

# DESARROLLO DE MODELOS Y VISTAS EN ODOO

<b>Gestión de una biblioteca.</b>	<b>2</b>
MODELO DE DATOS:	2
ESTRUCTURA DE UN MÓDULO EN ODOO.	2
MODELO EN Python	4
VISTAS	5
SEGURIDAD DE ACCESO AL MODELO:	9
INFORMES	11
INSTALACIÓN DEL MÓDULO	13
<b>Gestión de una biblioteca de películas.</b>	<b>14</b>
MODELO DE DATOS.	14
MODELO EN Python.	14
VISTAS.	15
SEGURIDAD DE ACCESO AL MODELO.	17
INFORMES	19
CREAR ROLES DE USUARIO	20

## Gestión de una biblioteca.

### MODELO DE DATOS:

Entidad	Atributos
Autor	nombre : Texto libros : libros que ha escrito o colaborado para su creación.
Libros	nombre : Texto precio : Float PVP : Float descripcion : Text estado : Selection {Bien conservado, Regular, Deteriorado} autores : autores que han escrito el libro.

### ESTRUCTURA DE UN MÓDULO EN ODOO.

Crear la estructura base del módulo con SCAFFOLD en la carpeta addons de Odoo:

```
sudo ../odoo-bin scaffold valle_libro
```

__init__.py	
__manifest__.py	Configuración del módulo.
controllers	Acciones de ventana.
demo	Datos de ejemplo.
models	Modelo ORM de la base de datos.
security	Permisos sobre el módulo.
views	Vistas y elementos de menú.

Modificar \_\_manifest\_\_.py para introducir los datos del módulo y descomentar la línea data de seguridad ('security/ir.model.access.csv',):

```
# -*- coding: utf-8 -*-
{
    'name': "Gestión de Biblioteca",
```

```
'summary': """" Módulo Odoo para la gestión de libros""",

'description': """" Gestión de libros """,

'author': "Rafael Reina",
'website': "http://www.iesvalleinclan.es",

# Categories can be used to filter modules in modules listing
# Check
https://github.com/odoo/odoo/blob/15.0/odoo/addons/base/data/ir_module_
category_data.xml
# for the full list
'category': 'Uncategorized',
'version': '0.1',

# any module necessary for this one to work correctly
'depends': ['base'],

# always loaded
'data': [
    'security/ir.model.access.csv',
    'views/views.xml',
    'views/templates.xml',
],
# only loaded in demonstration mode
'demo': [
    'demo/demo.xml',
],
}
```

## MODELO EN Python

Modificar models.py para crear los modelos (clases Python: atributos y métodos).

El atributo `__name` del modelo tiene el valor en formato `nombreModulo.modelo`

Estado de los libros: “Bien conservado”, “Regular”, “Deteriorado”.

Un libro está escrito por varios autores y un autor puede escribir varios libros (relación Many2many):# -\*- coding: utf-8 -\*-

```
from odoo import models, fields, api
ESTADO = [('B', 'B - Bien conservado'),
          ('R', 'R - Regular'),
          ('D', 'D - Deteriorado')]

#MODELO Libro
class libro(models.Model):
    _name = 'valle_libro.libro'
    _description = 'Libro de la Biblioteca'

    nombre = fields.Text()
    precio = fields.Float()
    PVP = fields.Float(compute="_value_pc", store=True)
    descripcion = fields.Text()
    estado = fields.Selection(ESTADO, default=ESTADO[0][0])

    autores=fields.Many2many('valle_libro.autor', string='Autores')

    @api.depends('precio')
    def _value_pc(self):
        for record in self:
            record.PVP = record.precio * 1.05

#MODELO Autor
class autor(models.Model):
    _name = 'valle_libro.autor'
    _description = 'Autor del libro'

    nombre = fields.Text()
    libros=fields.Many2many('valle_libro.libro', string='Libros')
```

## VISTAS

Carpeta views/views.xml.

Los ids de las vistas siempre deben contener el nombre en el formato nombreModulo.nombre\_vista.

