

# Programación de Servicios y Procesos: Práctica de Docker Local

Álvaro Del Valle Fernández

December 15, 2025



# 1 Introducción

En esta práctica completaré 4 ejercicios diferentes en Odoo usando como referencia los archivos del repositorio de git entregado.

Antes de comenzar con los ejercicios crearé un odoo completamente nuevo en la versión 18, realizando más ajustes para no tener problemas constantes de nuevo.

```
docker-compose up -d
```

```
PS C:\Users\delva\Desktop\Practica4\odooTest4> docker-compose up -d
time="2025-12-12T05:03:36+01:00" level=warning msg="C:\\Users\\delva\\Desktop\\Practica4\\odooTest4\\docker-co
mpose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confus
ion"
[+] Running 2/2
 ✓ Container odootest4-postgres-1 Running 0.0s
 ✓ Container odootest4-odoo-1 Started 1.8s
PS C:\Users\delva\Desktop\Practica4\odooTest4>
```

Credenciales dentro de Odoo:

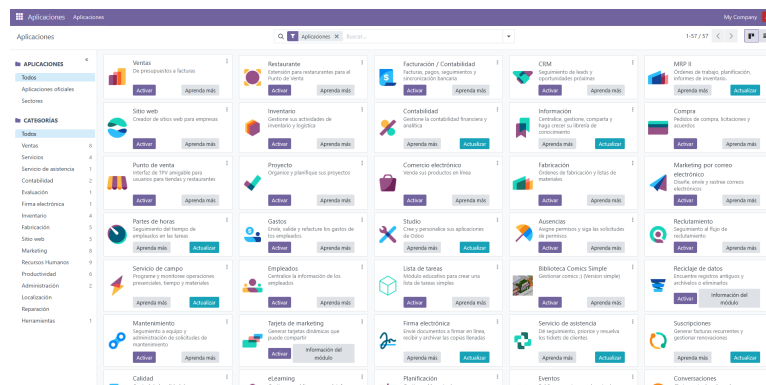


Warning, your Odoo database manager is not protected. To secure it, we have generated the following master password for it:

**smhs-ptws-ugxs**

You can change it below but be sure to remember it, it will be asked for future operations on databases.

Master Password	<input type="text" value="smhs-ptws-ugxs"/>	
Database Name	<input type="text" value="practica4"/>	
Email	<input type="text" value="admin@test.com"/>	
Password	<input type="text" value="admin"/>	
Phone Number	<input type="text"/>	
Language	<input type="text" value="Spanish / Español"/>	
Country	<input type="text" value="Spain"/>	
Demo Data	<input type="checkbox"/>	
<a href="#">Create database</a> or <a href="#">restore a database</a>		

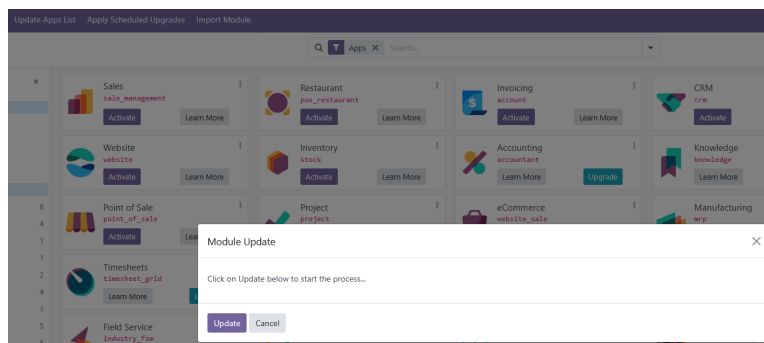
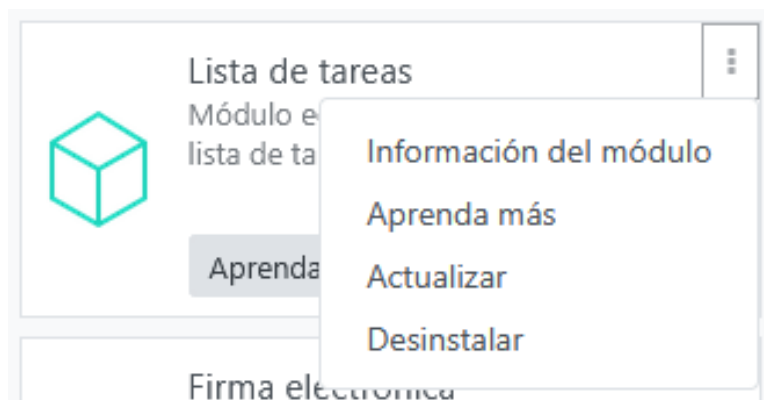


Cada vez que tengo que probar los cambios realizados, debo usar el comando en docker:

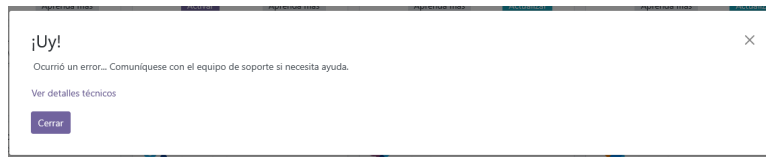
```
docker-compose restart odoo
docker-compose up -d
```

```
PS C:\Users\delva> cd .\Desktop\Practica4\odooTest4\
PS C:\Users\delva\Desktop\Practica4\odooTest4> docker-compose restart odoo
time="2025-12-14T01:53:43+01:00" level=warning msg="C:\\Users\\delva\\Desktop\\Practica4\\odooTest4\\docker-co
mpose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confus
ion"
[+] Restarting 1/1
✓ Container odootest4-odoo-1 Started 1.6s
PS C:\Users\delva\Desktop\Practica4\odooTest4>
```

Tras ello debo hacer click en los tres puntos del modulo indicado y darle a "Actualizar"



Esto puede que me de una notificación de error, normalmente sin indicar el origen del error.



