

Arquitectura Aplicacion Movil

Manuel Añel García

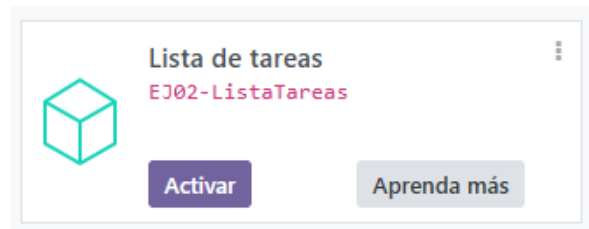
9 de diciembre de 2025

1. Introduction

En este documento se va a explicar el trabajo de SGE que consta de 4 actividades en las que se van a modificar, crear y ampliar módulos de Odoo, para poder empezar con esta práctica hay que reutilizar el contenedor que teníamos ya de Odoo y clonar el repositorio de módulos dicho en los requisitos de la práctica.

2. Actividad 01

Primero tenemos que activar el modulo en el que vamos a trabajar, buscamos "lista de tareas" activamos.



Tenemos que entrar en addons al modulo EJ02-ListaTareas y vamos a su view.xml para hacer otra vista pero en formato kanban, para empezar tenemos que agregar kanban en el view mode como se ve a continuación.

```
1 <field name="view_mode">list,form,kanban</field>
```

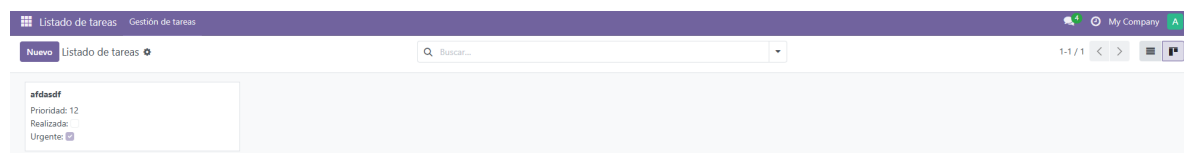
Hay que crear un nuevo record que va a pertenecer al formato kanaban, en los comentarios del código está explicado el funcionamiento

```

1 <record id="view_lista_tareas_kanban" model="ir.ui.view">
2   <!-- Definimos el nombre interno, modelo odoo y arquitectura -->
3   <field name="name">lista.tareas.kanban</field>
4   <field name="model">lista_tareas.lista</field>
5   <field name="arch" type="xml">
6
7     <kanban>
8       <!-- Definimos los campos del modelo -->
9       <field name="tarea"/>
10      <field name="prioridad"/>
11      <field name="urgente"/>
12      <field name="realizada"/>
13
14      <templates>
15        <!-- Creamos una caja kanban -->
16        <t t-name="kanban-box">
17          <div class="oe_kanban_global_click o_kanban_record">
18
19            <!-- Título de la tarea -->
20            <div class="o_kanban_record_top">
21              <strong><t t-esc="record.tarea.value"/></strong>
22            </div>
23
24            <!-- Prioridad -->
25            <div>
26              Prioridad: <t t-esc="record.prioridad.value"/>
27            </div>
28
29            <!-- Checkbox realizada -->
30            <div>
31              Realizada: <field name="realizada"/>
32            </div>
33
34            <!-- Checkbox urgente -->
35            <div>
36              Urgente: <field name="urgente"/>
37            </div>
38
39          </div>
40        </t>
41      </templates>
42    </kanban>
43
44  </field>
45 </record>

```

Podemos ver como se ve la lista de tareas en formato kanban y como se puede alternar la vista arriba a la derecha



Ahora hay que modificar las tareas para que tengan una fecha asignada y crear una nueva vista para visualizarlas en un formato Calendario según esa fecha. Empezamos poniendo en el archivo python un nuevo campo que va a ser fecha con formato Date.

```

1 fecha = fields.Date(string="Fecha")

```

Agregamos el campo fecha en la vista list y form

```
1 <field name="fecha"/>
```

