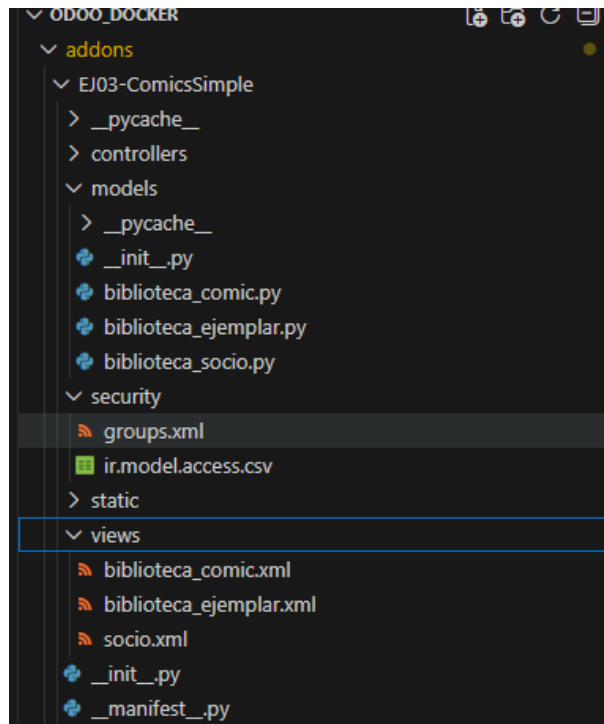


Figura 8: Estructura del módulo de biblioteca

Los archivos base incluidos eran el modelo `biblioteca_comic.py` y el archivo XML `biblioteca_comics.xml`.

Se crearon dos archivos XML adicionales:

- `biblioteca_ejemplar.xml`
- `socio.xml`

Y los modelos correspondientes:

- `biblioteca_ejemplar.py`
- `biblioteca_socio.py`

**Importante:** es necesario añadir en el archivo `manifest` del módulo las nuevas vistas que se han creado.

```

addons > E03-ComicsSimple > ..manifest.py
1 # -*- coding: utf-8 -*-
2
3 'name': "Biblioteca Comics Simple", # Título del módulo
4 'summary': "Gestionar comics (.) (Version simple)", # Resumen de la funcionalidad
5 'description': "",
6 'gestor de bibliotecas (Version Simple)
7
8
9
10 #Indicamos que es una aplicación.
11
12
13 'application': True,
14 'author': "Sergi García",
15 'website': "http://apuntespinformatica.es",
16 'category': "Tools",
17 'version': "0.1",
18 'depends': ["base"],
19
20
21 'data': [
22     #Estos dos primeros ficheros:
23     #1) El primero indica grupo de seguridad basado en rol
24     #2) El segundo indica la política de acceso del modelo
25     #Mas información en https://www.odoo.com/documentation/17.0/es/developer/howtos/rtrtraining/05_securityintro.html
26     # en www.odoo.com/help/creating-security-groups-odoo/
27     'security/groups.xml',
28     'security/ir.model.access.csv',
29
30     #Cargamos la vista de la biblioteca de comics
31     'views/biblioteca_comic.xml',
32     'views/socio.xml',
33     'views/ejemplar.xml',
34     'views/ejemplar_comic.xml',
35 ],
36 # Fichero con data de demo si se inicializa la base de datos con "demo data" (No incluido en ejemplo)
37 # 'demo': [
38 #     'demo.xml'
39 # ],
40