Los logs de Docker son mi principal fuente de informacion para detectar estos errores y arreglarlos.

```
Traceback (most recent call last):
  File "/usr/lib/python3/dist-packages/odoo/http.py", line 2576, in __call__
    response = request._serve_db()
    ~~~~~
  File "/usr/lib/python3/dist-packages/odoo/http.py", line 2103, in _serve_db
    return self._transactioning()
    ~~~~~
  File "/usr/lib/python3/dist-packages/odoo/http.py", line 2166, in _transactioning
    return service_model.retryng(func, env=self.env)
    ~~~~~
  File "/usr/lib/python3/dist-packages/odoo/service/model.py", line 156, in retryng
    result = func()
    ~~~~~
  File "/usr/lib/python3/dist-packages/odoo/http.py", line 2133, in _serve_ir_http
    response = self.dispatcher.dispatch(rule.endpoint, args)
    ~~~~~
  File "/usr/lib/python3/dist-packages/odoo/http.py", line 2381, in dispatch
    result = self.request.registry['ir.http']._dispatch(endpoint)
    ~~~~~
  File "/usr/lib/python3/dist-packages/odoo/addons/base/models/ir_http.py", line 333, in _dispatch
```

Cuando tengo algun error total de odoo uso este comando, el cual borra todos los volúmenes y me permite crearlos de nuevo cuando se corrompen, cosa que es común al realizar cambios relacionados con manejo de datos.

```
docker volume prune -f
```

## 2 Actividad 01

### 2.1 Objetivo

Para realizar este ejercicio debo añadir el datetime y mostrarla en una vista nueva tipo Kanban, junto a otra vista tipo calendario, para ello debo crear dos views nuevas. Kanban es un tipo de vista que muestra la informacion en columnas mediante tarjetas, ideal para esta lista de tareas.

### 2.2 Implementación

Usando de referencia base el archivo view de git, junto a ListaTareasKanban para aprenderla estructura tipo Kanban, creé las dos vistas pedidas.

La vista Kanban:

```

<?xml version="1.0" encoding="utf-8"?>
<odoo>
  <data>
    <record id="view_lista_tareas_kanban" model="ir.ui.view">
      <field name="name">lista.tareas.kanban</field>
      <field name="model">lista_tareas.lista</field>
      <field name="arch" type="xml">
        <kanban default_group_by="estado">
          <field name="tarea"/>
          <field name="prioridad"/>
          <field name="urgente"/>
          <field name="realizada"/>
          <field name="fecha_asignada"/>
          <field name="estado"/>

          <templates>
            <t t-name="kanban-box">
              <div class="oe_kanban_card oe_kanban_global_click">
                <div>
                  <strong><field name="tarea"/></strong>
                </div>
                <div>
                  Fecha: <field name="fecha_asignada"/>
                </div>
                <div>
                  Prioridad: <field name="prioridad"/>
                </div>
                <div>
                  <field name="realizada"/> Realizada
                </div>
              </div>
            </t>
          </templates>
        </kanban>
      </field>
    </record>
  </data>
</odoo>

```

Y la vista del calendario:

```

<?xml version="1.0" encoding="utf-8"?>
<odoo>
  <data>

    <record id="view_lista_tareas_calendar" model="ir.ui.view">
      <field name="name">lista.tareas.calendar</field>

```

```

<field name="model">lista_tareas.lista</field>
<field name="arch" type="xml">
  <calendar string="Tareas por Fecha"
    date_start="fecha_asignada"
    date_stop="fecha_asignada"
    color="prioridad"
    mode="month"
    event_open_popup="true">
    <field name="tarea"/>
    <field name="prioridad"/>
    <field name="estado"/>
    <field name="urgente"/>
    <field name="realizada"/>
  </calendar>
</field>
</record>
</data>
</odoo>

```

Elementos cambiados en manifest para cargar las vistas propiamente:

```

'data': [
    # Permisos de acceso al modelo
    'security/ir.model.access.csv',

    # Archivo XML con las vistas, menús y acciones del modelo
    'views/views.xml',
    'views/kanban.xml',
    'views/calendario.xml',
]

```

También hay que realizar cambios en views debido a las dos vistas nuevas, añadiendo calendar y kanban al viewMode:

```

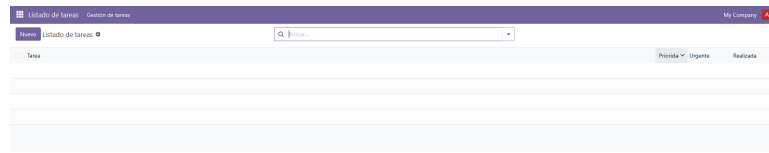
<record model="ir.actions.act_window" id="action_lista_tareas">
<!-- Título visible de la ventana -->
<field name="name">Listado de tareas</field>

<!-- Modelo al que se refiere la acción -->
<field name="res_model">lista_tareas.lista</field>

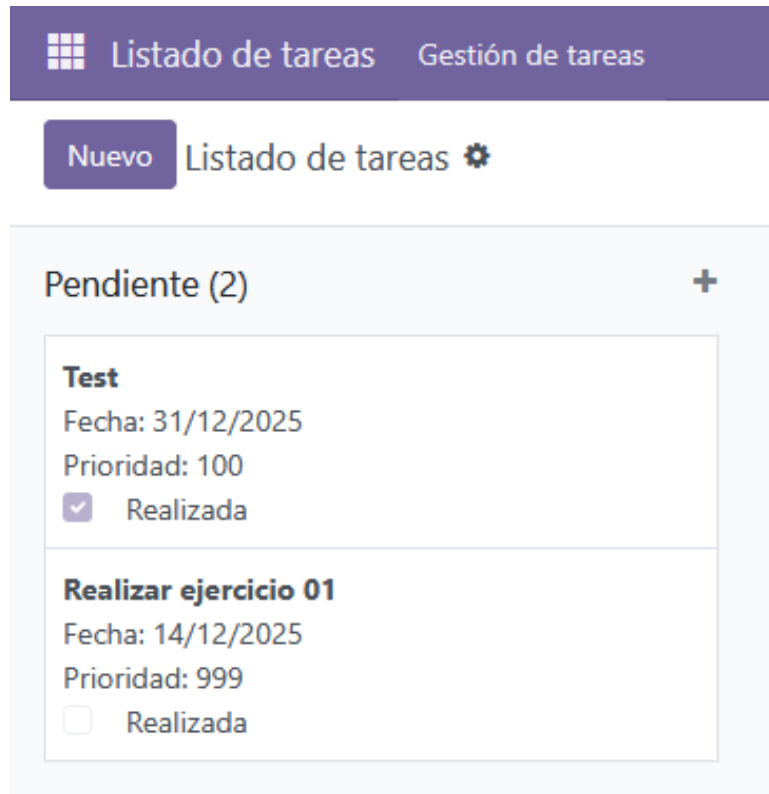
<!-- Tipo de vistas que se mostrarán:
    list → vista en tabla
    form → vista en formulario -->
<field name="view_mode">kanban,calendar,list,form</field>

```

Al enlazar el manifest junto el views permite mostrar correctamente las actualizaciones realizadas. Ejemplo de dos creadas ya con las modificaciones.