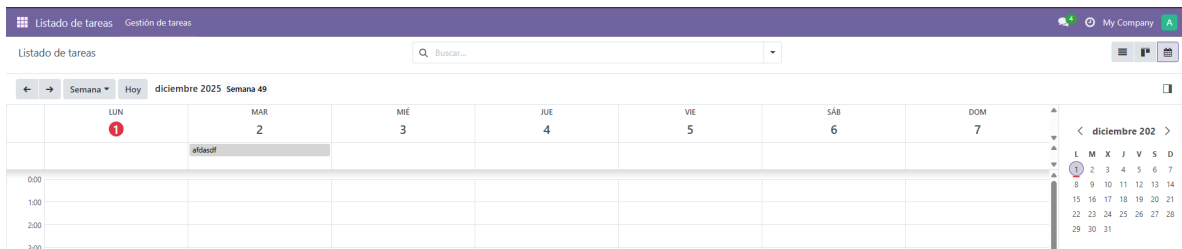
Donde antes pusimos kanban para para añadir el modo kanban hacemos lo mismo pero con calendar.

```
1 <field name="view_mode">list,form,kanban,calendar</field>
```

Ahora creamos el nuevo record para hacer la nueva vista

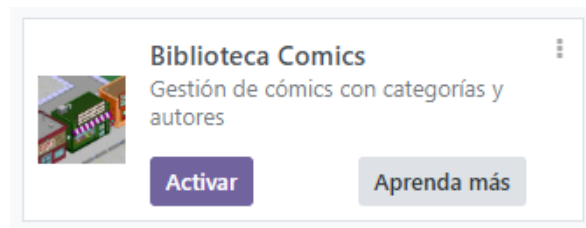
```
1 <record id="view_lista_tareas_calendar" model="ir.ui.view">
2   <!-- Definimos el nombre interno, modelo odoo y arquitectura -->
3   <field name="name">lista.tareas.calendar</field>
4   <field name="model">lista_tareas.lista</field>
5   <field name="arch" type="xml">
6     <!-- Creamos el calendario y decimos con el date_start que se coloque mediante
7       el campo fecha -->
8     <calendar string="Calendario de tareas" date_start="fecha">
9       <!-- Se muestra el nombre de la tarea -->
10      <field name="tarea"/>
11    </calendar>
12  </field>
</record>
```

Vemos como se muestra correctamente en la fecha que puse



3. Actividad 02

Primero hay que encontrar el modulo sobre el que vamos a trabajar que es y activarlo



Aquí lo que hacemos es crear un nuevo archivo .py para manejar los socios

```
1 class BibliotecaSocio(models.Model):
2     # Nombre interno del modelo en Odoo
3     _name = 'biblioteca.socio'
4     # Descripción del modelo
5     _description = 'Socio de la biblioteca'
6
7     # Campo para el nombre del socio, obligatorio
8     nombre = fields.Char(string='Nombre', required=True)
9     # Campo para el apellido del socio, obligatorio
10    apellido = fields.Char(string='Apellido', required=True)
11    # Campo para el identificador único del socio
12    identificador = fields.Char(string='ID Socio', required=True, unique=True)
13
14    prestamo_ids = fields.One2many( # Relación One2many: un socio puede tener varios
        ejemplares prestados
15        'biblioteca.ejemplar', # Es el modelo relacionado
16        'socio_id', #Es el campo en el modelo relacionado que apunta a este socio
17        string='Ejemplares Prestados'
18    )
19
20    # Método para mostrar cómo se representa cada socio en campos Many2one
21    def name_get(self):
22        result = []
23        for record in self:
24            nombre_completo = f"{record.nombre} {record.apellido} ({record.
                identificador})"
25            result.append((record.id, nombre_completo))
26        return result
```