```

Figura 9: Manifest del módulo de biblioteca

## Modelo biblioteca ejemplar

```

1 from odoo import models, fields, api
2 from odoo.exceptions import ValidationError
3 from datetime import date
4
5 class EjemplarComic(models.Model):
6     _name = 'biblioteca.ejemplar'
7
8     _description = 'Ejemplar de Comic para prestamo'
9
10    #relacion de muchos a uno , un comic puede tener varios ejemplares
11    comic_id = fields.Many2one('biblioteca.comic', string='Comic', required=True)
12
13    socio_id = fields.Many2one('biblioteca.socio', string='Socio')
14
15    fecha_prestamo = fields.Date('Fecha de prestamo')
16
17    fecha_devolucion = fields.Date('fecha de devolucion')
18
19    estado = fields.Selection([('disponible', 'Disponible'),
20                              ('prestado', 'Prestado')],
21                             string='Estado',
22                             default='disponible')
23
24    #Comprobar que la fecha de prestamos no puede ser posterior al día actual
25    @api.constrains('fecha_prestamo')
26    def comprobar_fecha_prestamo(self):
27        for cada_registro in self:
28            #comprobacion
29            if cada_registro.fecha_prestamo and cada_registro.fecha_prestamo > date.today():
30                #mensaje de error
31                raise ValidationError("La fecha de prestamo no puede ser posterior al día actual.")
32
33
34    @api.constrains('fecha_devolucion')
35    def comprobar_fecha_devolucion(self):
36        for cada_registro in self:
37            #comprobacion
38            if cada_registro.fecha_devolucion and cada_registro.fecha_devolucion < date.today():
39                #mensaje de error
40                raise ValidationError("La fecha de devolucion no puede ser inferior al día actual.")
41
42
43    # Cambiar estado automáticamente si se asigna un socio
44    @api.onchange('socio_id')
45    def _onchange_socio(self):
46        if self.socio_id:
47            self.estado = 'prestado'
48        else:
49            self.estado = 'disponible'

```

Figura 10: Modelo biblioteca ejemplar

En la imagen se puede observar el modelo del ejemplar, donde se definen relaciones de muchos a uno. Un cómic puede tener varios ejemplares, y cada ejemplar puede estar asociado a un socio. Además, se incluyen los campos de fecha de préstamo, fecha de devolución y estado.

También se implementan comprobaciones para asegurar que la fecha de préstamo no sea posterior a la fecha actual y que la fecha de devolución no sea anterior a la fecha actual.

## Vista ejemplar

```
1 <?xml version="1.0" encoding="utf-8"?>
2 <odoo>
3     <!-- Acción -->
4     <record id="biblioteca_ejemplar_action" model="ir.actions.act_window">
5         <field name="name">Ejemplares de Comics</field>
6         <field name="res_model">biblioteca_ejemplar</field>
7         <field name="view_mode">list,form</field>
8     </record>
9
10    <!-- Menú -->
11    <menuitem name="Ejemplares" id="biblioteca_ejemplar_menu" parent="biblioteca_base_menu" action="biblioteca_ejemplar_action"/>
12
13    <!-- Vista formulario -->
14    <record id="biblioteca_ejemplar_view_form" model="ir.ui.view">
15        <field name="name">Formulario de Ejemplar</field>
16        <field name="model">biblioteca_ejemplar</field>
17        <field name="arch" type="xml">
18            <form>
19                <group>
20                    <field name="comic_id"/>
21                    <field name="socio_id"/>
22                    <field name="fecha_prestamo"/>
23                    <field name="fecha_devolucion"/>
24                    <field name="estado" readonly="1"/>
25                </group>
26            </form>
27        </field>
28    </record>
29
30    <!-- Vista lista -->
31    <record id="biblioteca_ejemplar_view_list" model="ir.ui.view">
32        <field name="name">Lista de Ejemplares</field>
33        <field name="model">biblioteca_ejemplar</field>
34        <field name="arch" type="xml">
35            <list>
36                <field name="comic_id"/>
37                <field name="socio_id"/>
38                <field name="fecha_prestamo"/>
39                <field name="fecha_devolucion"/>
40                <field name="estado"/>
41            </list>
42        </field>
43    </record>
44 </odoo>
```

Figura 11: Vista biblioteca ejemplar

## Modelo biblioteca socio

```
addons > E03-ComicsSimple > models > biblioteca_socio.py > Socio
1 from odoo import models, fields, api
2
3
4 class Socio(models.Model):
5     _name = 'biblioteca.socio'
6     _description = 'Socio biblioteca prueba'
7
8     _rec_name = 'nombre'
9     #campo de nombre del socio
10    nombre = fields.Char(string='Nombre')
11    #campo apellidos del socio
12    apellido = fields.Char(string='Apellido')
13    #identificador unico del socio
14    identificador = fields.Char(string='Identificador', required=True)
15
16
17    _sql_constraints = [
18        ('identificador_uniq', 'UNIQUE (identificador)', 'El identificador del socio debe ser único.')
19    ]
```

Figura 12: Modelo biblioteca socio

En este modelo se definen los campos correspondientes a los socios: nombre, apellidos y un identificador.