Una vez entrado al Listado:






Vista del menu al seleccionar "Nuevo"



 **Listado de tareas**

Gestión de tareas

Nuevo

Listado de tareas  
Nuevo   

Tarea

Realizada ☐

**PRIORIDAD Y URGENCIA**

Fecha Asignada

31/12/2025

Prioridad

0

Urgente

☐

Ahora tenemos la opcion nueva del calendario, en la cual, si le damos click nos muestra la view nueva, aqui podemos seleccionar el dia de entrega en el calendario.

Con esto el ejercicio esta completado, me encontré con numerosos errores, el principal siendo que al realizar cambios no estaba actualizando correctamente docker ni la aplicacion por lo que no podia ver mi progreso. Odoos me indico multiples veces de errores inesperados pero sin dar un codigo de error claro, lo que me hizo perder mucho tiempo. El principal error fue no añadir al manifest las views correctamente, nombrandola calendar, calendario y viewcalendario.xml pensando que era otro elemento el que unia las views entre si. Otro error fue un field name incorrecto que use de views.xml, el cual era llamado pero pero al no existir en models.py no permitia al programa funcionar.

## 3 Actividad 02

### 3.1 Objetivo

Segun el modelo dado de biblioteca, debo añadir diferentes elementos: gestionar socios con nombre, apellido e indentificador, Opcion para gestionar los comics y sus prestamos. Ejemplares con prestado a socio y inicio fin de prestamo. Gestinar las fechas adecuadamente.

### 3.2 Implementación

Dentro de cambios basicos respecto al documento entregado, moví en manifest los valores installable y application debido a que no me detectaba la aplicacion en Odoos, siendo esta la solución.

```
'installable': True,
'application': True,
```

Dentro de biblioteca comic, cree los elementos para gestionar y mapear los campos necesarios para los modelos:

```
from odoo import models, fields, api, _
from odoo.exceptions import ValidationError
from datetime import date

class BibliotecaComic(models.Model):
    _name = 'biblioteca.comic'
    _description = 'Comic'

    nombre = fields.Char(string="Título", required=True)
    estado = fields.Selection([
        ('borrador', 'Borrador'),
        ('publicado', 'Publicado'),
        ('archivado', 'Archivado')
    ], string="Estado", default='borrador')
    paginas = fields.Integer(string="Número de páginas")
    activo = fields.Boolean(string="Activo", default=True)
    autor_ids = fields.Many2many('biblioteca.autor', string="Autores")

    def archivar(self):
        self.write({'activo': False, 'estado': 'archivado'})

class BibliotecaAutor(models.Model):
    _name = 'biblioteca.autor'
    _description = 'Autor'

    nombre = fields.Char(string="Nombre", required=True)
    apellido = fields.Char(string="Apellido", required=True)

class BibliotecaSocio(models.Model):
    _name = 'biblioteca.socio'
    _description = 'Socio'

    identificador = fields.Char(string="ID Socio", required=True)
    nombre = fields.Char(string="Nombre", required=True)
    apellido = fields.Char(string="Apellido", required=True)

class BibliotecaEjemplar(models.Model):
    _name = 'biblioteca.ejemplar'
    _description = 'Ejemplar'

    comic_id = fields.Many2one('biblioteca.comic', string="Cómic", required=True)
    codigo = fields.Char(string="Código ejemplar", required=True)
    estado = fields.Selection([
```

```

        ('disponible', 'Disponible'),
        ('prestado', 'Prestado'),
    ], string="Estado", default='disponible')

socio_id = fields.Many2one('biblioteca.socio', string="Prestado a")
fecha_prestamo = fields.Date(string="Fecha préstamo")
fecha_devolucion = fields.Date(string="Fecha devolución")

@api.constrains('fecha_prestamo')
def _check_fecha_prestamo(self):
    hoy = date.today()
    for ejemplar in self:
        if ejemplar.fecha_prestamo and ejemplar.fecha_prestamo > hoy:
            raise ValidationError(_("Error!!
            Fecha no puede ser posterior a entrega."))

@api.constrains('fecha_devolucion')
def _check_fecha_devolucion(self):
    hoy = date.today()
    for ejemplar in self:
        if ejemplar.fecha_devolucion and ejemplar.fecha_devolucion < hoy:
            raise ValidationError(_("Error! No se puede devolver en una fecha anterior a hoy"))

```

Los ultimos constraints comprueban las restricciones del punto 4, no permitiendo y enviando error cuando la accion es incorrecta.

En ir.model.access.csv me ayuda a gestionar los permisos de acceso, con el comun funcionamiento de read,write,create y delete 1,1,1,1.

```

id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_biblioteca_comic,biblioteca.comic,model_biblioteca_comic,base.group_user,1,1,1,1
access_biblioteca_autor,biblioteca.autor,model_biblioteca_autor,base.group_user,1,1,1,1
access_biblioteca_socio,biblioteca.socio,model_biblioteca_socio,base.group_user,1,1,1,1
access_biblioteca_ejemplar,biblioteca.ejemplar,model_biblioteca_ejemplar,
base.group_user,1,1,1,1

```

Realicé algunos cambios en permisos debido a recurrentes errores a la hora de editar valores dentro de esta.

