

# **SISTEMAS DE GESTIÓN EMPRESARIAL**

**SGE05**

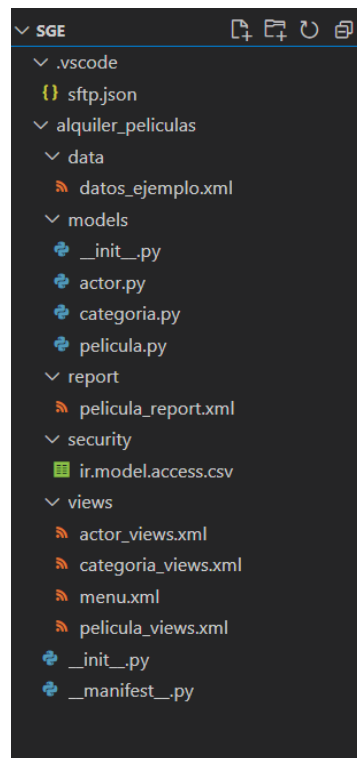
**José Mompeán Roca**

# ÍNDICE:

<b>1.-Definición de Modelos .....</b>	<b>3</b>
<b>2.-Uso de Campos Selection .....</b>	<b>6</b>
<b>3.-Definición de Campos Relacionales .....</b>	<b>7</b>
<b>4.-Vistas y Menús del Módulo .....</b>	<b>7</b>
<b>5.-Carga de Datos de Ejemplo .....</b>	<b>10</b>
<b>6.-Generación de Informes .....</b>	<b>10</b>
<b>7.-Seguridad y Permisos de Acceso.....</b>	<b>11</b>
<b>8.-Apartado Adicional .....</b>	<b>12</b>