## Vista socio

```
addons > [03-ComicsSimple > views > socio.xml
1  <?xml version="1.0" encoding="utf-8"?>
2  <odoo>
3      <!-- Vista formulario -->
4      <record id="biblioteca_socio_view_form" model="ir.ui.view">
5          <field name="name">Formulario de Socio</field>
6          <field name="model">biblioteca.socio</field>
7          <field name="arch" type="xml">
8              <form string="Socio">
9                  <group>
10                     <field name="nombre"/>
11                     <field name="apellido"/>
12                     <field name="identificador"/>
13                 </group>
14             </form>
15         </field>
16     </record>
17
18     <!-- Vista lista -->
19
20     <record id="biblioteca_socio_view_list" model="ir.ui.view">
21         <field name="name">Lista de Socios</field>
22         <field name="model">biblioteca.socio</field>
23         <field name="arch" type="xml">
24             <list string="Socios">
25                 <field name="nombre"/>
26                 <field name="apellido"/>
27                 <field name="identificador"/>
28             </list>
29         </field>
30     </record>
31
32     <!-- Acción para abrir el modelo -->
33     <record id="biblioteca_socio_action" model="ir.actions.act_window">
34         <field name="name">Socios</field>
35         <field name="res_model">biblioteca.socio</field>
36         <field name="view_mode">list,form</field>
37     </record>
38
39     <!-- Menú para acceder -->
40     <menuitem name="Socios" id="biblioteca_socio_menu" parent="biblioteca_base_menu" action="biblioteca_socio_action"/>
41 </odoo>
42
43
```

Figura 13: Vista biblioteca socio

## Ejemplos de uso

Creación de un cómic:

Archivar Comics	
Título	comics2
Autores	
Estado	Disponible
Numero de páginas	100
Activo	<input checked="" type="checkbox"/>
Descripción	comic ejemplo
Precio	10,00
Fecha publicación	02/12/2025
Portada Comic	
Valoración media lectores	10,0000

Figura 14: Creación de un cómic

En esta imagen se puede observar la creación de un nuevo cómic con sus campos correspondientes.

Resultado:

Archivar Comics	
Título	comics2
Autores	
Estado	Disponible
Numero de páginas	100
Activo	<input checked="" type="checkbox"/>
Descripción	comic ejemplo
Precio	10,00
Fecha publicación	02/12/2025
Portada Comic	
Valoración media lectores	10,0000

Figura 15: Listado de cómics creados

Creación de un nuevo socio:

Mi biblioteca (Simple) Comics Socios Ejemplares

Nuevo Socios

Q Buscar...

Nombre	Apellido	Identificador
manuel	alvarez	12345678

Figura 16: Creación de un socio

Creación de un ejemplar:

Se asigna el cómic creado anteriormente y el socio correspondiente, junto con los campos de fecha de préstamo y fecha de devolución.

Mi biblioteca (Simple) Comics Socios Ejemplares

Nuevo Ejemplares de Cómics

Comic comic2

Socio manuel

Fecha de préstamo 08/12/2025

fecha de devolucion 23/12/2025

Estado Prestado

Figura 17: Creación de un ejemplar

Resultado del ejemplar creado:

Mi biblioteca (Simple) Comics Socios Ejemplares

Nuevo Ejemplares de Cómics

Q Buscar...

1-1 / 1

Comic	Socio	Fecha de ...	fecha de ...	Estado
comic2	manuel	08/12/2025	23/12/2025	Disponible

Figura 18: Ejemplar creado correctamente