Dentro del xml use la misma estructura pero añadiendo los campos nuevos mencionados anteriormente en el pyhton.

```

<?xml version="1.0" encoding="utf-8"?>
<odoo>

<record id='biblioteca_comic_action' model='ir.actions.act_window'>
    <field name="name">Cómics</field>

```

```

        <field name="res_model">biblioteca.comic</field>
        <field name="view_mode">list,form</field>
    </record>

    <record id="biblioteca_comic_view_form" model="ir.ui.view">
        <field name="name">Comic - Form</field>
        <field name="model">biblioteca.comic</field>
        <field name="arch" type="xml">
            <form>
                <header>
                    <button type="object" name="archivar" string="Archivar"/>
                </header>
                <sheet>
                    <group>
                        <field name="nombre"/>
                        <field name="autor_ids" widget="many2many_tags"/>
                        <field name="estado"/>
                        <field name="paginas"/>
                    </group>
                </sheet>
            </form>
        </field>
    </record>

    <record id="biblioteca_comic_view_list" model="ir.ui.view">
        <field name="name">Comic - List</field>
        <field name="model">biblioteca.comic</field>
        <field name="arch" type="xml">
            <list>
                <field name="nombre"/>
                <field name="estado"/>
            </list>
        </field>
    </record>

    <record id='biblioteca_socio_action' model='ir.actions.act_window'>
        <field name="name">Socios</field>
        <field name="res_model">biblioteca.socio</field>
        <field name="view_mode">list,form</field>
    </record>

    <record id="biblioteca_socio_view_form" model="ir.ui.view">
        <field name="name">Socio - Form</field>
        <field name="model">biblioteca.socio</field>
        <field name="arch" type="xml">

```

```

        <form>
            <sheet>
                <group>
                    <field name="identificador"/>
                    <field name="nombre"/>
                    <field name="apellido"/>
                </group>
            </sheet>
        </form>
    </field>
</record>

<record id="biblioteca_socio_view_list" model="ir.ui.view">
    <field name="name">Socio - List</field>
    <field name="model">biblioteca.socio</field>
    <field name="arch" type="xml">
        <list>
            <field name="identificador"/>
            <field name="nombre"/>
            <field name="apellido"/>
        </list>
    </field>
</record>

<record id='biblioteca_ejemplar_action' model='ir.actions.act_window'>
    <field name="name">Ejemplares</field>
    <field name="res_model">biblioteca.ejemplar</field>
    <field name="view_mode">list,form</field>
</record>

<record id="biblioteca_ejemplar_view_form" model="ir.ui.view">
    <field name="name">Ejemplar - Form</field>
    <field name="model">biblioteca.ejemplar</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <group>
                    <group>
                        <field name="codigo"/>
                        <field name="comic_id"/>
                        <field name="estado"/>
                    </group>
                    <group>
                        <field name="socio_id"/>
                        <field name="fecha_prestamo"/>
                    </group>
                </sheet>
            </form>
        </field>
    </record>

```

```

        <field name="fecha_devolucion"/>
    </group>
</group>
</sheet>
</form>
</field>
</record>

<record id="biblioteca_ejemplar_view_list" model="ir.ui.view">
    <field name="name">Ejemplar - List</field>
    <field name="model">biblioteca.ejemplar</field>
    <field name="arch" type="xml">
        <list>
            <field name="codigo"/>
            <field name="comic_id"/>
            <field name="estado"/>
            <field name="socio_id"/>
        </list>
    </field>
</record>

<menuitem name="Biblioteca" id="biblioteca_menu_root" />

<menuitem name="Cómics" id="biblioteca_comic_menu"
    parent="biblioteca_menu_root"
    action="biblioteca_comic_action"/>

<menuitem name="Socios" id="biblioteca_socio_menu"
    parent="biblioteca_menu_root"
    action="biblioteca_socio_action"/>

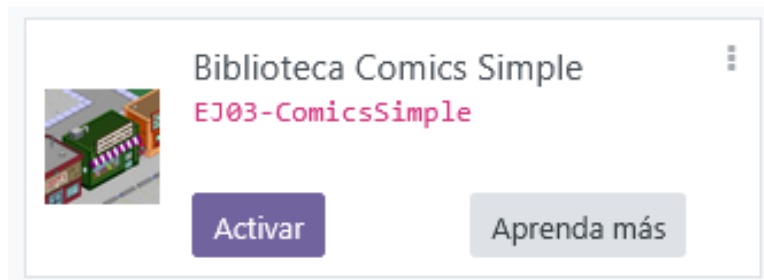
<menuitem name="Ejemplares" id="biblioteca_ejemplar_menu"
    parent="biblioteca_menu_root"
    action="biblioteca_ejemplar_action"/>

</odoo>

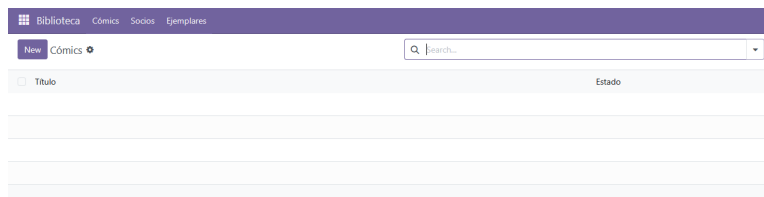
```

Realicé otra version con views separadas en diferentes archivos, una por cada vista siendo practicamente el mismo codigo, el problema fue que Odoo no era capaz de lanzarlo sin dar un codigo de error desconocido, esta version es mas estable.

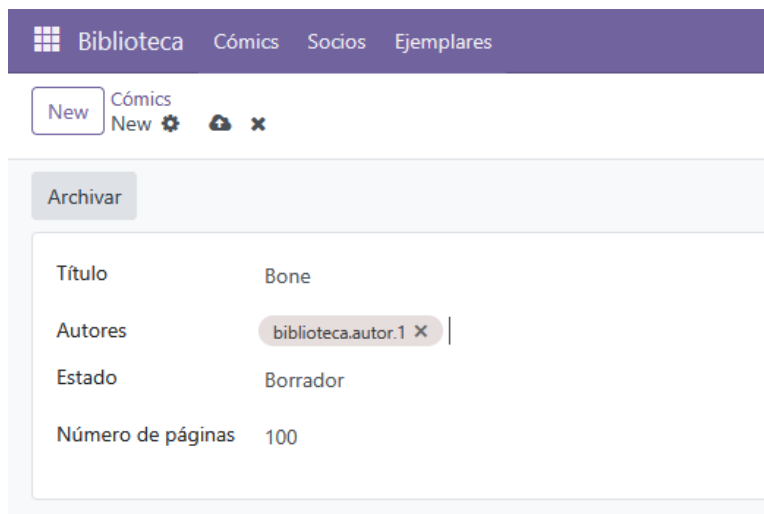
Una vez que Odoo es estable y permita Activar la app:



Podemos acceder a nuestra app actualizada:



Ahora muestra las vistas distintas en la barra, Comics, Socios y Ejemplares. Podemos añadir un comic nuevo:



Y añadir el autor si no esta en el manifest.py como en el ejercicio original.



