

1 Introducción

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

Antes de comenzar con los ejercicios crear un odoo completamente nuevo en la version 18, realizando mas 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mpose.yml: the attribute 'version' is obsolete, it will be ignored, please remove it to avoid potential confus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

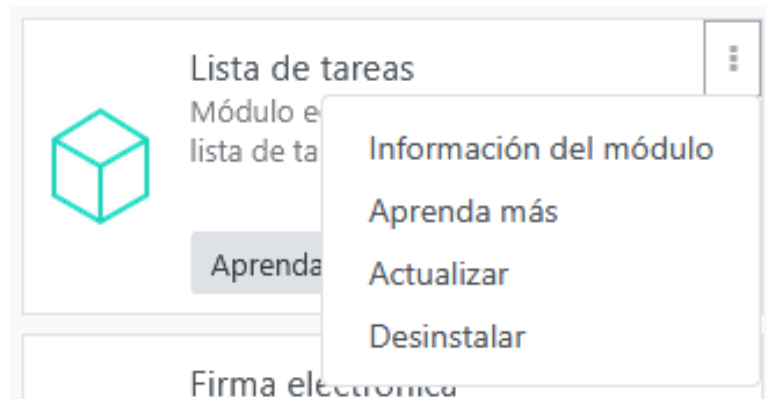
You can change it below but be sure to remember it, it will be asked for future operations on databases.

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"/>	
<input type="button" value="Create database"/> or restore a database		

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

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

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 información 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 2183, 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
```

2 01: Gestión de Tareas Mejorada

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 el archivo view de git, junto guias para aprender la estructura Kanban, cree 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>

```

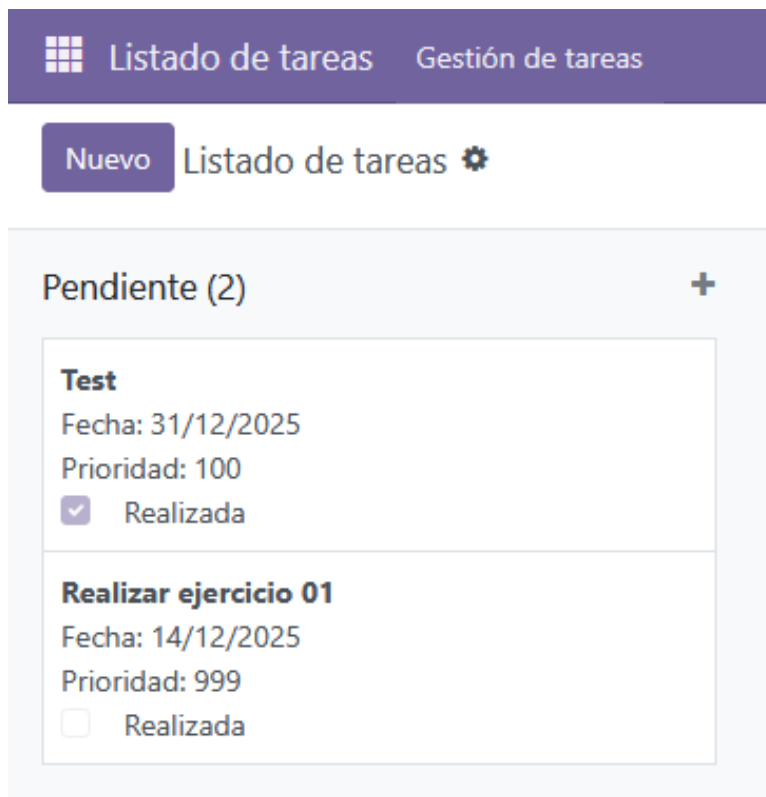
```

<!-- 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.






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.

Listado de tareas

Gestión de tareas

Nuevo

Listado de tareas

Nuevo

Tarea

Realizada

PRIORIDAD Y URGENCIA

Fecha Asignada

Prioridad

Urgente

<

diciembre 2025

>

#	lun	mar	mié	jue	vie	sáb	dom
49	1	2	3	4	5	6	7
50	8	9	10	11	12	13	14
51	15	16	17	18	19	20	21
52	22	23	24	25	26	27	28
1	29	30	31				

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 02: Biblioteca de Cómics Ampliada

3.1 Objetivo

3.2 Implementación

3.2.1 Modelo de Datos

4 03: Gestión Hospitalaria

4.1 Objetivo

4.2 Implementación

4.2.1 Modelo de Datos

5 04: Gestión de Ciclos Formativos

5.1 Objetivo

5.2 Implementación

5.2.1 Modelo de Datos