## 1. Se deben definir al menos 3 modelos:

### a) Se organiza el módulo de manera apropiada

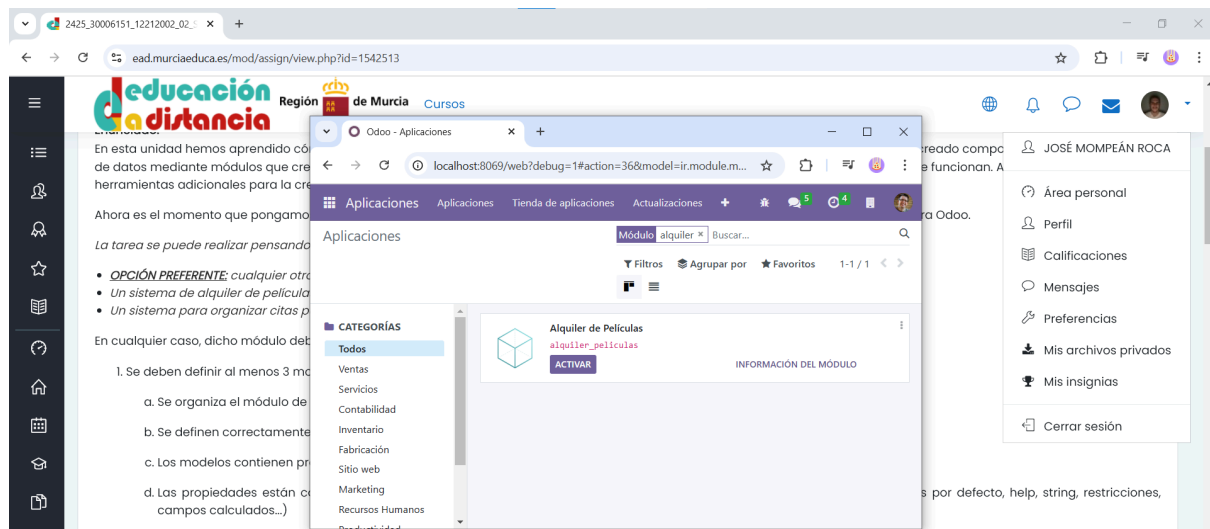


/home/odoo/modulos/alquiler_peliculas/					
Nombre	Tamaño	Modificado	Permisos	Propieta...	
..		10/02/2025 1:36:14	rw-rw-rw-	odoo	
__pycache__		16/02/2025 7:33:08	rw-r-xr-x	odoo	
data		10/02/2025 1:48:07	rw-rw-rw-	odoo	
models		16/02/2025 7:33:08	rw-rw-rw-	odoo	
report		16/02/2025 7:27:52	rw-rw-r-x	odoo	
security		10/02/2025 1:48:07	rw-rw-rw-	odoo	
views		16/02/2025 13:55:14	rw-rw-rw-	odoo	
__init__.py	1 KB	10/02/2025 1:48:07	rw-rw-rw-	odoo	
__manifest__.py	1 KB	16/02/2025 13:55:10	rw-rw-rw-	odoo	

## b) Se definen correctamente los archivos de configuración del módulo

```
alquiler_peliculas > __manifest__.py
1  {
2      'name': 'Alquiler de Películas',
3      'version': '1.0',
4      'depends': ['base', 'contacts'],
5      'author': 'José Mompeán Roca',
6      'category': 'Gestión',
7      'sequence': 1,
8      'summary': 'Sistema de gestión de alquiler de películas',
9      'description': 'Este módulo permite gestionar películas, actores y categorías para un sistema de alquiler.',
10     'data': [
11         'views/pelicula_views.xml',
12         "views/categoria_views.xml",
13         "views/actor_views.xml",
14         'views/menu.xml'
15     ],
16     'installable': True,
17     'application': True,
18     'auto_install': False
19 }
20
```

```
alquiler_peliculas > models > __init__.py
1  from . import actor
2  from . import categoria
3  from . import pelicula
```



### c) Los modelos contienen propiedades de diferentes tipos básicos.

```
alquiler_peliculas > models > actor.py > Actor
1  from odoo import models, fields # type: ignore
2
3  class Actor(models.Model):
4      _name = 'alquiler.actor'
5      _description = 'Actor'
6
7      name = fields.Char(string='Nombre', required=True)
8      fecha_nacimiento = fields.Date(string='Fecha de Nacimiento')
9      nacionalidad = fields.Char(string='Nacionalidad')
10     pelicula_ids = fields.Many2many('alquiler.pelicula', string='Películas')
```

```
alquiler_peliculas > models > categoria.py > Categoria
1  from odoo import models, fields # type: ignore
2
3  class Categoria(models.Model):
4      _name = 'alquiler.categoria'
5      _description = 'Categoría'
6
7      name = fields.Char(string='Nombre', required=True)
8      descripcion = fields.Text(string='Descripción')
9      pelicula_ids = fields.One2many('alquiler.pelicula', 'categoria_id', string='Películas')
```

```
from odoo import models, fields # type: ignore

class Pelicula(models.Model):
    _name = 'alquiler.pelicula'
    _description = 'Película'

    name = fields.Char(string='Título', required=True)
    descripcion = fields.Text(string='Descripción')
    duracion = fields.Float(string='Duración (minutos)', digits=(6, 2))
    fecha_estreno = fields.Date(string='Fecha de Estreno')
    disponible = fields.Boolean(string='Disponible', default=True)
    categoria_id = fields.Many2one('alquiler.categoria', string='Categoría')
    actor_ids = fields.Many2many('alquiler.actor', string='Actores')

    formato = fields.Selection([
        ('dvd', 'DVD'),
        ('blu-ray', 'Blu-ray'),
        ('digital', 'Digital')
    ], string='Formato', default='digital')
```