Create Autores

Nombre

Jeff

Apellido

Smith

Save & Close

Save & New

Discard

Podemos archivarlo y quedará guardado.

Biblioteca

Cómic

Socios

Ejemplares

New

Cómic

Search...

Título

Estado

Bone

Archivado

Podemos crear un nuevo socio:

Biblioteca

Cómic

Socios

Ejemplares

New

Socios

New

ID Socio

1029384756

Nombre

Ramon

Apellido

Perez

Y dentro de ejemplares:

Biblioteca

Cómic

Socios

Ejemplares

New

Ejemplares

Search...

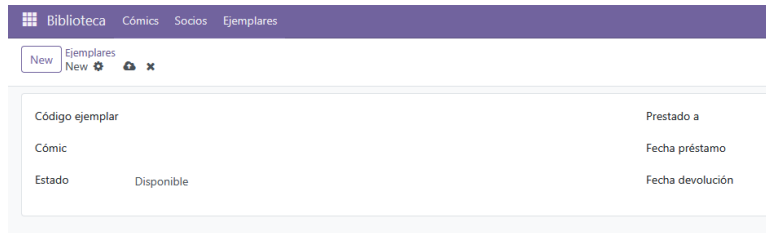
Código ejemplar

Cómic

Estado

Prestado a

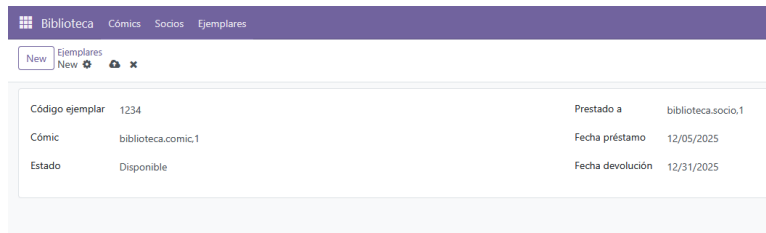
Cumple los campos a crear el prestamo:



The screenshot shows the 'Biblioteca' application interface. At the top, there is a navigation bar with 'Biblioteca', 'Cómics', 'Socios', and 'Ejemplares'. Below the navigation bar, there is a 'New' button and a 'Ejemplares' tab. The main form has the following fields:

Código ejemplar		Prestado a	
Cómic		Fecha préstamo	
Estado	Disponible	Fecha devolución	

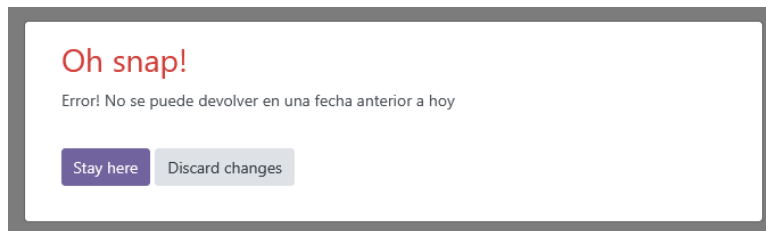
Completamos los datos con la información correcta:



The screenshot shows the 'Biblioteca' application interface with the 'Ejemplares' form filled with data:

Código ejemplar	1234	Prestado a	biblioteca.socio,1
Cómic	biblioteca.comic,1	Fecha préstamo	12/05/2025
Estado	Disponible	Fecha devolución	12/31/2025

Y tenemos creado el sistema correctamente, con las restricciones de fecha de prestamo activas.



The screenshot shows an error message in a modal dialog. The message is:

**Oh snap!**  
Error! No se puede devolver en una fecha anterior a hoy

Below the message are two buttons: 'Stay here' and 'Discard changes'.

## 4 Actividad 03

### 4.1 Objetivo

Este modulo trata sobre la creación de un hospital con tres modelos, paciente, médico y consulta. Estos deben de tener sus campos específicos y definir las relaciones entre medico/pacientes.

Este modulo me generó numerosos problemas al corromperse la base de datos, perdiendo horas en solucionar el error sin perder datos, teniendo que rehacer odoo y borrar mediante comandos los datos ya almacenados. Posiblemente generados por configuración errónea en la carpeta security.

## 4.2 Implementación

Configure las actions y views en hospital views.xml en vez de crear archivos separados, no soy capaz de hacerlo funcionar correctamente del otro metodo.

```
<?xml version="1.0" encoding="utf-8"?>
<odoo>

  <record id="hospital_paciente_action" model="ir.actions.act_window">
    <field name="name">Pacientes</field>
    <field name="res_model">hospital.paciente</field>
    <field name="view_mode">list,form</field>
  </record>

  <record id="hospital_paciente_view_form" model="ir.ui.view">
    <field name="name">Paciente - Formulario</field>
    <field name="model">hospital.paciente</field>
    <field name="arch" type="xml">
      <form>
        <sheet>
          <group>
            <group>
              <field name="nombre"/>
              <field name="apellidos"/>
            </group>
          </group>
          <group>
            <field name="sintomas"/>
          </group>
          <field name="consulta_ids">
            <list>
              <field name="fecha_consulta"/>
              <field name="medico_id"/>
              <field name="diagnostico"/>
            </list>
          </field>
        </sheet>
      </form>
    </field>
  </record>

  <record id="hospital_paciente_view_list" model="ir.ui.view">
    <field name="name">Paciente - Lista</field>
    <field name="model">hospital.paciente</field>
    <field name="arch" type="xml">
      <list>
```

```

        <field name="nombre"/>
        <field name="apellidos"/>
        <field name="sintomas"/>
    </list>
</field>
</record>

<record id="hospital_medico_action" model="ir.actions.act_window">
    <field name="name">Médicos</field>
    <field name="res_model">hospital.medico</field>
    <field name="view_mode">list,form</field>
</record>

<record id="hospital_medico_view_form" model="ir.ui.view">
    <field name="name">Médico - Formulario</field>
    <field name="model">hospital.medico</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <group>
                    <group>
                        <field name="nombre"/>
                        <field name="apellidos"/>
                        <field name="numero_colegiado"/>
                    </group>
                </group>
                <field name="consulta_ids">
                    <list>
                        <field name="fecha_consulta"/>
                        <field name="paciente_id"/>
                        <field name="diagnostico"/>
                    </list>
                </field>
            </sheet>
        </form>
    </field>
</record>

<record id="hospital_medico_view_list" model="ir.ui.view">
    <field name="name">Médico - Lista</field>
    <field name="model">hospital.medico</field>
    <field name="arch" type="xml">
        <list>
            <field name="nombre"/>
            <field name="apellidos"/>
            <field name="numero_colegiado"/>
        </list>
    </field>
</record>