Para poder tener acceso desde libros a los autores y viceversa, es necesario añadir en las vistas de ambos modelos los campos libros y autores relacionados mediante Many2many en ambos modelos:

```
<odoo>
<data>
  <!-- explicit list view definition -->

  <record model="ir.ui.view" id="valle_libro.libro_list">
    <field name="name">Libros</field>
    <field name="model">valle_libro.libro</field>
    <field name="arch" type="xml">
      <tree>
        <field name="nombre"/>
        <field name="PVP"/>
        <field name="autores"/>
      </tree>
    </field>
  </record>

  <record model="ir.ui.view" id="valle_libro.autor_list">
    <field name="name">Autores</field>
    <field name="model">valle_libro.autor</field>
    <field name="arch" type="xml">
      <tree>
        <field name="nombre"/>
        <field name="libros"/>
      </tree>
    </field>
  </record>

  <!-- actions opening views on models -->

  <record model="ir.actions.act_window"
id="valle_libro.action_window_libro">
    <field name="name">Lista de Libros</field>
```

```
<field name="res_model">valle_libro.libro</field>
<field name="view_mode">tree,form</field>
</record>

<record model="ir.actions.act_window"
id="valle_libro.action_window_autor">
  <field name="name">Lista de Autores</field>
  <field name="res_model">valle_libro.autor</field>
  <field name="view_mode">tree,form</field>
</record>

<!-- server action to the one above -->
<!--
<record model="ir.actions.server" id="valle_libro.action_server">
  <field name="name">SERVIDOR</field>
  <field name="model_id" ref="model_valle_libro_libro"/>
  <field name="state">code</field>
  <field name="code">
    action = {
      "type": "ir.actions.act_window",
      "view_mode": "tree,form",
      "res_model": model._name,
    }
  </field>
</record>

-->

<!-- Top menu item -->

<menuitem name="LIBROS" id="valle_libro.menu_root"/>

<!-- menu categories -->

<menuitem name="Biblioteca" id="valle_libro.menu1"
parent="valle_libro.menu_root"/>

<!-- actions -->

<menuitem name="Lista DE LIBROS" id="valle_libro.menu_libro"
```

```
parent="valle_libro.menu1"
    action="valle_libro.action_window_libro"/>

    <menuitem name="Lista de autores" id="valle_libro.menu_autor"
parent="valle_libro.menu1"
    action="valle_libro.action_window_autor"/>

</data>
</odoo>
```

Odoo - Nuevo - Chromium

Odoo - Nuevo

localhost:8069/web#ids=1&menu\_id=234&action=316&model=valle\_libro.libro&view\_type=form

LIBROS Biblioteca

Administrator (AdaptacionVentas)

Lista de Libros / Nuevo

Guardar Descartar

Nombre	<input type="text"/>										
Precio	<input type="text" value="0,00"/>	Pvp	<input type="text" value="0,00"/>								
Descripción	<input type="text"/>										
Estado	<input type="text" value="B - Bien conservado"/>										
Autores	<table> <thead> <tr> <th>Nombre</th> <th>Libros</th> </tr> </thead> <tbody> <tr> <td colspan="2">Agregar línea</td> </tr> <tr> <td><input type="text"/></td> <td><input type="text"/></td> </tr> <tr> <td><input type="text"/></td> <td><input type="text"/></td> </tr> </tbody> </table>			Nombre	Libros	Agregar línea		<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Nombre	Libros										
Agregar línea											
<input type="text"/>	<input type="text"/>										
<input type="text"/>	<input type="text"/>										

Odoo - Nuevo - Chromium

Odoo - Nuevo

localhost:8069/web#cid=1&menu\_id=234&action=317&model=valle\_libro.autor&view\_type=form

LIBROS Biblioteca

Administrator (AdaptacionVentas)