d) Las propiedades están completamente definidas haciendo uso de los parámetros de definición adecuados (required, valores por defecto, help, string, restricciones, campos calculados...)

```
alquiler_peliculas > models > pelicula.py > ...
1 from odoo import models, fields, api # type: ignore
2 from odoo.exceptions import ValidationError # type: ignore
3 class Pelicula(models.Model):
4     _name = 'alquiler.pelicula'
5     _description = 'Pelicula'
6
7     name = fields.Char(
8         string='Título',
9         required=True,
10        help="Introduce el título de la película."
11    )
12    descripcion = fields.Text(
13        string='Descripción',
14        help="Descripción general de la película."
15    )
16    duracion = fields.Float(
17        string='Duración (minutos)',
18        digits=(6, 2),
19        required=True,
20        help="Duración en minutos. Debe ser mayor que 0."
21    )
22    fecha_estreno = fields.Date(
23        string='Fecha de Estreno',
24        required=True,
25        help="Fecha en la que se estrenó la película."
26    )
27    disponible = fields.Boolean(
28        string='Disponible',
29        default=True,
30        help="Indica si la película está disponible para alquiler."
31    )
32    categoria_id = fields.Many2one(
33        'alquiler.categoria',
34        string='Categoría',
35        required=True,
36        help="Selecciona la categoría a la que pertenece la película."
37    )
```

```
    actor_ids = fields.Many2many(
        'alquiler.actor',
        string='Actores',
        help="Selecciona los actores que participan en la película."
    )
    formato = fields.Selection([
        ('dvd', 'DVD'),
        ('blu-ray', 'Blu-ray'),
        ('digital', 'Digital')
    ],
        string='Formato',
        default='digital',
        required=True,
        help="Selecciona el formato en el que está disponible la película."
    )
    @api.constrains('duracion')
    def _check_duracion(self):
        """ Validación para asegurarse de que la duración sea mayor que 0. """
        for record in self:
            if record.duracion <= 0:
                raise ValidationError("La duración debe ser mayor que 0 minutos.")
```

2. Se define al menos un campo *selection* (Podéis crear la lista de tuplas directamente en código o cargando datos desde un fichero)

```
formato = fields.Selection([
    ('dvd', 'DVD'),
    ('blu-ray', 'Blu-ray'),
    ('digital', 'Digital')
],
    string='Formato',
    default='digital',
    required=True,
    help="Selecciona el formato en el que está disponible la película."
)
```

### 3. Se definen campos *relacionales*:

#### a) Many2one

```
categoria_id = fields.Many2one(
    'alquiler.categoria',
    string='Categoria',
    required=True,
    help="Selecciona la categoría a la que pertenece la película."
)
```

#### b) One2many

```
class Categoria(models.Model):
    _name = 'alquiler.categoria'
    _description = 'Categoria'

    name = fields.Char(string='Nombre', required=True)
    descripcion = fields.Text(string='Descripción')
    pelicula_ids = fields.One2many('alquiler.pelicula', 'categoria_id', string='Películas')
```

#### c) Many2many

```
actor_ids = fields.Many2many(
    'alquiler.actor',
    string='Actores',
    help="Selecciona los actores que participan en la película."
)
```

### 4. Se definen vistas y menús para los modelos creados:

```
alquiler_peliculas > views > actor_views.xml
1  <odoo>
2  <record id="view_actor_form" model="ir.ui.view">
3  <field name="name">alquiler.actor.form</field>
4  <field name="model">alquiler.actor</field>
5  <field name="arch" type="xml">
6  <form>
7  <sheet>
8  <group>
9  <field name="name"/>
10 <field name="fecha_nacimiento"/>
11 <field name="nacionalidad"/>
12 <field name="pelicula_ids"/>
13 </group>
14 </sheet>
15 </form>
16 </field>
17 </record>
18
19 <record id="view_actor_tree" model="ir.ui.view">
20 <field name="name">alquiler.actor.tree</field>
21 <field name="model">alquiler.actor</field>
22 <field name="arch" type="xml">
23 <tree>
24 <field name="name"/>
25 <field name="nacionalidad"/>
26 </tree>
27 </field>
28 </record>
29 </odoo>
```

```

alquiler_peliculas > views > categoria_views.xml
1  <odoo>
2    <record id="view_categoria_form" model="ir.ui.view">
3      <field name="name">alquiler.categoria.form</field>
4      <field name="model">alquiler.categoria</field>
5      <field name="arch" type="xml">
6        <form>
7          <sheet>
8            <group>
9              <field name="name"/>
10             <field name="descripcion"/>
11             <field name="pelicula_ids"/>
12            </group>
13          </sheet>
14        </form>
15      </field>
16    </record>
17
18    <record id="view_categoria_tree" model="ir.ui.view">
19      <field name="name">alquiler.categoria.tree</field>
20      <field name="model">alquiler.categoria</field>
21      <field name="arch" type="xml">
22        <tree>
23          <field name="name"/>
24        </tree>
25      </field>
26    </record>
27  </odoo>