```

```

        </list>
    </field>
</record>

<record id="hospital_consulta_action" model="ir.actions.act_window">
    <field name="name">Consultas</field>
    <field name="res_model">hospital.consulta</field>
    <field name="view_mode">list,form</field>
</record>

<record id="hospital_consulta_view_form" model="ir.ui.view">
    <field name="name">Consulta - Formulario</field>
    <field name="model">hospital.consulta</field>
    <field name="arch" type="xml">
        <form>
            <sheet>
                <group>
                    <group>
                        <field name="paciente_id"/>
                        <field name="medico_id"/>
                        <field name="fecha_consulta"/>
                    </group>
                </group>
                <group>
                    <field name="diagnostico"/>
                </group>
            </sheet>
        </form>
    </field>
</record>

<record id="hospital_consulta_view_list" model="ir.ui.view">
    <field name="name">Consulta - Lista</field>
    <field name="model">hospital.consulta</field>
    <field name="arch" type="xml">
        <list>
            <field name="fecha_consulta"/>
            <field name="paciente_id"/>
            <field name="medico_id"/>
            <field name="diagnostico"/>
        </list>
    </field>
</record>

<menuitem id="hospital_menu_root" name="Hospital"/>

```

```

<menuitem id="hospital_pacientes_menu"
          name="Pacientes"
          parent="hospital_menu_root"
          action="hospital_paciente_action"/>

<menuitem id="hospital_medicos_menu"
          name="Médicos"
          parent="hospital_menu_root"
          action="hospital_medico_action"/>

<menuitem id="hospital_consultas_menu"
          name="Consultas"
          parent="hospital_menu_root"
          action="hospital_consulta_action"/>

</odoo>

```

El manifest e inits siguen la misma estructura que los otros ejercicios.

En models hospital.py añado los campos de nombre,apellidos,sintomas e id consultas para relacionarlo propiamente.

Con One2may indico la relación entre consultas cumpliendo el apartado 3.

```

# -*- coding: utf-8 -*-
from odoo import models, fields, api

class HospitalPaciente(models.Model):
    _name = 'hospital.paciente'
    _description = 'Paciente'
    nombre = fields.Char(string="Nombre", required=True)
    apellidos = fields.Char(string="Apellidos", required=True)
    sintomas = fields.Text(string="Síntomas")
    consulta_ids = fields.One2many('hospital.consulta', 'paciente_id', string="Consultas")

class HospitalMedico(models.Model):
    _name = 'hospital.medico'
    _description = 'Médico'
    nombre = fields.Char(string="Nombre", required=True)
    apellidos = fields.Char(string="Apellidos", required=True)
    numero_colegiado = fields.Char(string="Número de colegiado", required=True)

    consulta_ids = fields.One2many('hospital.consulta', 'medico_id', string="Consultas")

class HospitalConsulta(models.Model):
    _name = 'hospital.consulta'
    _description = 'Consulta'
    paciente_id = fields.Many2one('hospital.paciente', string="Paciente", required=True)

```

```

medico_id = fields.Many2one('hospital.medico', string="Médico", required=True)
diagnostico = fields.Text(string="Diagnóstico", required=True)
fecha_consulta = fields.Datetime(string="Fecha de consulta",
    default=fields.Datetime.now)

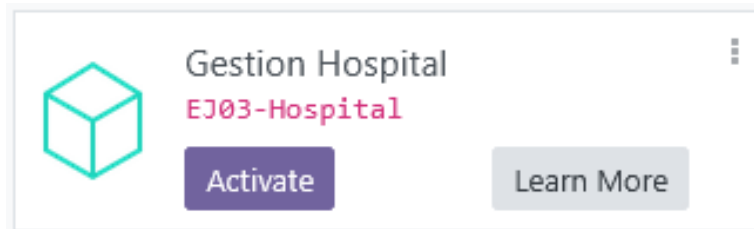
```

```

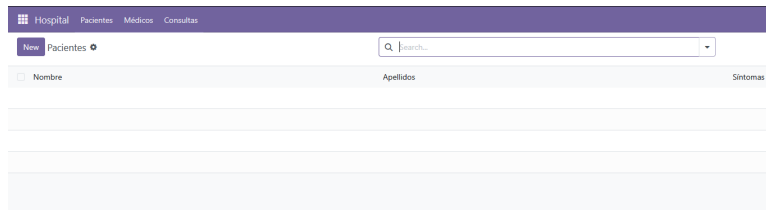
paciente_nombre = fields.Char(related='paciente_id.nombre', store=True,
    string="Nombre Paciente")
medico_nombre = fields.Char(related='medico_id.nombre', store=True,
    string="Nombre Médico")

```

Con esto cumplimos los multiples 1 N entre paciente y consulta, medico y consulta, junto N 1 de consulta y paciente, consulta y médico.



Una vez activada la app nos mostrará correctamente el menu con las views indicadas.