Lista de Autores / Nuevo

Guardar Descartar

Nombre

Libros

Nombre	Pvp	Autores
Agregar linea		



## SEGURIDAD DE ACCESO AL MODELO:

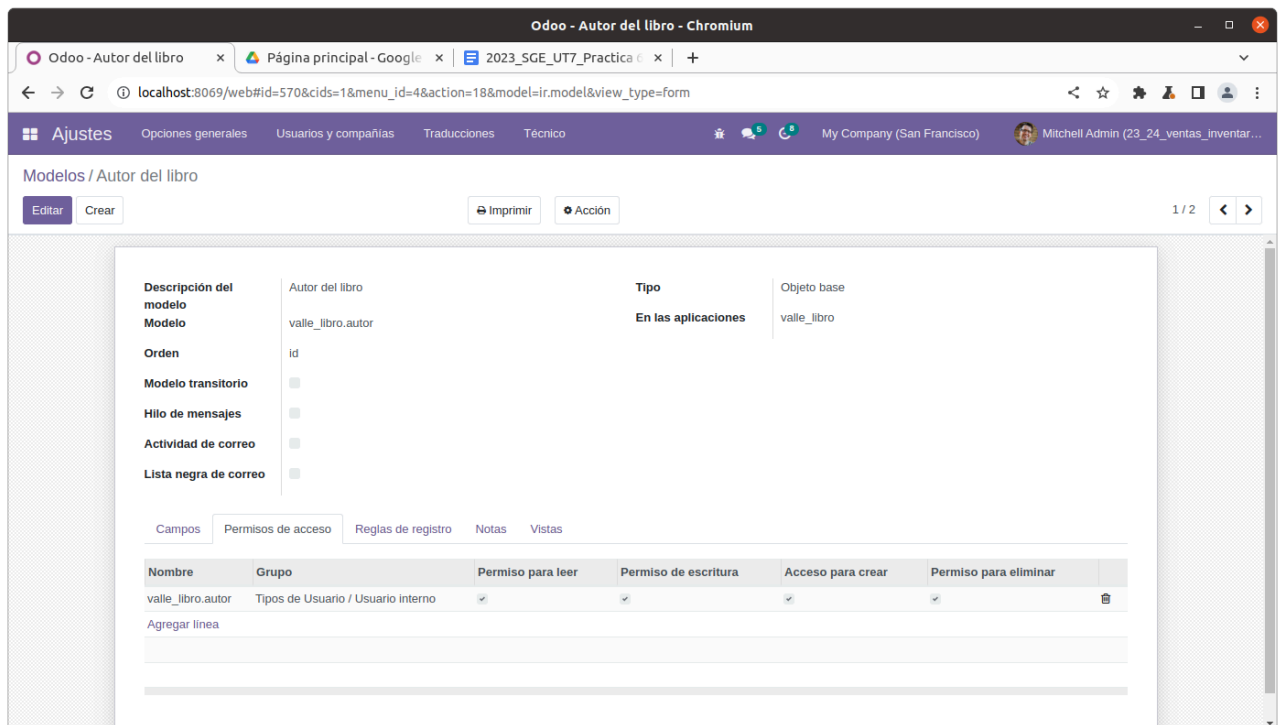
Modificar el fichero security/ir.model.access.csv para permitir el acceso desde las vistas a ambos modelos:

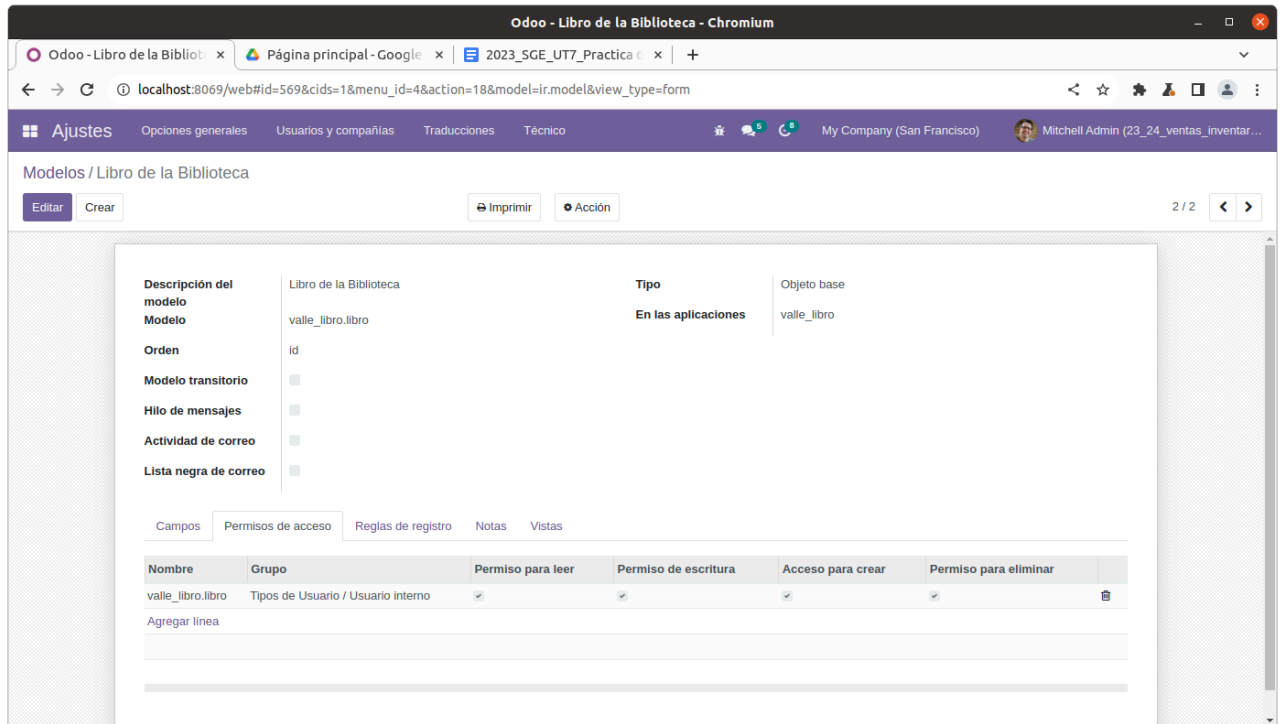
id,name,model\_id:id,group\_id:id,perm\_read,perm\_write,perm\_create,perm\_unlink

access\_valle\_libro\_libro, valle\_libro.libro, model\_valle\_libro\_libro, base.group\_user, 1, 1, 1, 1

access\_valle\_libro\_autor, valle\_libro.autor, model\_valle\_libro\_autor, base.group\_user, 1, 1, 1, 1

En la interfaz de Odoo se vería de la siguiente manera:





Por último hay que modificar el `__manifest__.py` para que se pueda cargar el fichero de seguridad anterior.

# always loaded

```
'data': [
    'security/ir.model.access.csv',
    'views/views.xml',
    'views/templates.xml',
],
```

## INFORMES

Informe de libros. (reports/informe\_libros.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<odoo>
  <report
    id="valle_libro.libro"
    string="Informe Libro"
    model="valle_libro.libro"
    report_type="qweb-pdf"
    name="valle_libro.informe_libro_view"
    file="valle_libro.informe_libro_view"/>
  <template id="informe_libro_view">
    <t t-call="web.basic_layout">
      <t t-foreach="docs" t-as="libro">
        <div class="page">
          <div>
            <strong>Nombre:</strong>
            <span t-field="libro.nombre"/>
          </div>
          <div>
            <strong>Descripcion:</strong>
            <span t-field="libro.descripcion"/>
          </div>
          <div>
            <strong>Precio con IVA:</strong>
            <span t-field="libro.PVP"/>
          </div>
          <h2>AUTORES</h2>
          <hr></hr>
          <t t-foreach="libro.autores" t-as="autor">
            <div>
              <strong>Nombre:</strong>
              <span t-field="autor.nombre"/>
            </div>
          </t>
        </div>
      </t>
    </template>
  </odoo>
```

Añadimos el informe al manifest:

```
'data': [  
    'reports/informe_libros.xml',  
    ...  
]
```

Libros escritos por un autor.

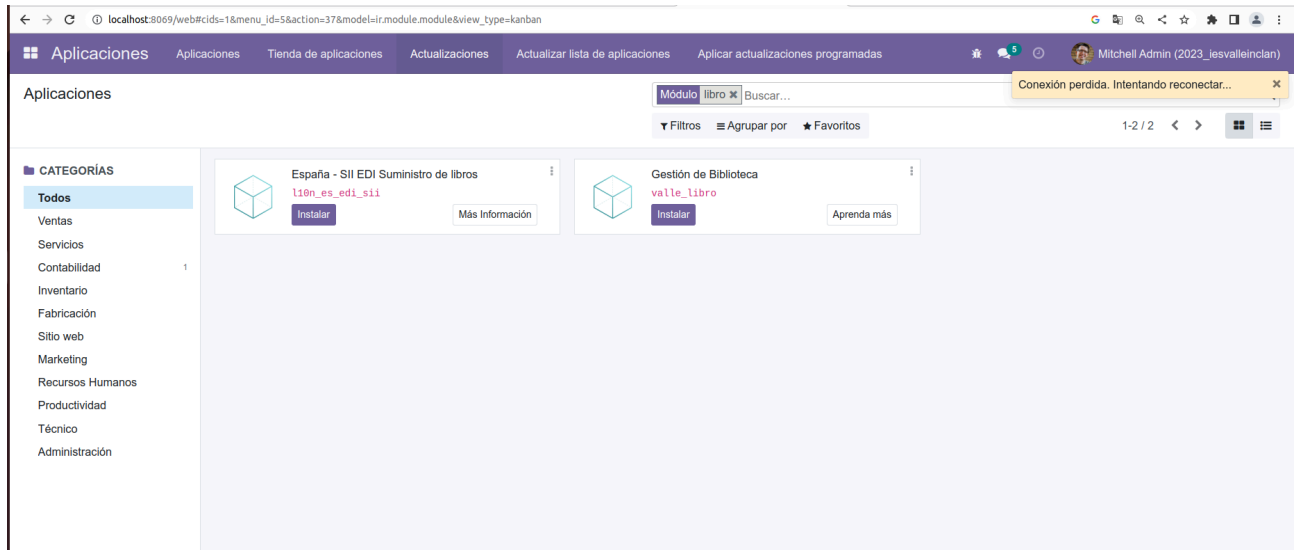
## INSTALACIÓN DEL MÓDULO

Reiniciar Odoo.

Ajustes → Activar modo desarrollador

Aplicaciones → Actualizar lista de aplicaciones

Buscar el módulo en la lista:



## Gestión de una biblioteca de películas.

### MODELO DE DATOS.

Entidad	Datos
Género	nombre : texto películas: película de la base de datos que pertenecen a este género
Película	director : Texto color : si es en color o B/N duracion : Duración en minutos industria : Selection {"Hollywood", "Europea", "Otras"} fecha : "Fecha de estreno de la película" sinopsis : Texto genero : género al que pertenece la película

### MODELO EN Python.

#### PELÍCULA

```
class lapeliculera_pelicula(models.Model):
    _name = 'lapeliculera.pelicula'
    _description = 'Película'

    name = fields.Char(string="Titulo", required=True, help="Introduce
el título de la película")
    director =
fields.Char(string="Director", required=True, help="Nombre del director")
    color = fields.Boolean(string="Color")
    duracion = fields.Integer(string="Duración en minutos")
    industria = fields.Selection([("0", "Hollywood"), ("1",
"Europea"), ("2", "Otras")])
    fecha = fields.Date(string="Fecha de estreno de la película")
    sinopsis = fields.Text(string="Sinopsis")

    #relación entre modelos: toda película pertenece a un género
    (Many2one)
    genero =
fields.Many2one('lapeliculera.genero', string="Género", required=True)
```

## GÉNERO

```
class lapeliculera_genero(models.Model):
    _name = 'lapeliculera.genero'
    _description = 'Género cinematográfico'

    name = fields.Char(string="Género", required=True, help="Introduce el género cinematográfico de la película")
    comentario = fields.Text(string="Comentarios")

    #relación entre modelos: varias películas pueden pertenecer a un mismo género
    #(One2many hay que especificar el campo de relación definido en la otra tabla como Many2one)
    pelicula = fields.One2many('lapeliculera.pelicula', "genero", string="Películas")
```

## VISTAS.