Ahora haay que crear una nueva vista xml para el modelo que acabamos de crear

```
1 <odoo>
2
3 <!-- Acción que abre la vista de Socios -->
4 <record id="biblioteca_socio_action" model="ir.actions.act_window">
5     <field name="name">Socios</field> <!-- Nombre de la acción -->
6     <field name="res_model">biblioteca.socio</field> <!-- Modelo que abre -->
7     <field name="view_mode">list,form</field> <!-- Tipos de vista -->
8     <field name="help" type="html">
9         <p>
10             Crea y gestiona los socios de la biblioteca.
11         </p>
12     </field>
13 </record>
14
15 <!-- Menú que apunta a la acción de Socios -->
16 <menuitem name="Socios" id="menu_biblioteca_socio" parent="biblioteca_base_menu"
17     action="biblioteca_socio_action"/>
18
19 <!-- Vista lista de socios -->
20 <record id="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>
25             <field name="nombre"/> <!-- Muestra el nombre -->
26             <field name="apellido"/> <!-- Muestra el apellido -->
27             <field name="identificador"/> <!-- Muestra el ID del socio -->
28         </list>
29     </field>
30 </record>
31
32 <!-- Vista formulario de socio -->
33 <record id="view_form_socio" model="ir.ui.view">
34     <field name="name">Formulario Socio</field>
35     <field name="model">biblioteca.socio</field>
36     <field name="arch" type="xml">
37         <form string="Socio de la Biblioteca">
38             <group>
39                 <field name="nombre"/> <!-- Campo nombre -->
40                 <field name="apellido"/> <!-- Campo apellido -->
41                 <field name="identificador"/> <!-- Campo ID socio -->
42                 <field name="prestamo_ids" readonly="1"/> <!-- Relación con los
43                     ejemplares prestados, solo lectura -->
44             </group>
45         </form>
46     </field>
47 </record>
48 </odoo>
```

Aquí podemos ver como se crean los socios

Nombre

Apellido

ID Socio

| Ejemplares Prestados | Cómic | Prestado a | Estado | Fecha de ... Fecha pre... |
|----------------------|-------|------------|--------|---------------------------|
| | | | | |
| | | | | |
| | | | | |

Aqui podemos ver un socio que tiene un ejemplar, esto ya lo veremos mas adelante

<

Ahora hay que crear un nuevo modelo para los ejemplares y tener en cuenta las fechas de préstamo.

```
1 class BibliotecaEjemplar(models.Model):
2     _name = 'biblioteca.ejemplar' # Nombre técnico del modelo
3     _description = 'Ejemplar de cómic prestable' # Descripción del modelo
4
5     # Campo Many2one que enlaza con el cómic al que pertenece este ejemplar (un comic
6     # puede tener varios ejemplares)
7     comic_id = fields.Many2one(
8         'biblioteca.comic',
9         string='Cómico', # Etiqueta que se muestra en la interfaz
10        required=True, # Campo obligatorio
11        ondelete='cascade' # Si se borra el cómic, se borran sus ejemplares
12    )
13
14    # Campo Many2one que enlaza con el socio que tiene prestado el ejemplar (un socio
15    # puede tener varios ejemplares)
16    socio_id = fields.Many2one(
17        'biblioteca.socio',
18        string='Prestado a' # Etiqueta en la interfaz
19    )
20
21    # Fecha en que se realizó el préstamo
22    fecha_prestamo = fields.Date(string='Fecha de préstamo')
23    # Fecha prevista para devolver el ejemplar
24    fecha_devolucion_prevista = fields.Date(string='Fecha prevista de devolución')
25
26    # Estado del ejemplar: disponible o prestado
27    estado = fields.Selection(
28        [('disponible', 'Disponible'),
29         ('prestado', 'Prestado')],
30        string='Estado',
31        default='disponible' # Valor por defecto
32    )
33
34    # La fecha de préstamo no puede ser futura
35    @api.constrains('fecha_prestamo')
36    def _check_fecha_prestamo(self):
37        hoy = fields.Date.today() # Fecha actual
38        for record in self:
39            if record.fecha_prestamo and record.fecha_prestamo > hoy:
40                raise ValidationError(
41                    'La fecha de préstamo no puede ser posterior al día actual.'
42                )
43
44    #La fecha prevista de devolución no puede ser anterior a hoy
45    @api.constrains('fecha_devolucion_prevista')
46    def _check_fecha_devolucion(self):
47        hoy = fields.Date.today() # Fecha actual
48        for record in self:
49            if record.fecha_devolucion_prevista and record.fecha_devolucion_prevista <
50                hoy:
51                raise ValidationError(
52                    'La fecha prevista de devolución no puede ser anterior al día
53                    actual.'
54                )
55
56    # Cambio automático del estado cuando se selecciona o quita un socio
57    @api.onchange('socio_id')
58    def _onchange_socio(self):
59        for record in self:
60            if record.socio_id: # Si hay un socio asignado
61                record.estado = 'prestado' # Cambia a prestado
62            else:
63                record.estado = 'disponible' # Si no hay socio, está disponible
```