```

```

alquiler_peliculas > views > menu.xml
1  <?xml version="1.0" encoding="UTF-8"?>
2  <odoo>
3    <record id="action_actor" model="ir.actions.act_window">
4      <field name="name">Actores</field>
5      <field name="res_model">alquiler.actor</field>
6      <field name="view_mode">tree,form</field>
7      <field name="help" type="html">
8        <p>Aquí puedes gestionar los actores de las películas.</p>
9      </field>
10    </record>
11
12    <record id="action_pelicula" model="ir.actions.act_window">
13      <field name="name">Películas</field>
14      <field name="res_model">alquiler.pelicula</field>
15      <field name="view_mode">tree,form,kanban</field>
16      <field name="help" type="html">
17        <p>Aquí puedes gestionar las películas disponibles.</p>
18      </field>
19    </record>
20
21    <record id="action_categoria" model="ir.actions.act_window">
22      <field name="name">Categorías</field>
23      <field name="res_model">alquiler.categoria</field>
24      <field name="view_mode">tree,form</field>
25      <field name="help" type="html">
26        <p>Aquí puedes gestionar las categorías de películas.</p>
27      </field>
28    </record>
29
30    <menuitem id="menu_alquiler_root" name="Alquiler de Películas" sequence="1"/>
31    <menuitem id="menu_alquiler_actor" name="Actores" parent="menu_alquiler_root" action="action_actor" sequence="2"/>
32    <menuitem id="menu_alquiler_pelicula" name="Películas" parent="menu_alquiler_root" action="action_pelicula" sequence="3"/>
33    <menuitem id="menu_alquiler_categoria" name="Categorías" parent="menu_alquiler_root" action="action_categoria" sequence="4"/>
34  </odoo>
35

```





localhost:8069/web?debug=1&id=1&cids=1&menu\_id=116&action=139&model=alquiler.pelicula&view\_type=form

Alquiler de Películas Actores Películas Categorías

Películas / El Padrino

Acción 1 / 1 Nuevo

Título ? El Padrino

Descripción ?

Categoría ? Acción

Fecha de Estreno ?

Duración (minutos) ? 0,00

Disponible ? ☒

Formato ? Digital

Actores ? Marlon Brando

---

Alquiler de Películas Actores Películas Categorías

Categorías / Acción

Acción 1 / 1 Nuevo

Nombre ? Acción

Descripción ?

Películas ?

Título	Categoría	Fecha de Estreno	Disponible	Formato
El Padrino	Acción		<input checked="" type="checkbox"/>	Digital

Agregar una línea

## 5. El módulo cargará unos pocos datos de ejemplo para todos los modelos.

```
alquiler_peliculas > data > datos_ejemplo.xml
1 <odoo>
2   <record id="categoria_accion" model="alquiler.categoria">
3     <field name="name">Acción</field>
4   </record>
5
6   <record id="pelicula_1" model="alquiler.pelicula">
7     <field name="name">El Padrino</field>
8     <field name="categoria_id" ref="categoria_accion"/>
9   </record>
10
11   <record id="actor_1" model="alquiler.actor">
12     <field name="name">Marlon Brando</field>
13     <field name="pelicula_ids" eval="[(4, ref('pelicula_1'))]">
14   </record>
15 </odoo>
16
```