```
<odoo>
<data>
    <!-- explicit list view definition -->

    <record model="ir.ui.view" id="lapeliculera.genero_tree">
        <field name="name">Lista de Géneros</field>
        <field name="model">lapeliculera.genero</field>
        <field name="arch" type="xml">
            <tree>
                <field name="name"/>
                <field name="comentario"/>
            </tree>
        </field>
    </record>

    <record model="ir.ui.view" id="lapeliculera.genero_form">
        <field name="name">lapeliculera.genero.form</field>
        <field name="model">lapeliculera.genero</field>
        <field name="arch" type="xml">
            <form>
                <group colspan="2" col="2">
                    <field name="name"/>
                    <field name="comentario"/>
                </group>
            </form>
        </field>
    </record>
```

```

        </group>
    </form>
</field>
</record>

<record model="ir.ui.view" id="lapeliculera.pelicula_tree">
    <field name="name">Lista de Películas</field>
    <field name="model">lapeliculera.pelicula</field>
    <field name="arch" type="xml">
        <tree>
            <field name="name"/>
            <field name="director"/>
            <field name="duracion"/>
        </tree>
    </field>
</record>

<record model="ir.ui.view" id="lapeliculera.pelicula_form">
    <field name="name">lapeliculera.pelicula.form</field>
    <field name="model">lapeliculera.pelicula</field>
    <field name="arch" type="xml">
        <form>
            <group colspan="2" col="2">
                <field name="name"/>
                <field name="director"/>
                <field name="duracion"/>
            </group>
        </form>
    </field>
</record>

<!-- actions opening views on models -->

<record model="ir.actions.act_window"
id="lapeliculera.genero_action_window">
    <field name="name">lapeliculera.genero_action_window</field>
    <field name="res_model">lapeliculera.genero</field>
    <field name="view_mode">tree,form</field>
</record>

<record model="ir.actions.act_window"

```



```
id="lapeliculera.pelicula_action_window">
    <field name="name">lapeliculera.pelicula_action_window</field>
    <field name="res_model">lapeliculera.pelicula</field>
    <field name="view_mode">tree,form</field>
</record>
<!-- menu categories -->
    <menuitem name="Películas" id="lapeliculera.menu1"
parent="lapeliculera.menu_root"/>

<!-- actions -->

    <menuitem name="Género" id="lapeliculera.menu_genero"
parent="lapeliculera.menu1"
action="lapeliculera.genero_action_window"/>
    <menuitem name="Película" id="lapeliculera.menu_pelicula"
parent="lapeliculera.menu1"
action="lapeliculera.pelicula_action_window"/>