Ahora hay que crear una nueva vista para el modelo que acabamos de crear

```
1 <odoo>
2
3 <!-- Acción para abrir la vista de Ejemplares (listado y formulario) -->
4 <record id="biblioteca_ejemplar_action" model="ir.actions.act_window">
5   <field name="name">Ejemplares de Cómic</field> <!-- Nombre que verá el
6     usuario -->
7   <field name="res_model">biblioteca.ejemplar</field> <!-- Modelo al que apunta
8     -->
9   <field name="view_mode">list,form</field> <!-- Vistas disponibles: lista y
10     formulario -->
11 </record>
12
13 <!-- Menú que abre la acción de Ejemplares dentro del menú base "Mi biblioteca"
14 -->
15 <menuitem name="Ejemplares"
16   id="menu_biblioteca_ejemplar"
17   parent="biblioteca_base_menu"
18   action="biblioteca_ejemplar_action"/>
19
20 <!-- Vista Formulario para crear o editar un ejemplar -->
21 <record id="view_form_ejemplar" model="ir.ui.view">
22   <field name="name">Formulario Ejemplar</field> <!-- Nombre interno de la vista
23     -->
24   <field name="model">biblioteca.ejemplar</field> <!-- Modelo asociado -->
25   <field name="arch" type="xml">
26     <form string="Ejemplar de Cómic">
27       <group>
28         <field name="comic_id"/> <!-- Relación con el cómic
29           correspondiente -->
30         <field name="socio_id"/> <!-- Relación con el socio que lo tiene
31           prestado -->
32         <field name="estado" readonly="1"/> <!-- Estado del ejemplar (solo
33           lectura) -->
34         <field name="fecha_prestamo"/> <!-- Fecha del préstamo -->
35         <field name="fecha_devolucion_prevista"/> <!-- Fecha prevista de
36           devolución -->
37       </group>
38     </form>
39   </field>
40 </record>
41
42 <!-- Vista Lista para ver todos los ejemplares en forma de tabla -->
43 <record id="view_list_ejemplar" model="ir.ui.view">
44   <field name="name">Lista Ejemplares</field> <!-- Nombre interno de la vista
45     -->
46   <field name="model">biblioteca.ejemplar</field> <!-- Modelo asociado -->
47   <field name="arch" type="xml">
48     <list>
49       <field name="comic_id"/> <!-- Columna: Cómic -->
50       <field name="socio_id"/> <!-- Columna: Socio -->
51       <field name="estado"/> <!-- Columna: Estado -->
52       <field name="fecha_prestamo"/> <!-- Columna: Fecha de préstamo -->
53       <field name="fecha_devolucion_prevista"/> <!-- Columna: Fecha prevista
54         de devolución -->
55     </list>
56   </field>
57 </record>
58 </odoo>
```

Aquí tuve un fallo porque hay que acordarse de los modelos nuevos meterlos aquí

```
from . import biblioteca_comic
from . import biblioteca_comic_categoria
from . import biblioteca_socio
from . import biblioteca_ejemplar
```


Y esto lo mismo que hay que acordarse de meter las vistas nuevas

```
'data': [  
  'security/ir.model.access.csv',  
  'views/biblioteca_comic.xml',  
  'views/biblioteca_comic_categoria.xml',  
  'views/biblioteca_socio.xml',  
  'views/biblioteca_ejemplar.xml',  
],
```