Podemos añadir un paciente junto los campos adecuados:

Hospital
Pacientes
Médicos
Consultas

New
Pacientes
hospital.paciente.1

Nombre Juan  
Apellidos Ramon  
Síntomas Depresión

Fecha de consulta Médico Diagnóstico

Add a line

Agregar medico:

Hospital
Pacientes
Médicos
Consultas

New
Pacientes
hospital.paciente.1

Nombre Juan  
Apellidos Ramon  
Síntomas Depresión

Fecha de consulta Médico Diagnóstico

Add a line

Agregar consulta del medico al paciente:

Create Consultas

Paciente hospital.paciente.NewId\_1  
Médico hospital.medico.1  
Fecha de consulta 12/11/2025 19:06:57  
Diagnóstico Bohemio empedernidj

Save & Close Save & New Discard



## 5 Actividad 04

### 5.1 Objetivo

Crear un módulo para un instituto compuesto de los modelos ciclo formativo, modulo, alumno y profesor, cada uno con sus relaciones entre ellos, en este caso con relacion N M como modulos-alumno.

### 5.2 Implementación

Views en un solo archivo como el resto para eliminar posibles errores. Cada model es llamado y tiene las ids adecuadas, creando la estructura principal del trabajo:

```
<?xml version="1.0" encoding="utf-8"?>
<odoo>
  <record id="ciclo_formativo_action" model="ir.actions.act_window">
    <field name="name">Ciclos Formativos</field>
    <field name="res_model">ciclo.formativo</field>
    <field name="view_mode">list,form</field>
  </record>

  <menuitem name="Gestión Educativa" id="gestion_educativa_menu"/>

  <menuitem name="Ciclos Formativos" id="ciclo_formativo_menu"
    parent="gestion_educativa_menu"
    action="ciclo_formativo_action"/>

  <record id="ciclo_formativo_view_list" model="ir.ui.view">
    <field name="name">ciclo.formativo.list</field>
    <field name="model">ciclo.formativo</field>
    <field name="arch" type="xml">
      <list>
        <field name="name"/>
        <field name="codigo"/>
      </list>
    </field>
  </record>

  <record id="ciclo_formativo_view_form" model="ir.ui.view">
    <field name="name">ciclo.formativo.form</field>
    <field name="model">ciclo.formativo</field>
    <field name="arch" type="xml">
      <form>
        <sheet>
```

```

        <div class="oe_title">
            <label for="name" string="Ciclo Formativo"/>
            <h1><field name="name"/></h1>
            <label for="codigo" string="Código"/>
            <h2><field name="codigo"/></h2>
        </div>
        <group>
            <group>
                <field name="name"/>
                <field name="codigo"/>
            </group>
            <group>
                <field name="descripcion" widget="textarea"/>
            </group>
        </group>
    </sheet>
</form>
</field>
</record>

<record id="modulo_action" model="ir.actions.act_window">
    <field name="name">Módulos</field>
    <field name="res_model">modulo</field>
    <field name="view_mode">list,form</field>
</record>

<menuitem name="Módulos" id="modulo_menu"
    parent="gestion_educativa_menu"
    action="modulo_action"/>

<record id="modulo_view_list" model="ir.ui.view">
    <field name="name">modulo.list</field>
    <field name="model">modulo</field>
    <field name="arch" type="xml">
        <list>
            <field name="name"/>
            <field name="codigo"/>
            <field name="horas"/>
        </list>
    </field>
</record>

<record id="modulo_view_form" model="ir.ui.view">
    <field name="name">modulo.form</field>
    <field name="model">modulo</field>

```

```

<field name="arch" type="xml">
  <form>
    <sheet>
      <div class="oe_title">
        <label for="name" string="Módulo"/>
        <h1><field name="name"/></h1>
        <label for="codigo" string="Código"/>
        <h2><field name="codigo"/></h2>
      </div>
      <group>
        <group>
          <field name="name"/>
          <field name="codigo"/>
          <field name="horas"/>
        </group>
      </group>
    </sheet>
  </form>
</field>
</record>

<record id="alumno_action" model="ir.actions.act_window">
  <field name="name">Alumnos</field>
  <field name="res_model">alumno</field>
  <field name="view_mode">list,form</field>
</record>

<menuitem name="Alumnos" id="alumno_menu"
  parent="gestion_educativa_menu"
  action="alumno_action"/>
<record id="alumno_view_list" model="ir.ui.view">
  <field name="name">alumno.list</field>
  <field name="model">alumno</field>
  <field name="arch" type="xml">
    <list>
      <field name="name"/>
      <field name="apellidos"/>
      <field name="dni"/>
    </list>
  </field>
</record>
<record id="alumno_view_form" model="ir.ui.view">
  <field name="name">alumno.form</field>
  <field name="model">alumno</field>
  <field name="arch" type="xml">
    <form>

```

```

<sheet>
  <div class="oe_title">
    <label for="name" string="Alumno"/>
    <h1><field name="name"/> <field name="apellidos"/></h1>
    <label for="dni" string="DNI"/>
    <h2><field name="dni"/></h2>
  </div>
  <group>
    <group>
      <field name="name"/>
      <field name="apellidos"/>
      <field name="dni"/>
    </group>
    <group>
      <field name="fecha_nacimiento"/>
    </group>
  </group>
</sheet>
</form>
</field>
</record>
<record id="profesor_action" model="ir.actions.act_window">
  <field name="name">Profesores</field>
  <field name="res_model">profesor</field>
  <field name="view_mode">list,form</field>
</record>

<menuitem name="Profesores" id="profesor_menu"
  parent="gestion_educativa_menu"
  action="profesor_action"/>

<record id="profesor_view_list" model="ir.ui.view">
  <field name="name">profesor.list</field>
  <field name="model">profesor</field>
  <field name="arch" type="xml">
    <list>
      <field name="name"/>
      <field name="apellidos"/>
      <field name="dni"/>
    </list>
  </field>
</record>
<record id="profesor_view_form" model="ir.ui.view">
  <field name="name">profesor.form</field>
  <field name="model">profesor</field>
  <field name="arch" type="xml">

```

```

<form>
  <sheet>
    <div class="oe_title">
      <label for="name" string="Profesor"/>
      <h1><field name="name"/> <field name="apellidos"/></h1>
      <label for="dni" string="DNI"/>
      <h2><field name="dni"/></h2>
    </div>
    <group>
      <group>
        <field name="name"/>
        <field name="apellidos"/>
        <field name="dni"/>
      </group>
      <group>
        <field name="especialidad"/>
      </group>
    </group>
  </sheet>
</form>
</field>
</record>
</odoo>

```

Dentro de los modelos declaré las relaciones:

```

from odoo import models, fields

class Profesor(models.Model):
    _name = 'profesor'
    _description = 'Profesor'

    name = fields.Char(string='Nombre', required=True)
    apellidos = fields.Char(string='Apellidos', required=True)
    dni = fields.Char(string='DNI', required=True, size=9)
    especialidad = fields.Char(string='Especialidad')
    modulo_ids = fields.One2many('modulo', 'profesor_id', string='Módulos que imparte')

```

Como podemos ver las declaraciones se cumplen correctamente enlazando 1 N profesor módulos.