## 6. Definición de informes. Creación de al menos uno para alguno de los modelos definidos.

```
alquiler_peliculas > manifest.py
1
2 'name': 'Alquiler de Películas',
3 'version': '1.0',
4 'depends': ['base', 'contacts'],
5 'author': 'José Moneán Roca',
6 'category': 'Gestión',
7 'sequence': 1,
8 'summary': 'Sistema de gestión de alquiler de películas',
9 'description': 'Este módulo permite gestionar películas, actores y categorías para un sistema de alquiler.',
10 'data': [
11   'views/pelicula_views.xml',
12   'views/categoria_views.xml',
13   'views/actor_views.xml',
14   'views/menu.xml',
15   'report/pelicula_report.xml',
16 ],
17 'installable': True,
18 'application': True,
19 'auto_install': False
20
21
```

```
alquiler_peliculas > report > pelicula_report.xml
1 <odoo>
2   <report
3     id="report_pelicula"
4     model="alquiler.pelicula"
5     string="Reporte de Películas"
6     report_type="qweb-pdf"
7     file="alquiler.report_pelicula"
8     name="alquiler.report_pelicula"
9   />
10
11   <template id="report_pelicula">
12     <t t-call="web.html_container">
13       <t t-foreach="docs" t-as="pelicula">
14         <div>
15           <h2>Película: <t t-esc="pelicula.name"/></h2>
16           <p>Categoría: <t t-esc="pelicula.categoria_id.name"/></p>
17           <p>Formato: <t t-esc="pelicula.formato"/></p>
18           <p>Duración: <t t-esc="pelicula.duracion"/> minutos</p>
19           <p>Disponible: <t t-esc="pelicula.disponible"/></p>
20         </div>
21       </t>
22     </t>
23   </template>
24 </odoo>
25
```

## 7. Seguridad: definición de permisos de acceso.

```
alquiler_peliculas > security > ir.model.access.csv
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2 access_pelicula_user,Acceso Usuario Película,model_alquiler_pelicula,alquiler_peliculas.group_user,1,0,0,0
3 access_pelicula_admin,Acceso Administrador Película,model_alquiler_pelicula,alquiler_peliculas.group_admin,1,1,1,1
4 access_categoria_user,Acceso Usuario Categoría,model_alquiler_categoria,alquiler_peliculas.group_user,1,0,0,0
5 access_categoria_admin,Acceso Administrador Categoría,model_alquiler_categoria,alquiler_peliculas.group_admin,1,1,1,1
6 access_actor_user,Acceso Usuario Actor,model_alquiler_actor,alquiler_peliculas.group_user,1,0,0,0
7 access_actor_admin,Acceso Administrador Actor,model_alquiler_actor,alquiler_peliculas.group_admin,1,1,1,1
8
```

```
alquiler_peliculas > security > security.xml
1 <odoo>
2   <data noupdate="1">
3
4     <record id="group_admin" model="res.groups">
5       <field name="name">Administrador de Alquiler de Películas</field>
6       <field name="category_id" ref="base.module_category_management"/>
7       <field name="implied_ids" eval="[(4, ref('base.group_user'))]"/>
8     </record>
9
10    <record id="group_user" model="res.groups">
11      <field name="name">Usuario de Alquiler de Películas</field>
12      <field name="category_id" ref="base.module_category_management"/>
13      <field name="implied_ids" eval="[(4, ref('base.group_user'))]"/>
14    </record>
15  </data>
16 </odoo>
17
18
```

8. Se completará el módulo con algún apartado adicional no tratado en los puntos anteriores. Ejemplos: vista adicional search o kanban, filtros, hacer uso de botones en las vistas para realizar alguna acción, extensión/utilización de un modelo base del sistema (res.user, res.partner...) o de alguno de los modelos definidos, etc.

```
alquiler_peliculas > views > pelicula_views.xml
1  <odoo>
16  <record id="view_pelicula_form" model="ir.ui.view">
19    <field name="arch" type="xml">
20      <form>
21        <group>
22          <field name="name"/>
23          <field name="descripcion"/>
24          <field name="categoria_id"/>
25          <field name="fecha_estreno"/>
26          <field name="duracion"/>
27          <field name="disponible"/>
28          <field name="formato"/>
29          <field name="actor_ids" widget="many2many_tags"/>
30        </group>
31      </form>
32    </field>
33  </record>
34
35  <record id="view_pelicula_kanban" model="ir.ui.view">
36    <field name="name">pelicula.kanban</field>
37    <field name="model">alquiler.pelicula</field>
38    <field name="arch" type="xml">
39      <kanban>
40        <field name="name"/>
41        <field name="categoria_id"/>
42        <field name="formato"/>
43      </kanban>
44    </field>
45  </record>
46 </odoo>
47
```