</data>
</odoo>
```

## SEGURIDAD DE ACCESO AL MODELO.

Se le deben dar permisos a los usuarios para acceder al modelo y que el módulo funcione correctamente al instalarlo en el sistema. Para ello es necesario modificar el fichero del directorio security/ir.model.access.csv

id,name,model\_id:id,group\_id:id,perm\_read,perm\_write,perm\_create,perm\_unlink

access\_lapeliculera\_pelicula,lapeliculera.pelicula,model\_lapeliculera\_pelicula,base.group\_user,1,1,1,1

access\_lapeliculera\_genero,lapeliculera.genero,model\_lapeliculera\_genero,base.group\_user,1,1,1,1

Odoo - Género cinematográfico - Chromium

Odoo - Género cinematográfico

Modelos / Género cinematográfico

Editar Crear

Imprimir Acción

1 / 2

Descripción del modelo

Modelo

Orden

Modelo transitorio

Hilo de mensajes

Actividad de correo

Lista negra de correo

Género cinematográfico

lapeliculera.genero

id

☐

☐

☐

☐

Tipo

En las aplicaciones

Objeto base

lapeliculera

Campos Permisos de acceso Reglas de registro Notas Vistas

Nombre	Grupo	Permiso para leer	Permiso de escritura	Acceso para crear	Permiso para eliminar
usuario	Tipos de Usuario / Usuario interno	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Agregar línea

Odoo - Película

Odoo - Película

Modelos / Película

Editar Crear

Imprimir Acción

2 / 2

Descripción del modelo

Modelo

Orden

Modelo transitorio

Hilo de mensajes

Actividad de correo

Lista negra de correo

Película

lapeliculera.pelicula

id

☐

☐

☐

☐

Tipo

En las aplicaciones

Objeto base

lapeliculera

Campos Permisos de acceso Reglas de registro Notas Vistas

Nombre	Grupo	Permiso para leer	Permiso de escritura	Acceso para crear	Permiso para eliminar
usuario	Tipos de Usuario / Usuario interno	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Agregar línea

Por último hay que modificar el `__manifest__.py` para que se pueda cargar el fichero de seguridad anterior.

```
# always loaded
'data': [
    'security/ir.model.access.csv',
    'views/views.xml',
    'views/templates.xml',
```

],

## INFORMES

Informe de libros. (reports/informe\_pelicula.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<odoo>
  <report
    id="lapeliculera.informe_pelicula"
    string="Informe Pelicula"
    model="lapeliculera.pelicula"
    report_type="qweb-pdf"
    name="lapeliculera.informe_pelicula_view"
    file="lapeliculera.informe_pelicula_view"/>
  <template id="informe_pelicula_view">
    <t t-call="web.basic_layout">
      <t t-foreach="docs" t-as="pelicula">
        <div class="page">
          <h1 t-field="pelicula.name"/>
          <div>
            <strong>Dirigida por:</strong>
            <span t-field="pelicula.director"/>
          </div>
          <div>
            <strong>Pertenece al género:</strong>
            <span t-field="pelicula.genero"/>
          </div>
          <div>
            <strong>Sinopsis:</strong>
            <span t-field="pelicula.sinopsis"/>
          </div>
        </div>
      </t>
    </t>
  </template>
</odoo>
```

Hay que modificar el fichero `__manifest__.py` para que Odoo localice la ubicación de los informes definidos.

```
cd securit'data': [  
    'security/ir.model.access.csv',  
    'views/views.xml',  
    'views/templates.xml',  
    'reports/informe_pelicula.xml',  
    ],
```

## CREAR ROLES DE USUARIO

Creamos el fichero security/security.xml

```
<?xml version="1.0" encoding="utf-8"?>  
<odoo>  
    <record id="lapeliculera_jefe" model="res.groups">  
        <field name="name">Lapeliculera / Jefe</field>  
    </record>  
    <record id="lapeliculera_becario" model="res.groups">  
        <field name="name">Lapeliculera / Becario</field>  
    </record>  
</odoo>
```

Actualizamos el fichero de permisos para reflejar estos nuevos roles se modifica el fichero del directorio security/ir.model.access.csv:

```
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_u  
nlink  
access_lapeliculera_genero_jefe,Genero  
Jefe,model_lapeliculera_genero,lapeliculera_jefe,1,1,1,1  
access_lapeliculera_pelicula_jefe,Pelicula  
Jefe,model_lapeliculera_pelicula,lapeliculera_jefe,1,1,1,1  
access_lapeliculera_genero_becario,Genero  
Becario,model_lapeliculera_genero,lapeliculera_becario,1,0,0,0  
access_lapeliculera_pelicula_becario,Pelicula  
Becario,model_lapeliculera_pelicula,lapeliculera_becario,1,0,0,0
```

Modificamos también el `__manifest__.py` para indicar a Odoo dónde se ubica el fichero de definición de roles:

```
'data': [  
    'security/security.xml',
```

```
'security/ir.model.access.csv',
'views/views.xml',
'views/templates.xml',
'reports/informe_pelicula.xml',
],
```