En security ir.model.access.csv declaré los permisos de cada miembro, esta versión es estable y no me volvió a dar errores aunque aunque tengo dudas sobre la configuración final de los permisos:

```

id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_ciclo_formativo,ciclo.formativo,model_ciclo_formativo,role_director,1,1,1,1
access_ciclo_formativo_user,ciclo.formativo,model_ciclo_formativo,base.group_user,1,0,0,0

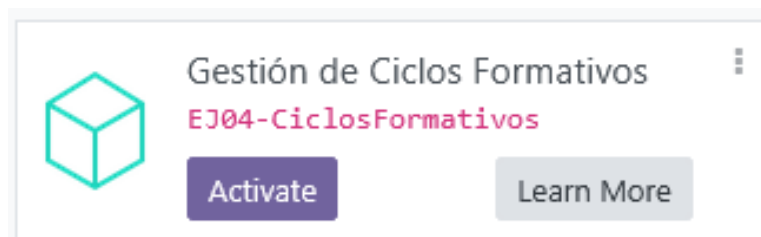
```

```

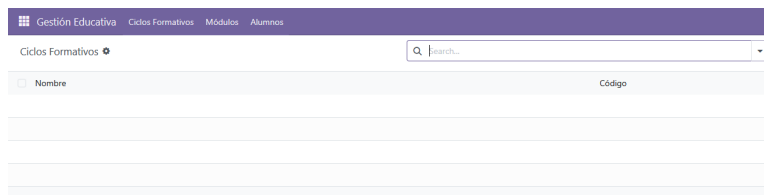
access_modulo,modulo,model_modulo,role_director,1,1,1,1
access_modulo_user,modulo,model_modulo,base.group_user,1,0,0,0
access_alumno,alumno,model_alumno,role_director,1,1,1,1
access_alumno_user,alumno,model_alumno,base.group_user,1,0,0,0
access_profesor_director,profesor,model_profesor,role_director,1,1,1,1
access_profesor_profesor,profesor,model_profesor,role_profesor,1,0,0,0
access_profesor_user,profesor,model_profesor,base.group_user,0,0,0,0

```

Tras llegar a uba version estable, intentaremos activar la app:



Tras entrar mostrará correctamente los elementos:



Error al cambiar los permisos y permitir solo al director editarlos, por lo que no podía usar los campos para añadir elementos nuevos.

Error en el ir.model.access.csv que produjo que no lo detectase sin dar un error claro.

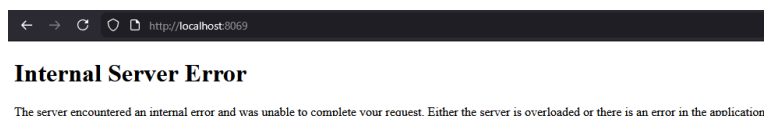
Comandos extra para intentar que abra el módulo:

```

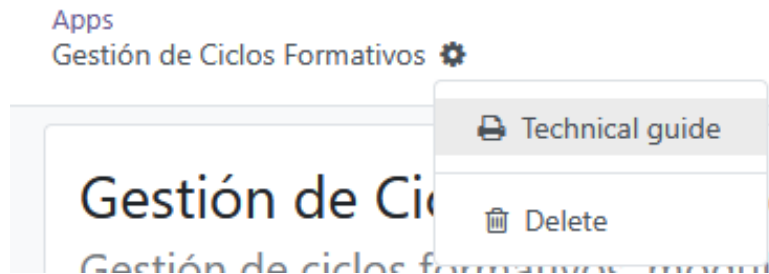
Get-ChildItem -Path . -Recurse -Filter "*.pyc" -File | Remove-Item -Force
Get-ChildItem -Path . -Recurse -Filter "__pycache__" -Directory | Remove-Item -Recurse -For

```

Multiples errores con las horas



Error continuo que no se solucionó hasta que seleccioné Delete y borre y reinstale la App de nuevo, tampoco sin indicar desde Odoo la causa del problema.



## 6 Conclusión

Mi mayor problema al realizar estos ejercicios fue la falta de logs adecuados que indicasen claramente el origen del problema, la mayor parte del tiempo fue dedicado a encontrar el origen del error y leer logs para comprender que estaba mal tras cada cambio. El mayor error que me encuentre fue en la actividad 03 donde Odoo dio un código de error al crear los archivos, creándolos de forma parcial, indicando que esa base de datos ya existía pero sin permitir lanzar la aplicación o editar directamente.

Para ello necesite realizar un

```
docker-compose down -v
docker volume prune -f
```

Borrabo la base de datos, aunque por algún motivo me seguía detectando los datos, por lo que necesité configurar Odoo de nuevo para que funcionase parcialmente de nuevo. No sé si existe una forma de realizar debugging con catch errors que permita realmente saber qué elemento falla sin tardar 30 minutos en arreglar el error.

Traceback (most recent call last):

```
File "/usr/lib/python3/dist-packages/odoo/http.py", line 2576, in __call__
```

```
    response = request._serve_db()
```

```
~~~~~
```

```
File "/usr/lib/python3/dist-packages/odoo/http.py", line 2103, in _serve_db
```

```
    return self._transactioning()
```

```
~~~~~
```

```

File "/usr/lib/python3/dist-packages/odoo/http.py", line 2166, in _transactioning
    return service_model.retryng(func, env=self.env)
    ~~~~~

File "/usr/lib/python3/dist-packages/odoo/service/model.py", line 156, in retryng
    result = func()
    ~~~~~

File "/usr/lib/python3/dist-packages/odoo/http.py", line 2133, in _serve_ir_http
    response = self.dispatcher.dispatch(rule.endpoint, args)
    ~~~~~

```

Siendo los errores similares a esto, indicando los puntos que son afectados en vez de el punto donde realmente da el error.