El modelo queda de la siguiente forma:

**Descripción del modelo**

**Modelo**

**Orden**

**Modelo transitorio**

**Hilo de mensajes**

**Actividad de correo**

**Lista negra de correo**

Pelicula

lapeliculera.pelicula

id

☐

☐

☐

☐

**Tipo**

**En las aplicaciones**

Objeto base

lapeliculera

Campos

Permisos de acceso

Reglas de registro

Notas

Vistas

Nombre	Grupo	Permiso para leer	Permiso de escritura	Acceso para crear	Permiso para eliminar	
Genero Becario	Lapeliculera / Becario	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Pelicula Becario	Lapeliculera / Becario	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
Genero Jefe	Lapeliculera / Jefe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	
Pelicula Jefe	Lapeliculera / Jefe	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	

[Agregar línea](#)

Para trabajar con el módulo es necesario crear un usuario en Odoo con esos roles:

Odoo - Mitchell Admin (copia) - Chromium

Recibidos [1,256] - rafael... 23\_24\_SemiasPracticas... 2023\_SCE\_U77\_Practica... Odoo - Mitchell Admin (copia) Informe Película.pdf Sinbeta

localhost:8065/web/ids-8&id=1&menu\_id=4&action=72&model=users&view\_type=form

Ajustes Opciones generales Usuarios y compañías Trabajadores Técnico

Usuarios / Mitchell Admin (copia)

Transferir Descartar

21/2

Acceso a la función de exportación	<input checked="" type="checkbox"/>	Direcciones en los pedidos de venta	<input type="checkbox"/>
Listas de precios avanzadas	<input type="checkbox"/>	Permitir la gestión de redondeos de efectivo	<input type="checkbox"/>
Contabilidad analítica	<input type="checkbox"/>	Etiquetas de contabilidad analítica	<input type="checkbox"/>
Lista de precios básicos	<input type="checkbox"/>	Descuentos en línea	<input type="checkbox"/>
Mostrar reporte de recepción en la validación	<input type="checkbox"/>	Mostrar Numero de Lot/serie en los Albaranes	<input type="checkbox"/>
Mostrar el número de serie y lote en las facturas	<input type="checkbox"/>	Mostrar Incoterms en los Pedidos de Ventas y en facturas relacionadas	<input type="checkbox"/>
Buscar pedidos confirmados	<input type="checkbox"/>	Editar de plantillas de correo	<input checked="" type="checkbox"/>
Gestionar diferentes propietarios de existencias	<input type="checkbox"/>	Administrar lotes / números de serie	<input checked="" type="checkbox"/>
Gestionar múltiples ubicaciones de almacen	<input type="checkbox"/>	Gestionar múltiples unidades de medida	<input type="checkbox"/>
Gestionar múltiples almacenes	<input type="checkbox"/>	Gestionar paquetes	<input type="checkbox"/>
Administrar empaquetado del producto	<input type="checkbox"/>	Administrar Variantes de Productos	<input checked="" type="checkbox"/>
Gestionar flujos de inventario push y pull	<input type="checkbox"/>	Gestionar las categorías de almacenamiento	<input type="checkbox"/>
Facturas pre-forma	<input type="checkbox"/>	Recibo de compra	<input type="checkbox"/>
Plantillas de presupuesto	<input checked="" type="checkbox"/>	Requerir una firma en sus pedidos de entrega	<input type="checkbox"/>
Recibo de venta	<input type="checkbox"/>	Impuesto B2B	<input checked="" type="checkbox"/>
Impuesto B2C	<input type="checkbox"/>	Utilizar reporte de recepción	<input type="checkbox"/>
Utilizar recedencia por ota	<input type="checkbox"/>		
Serie Report			
Creación de cartación	<input checked="" type="checkbox"/>	Multi-campaña	<input checked="" type="checkbox"/>
Multitenencia	<input type="checkbox"/>		
Otro			
Lapeleriera / Bencito	<input type="checkbox"/>	Lapeleriera / Zule	<input checked="" type="checkbox"/>