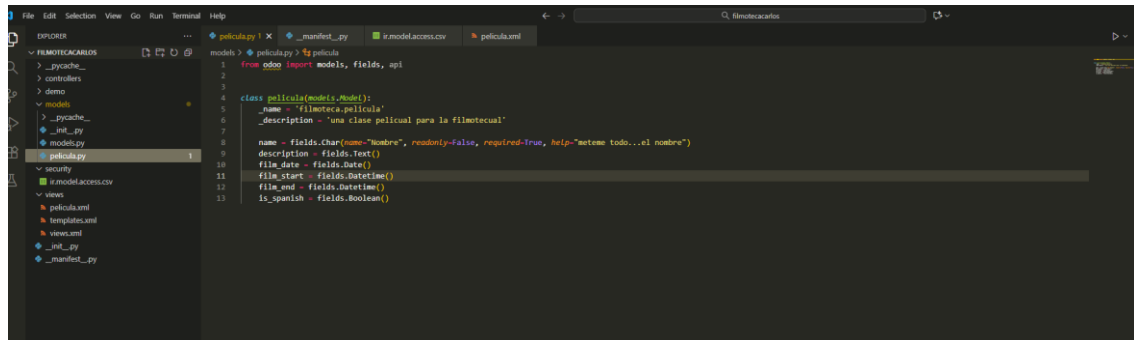


## PRÁCTICA 7 – AÑADIR MODELO AL MÓDULO FILMOTECA

Nombre...Carlos Barcenilla Villalon

.....Fecha.....

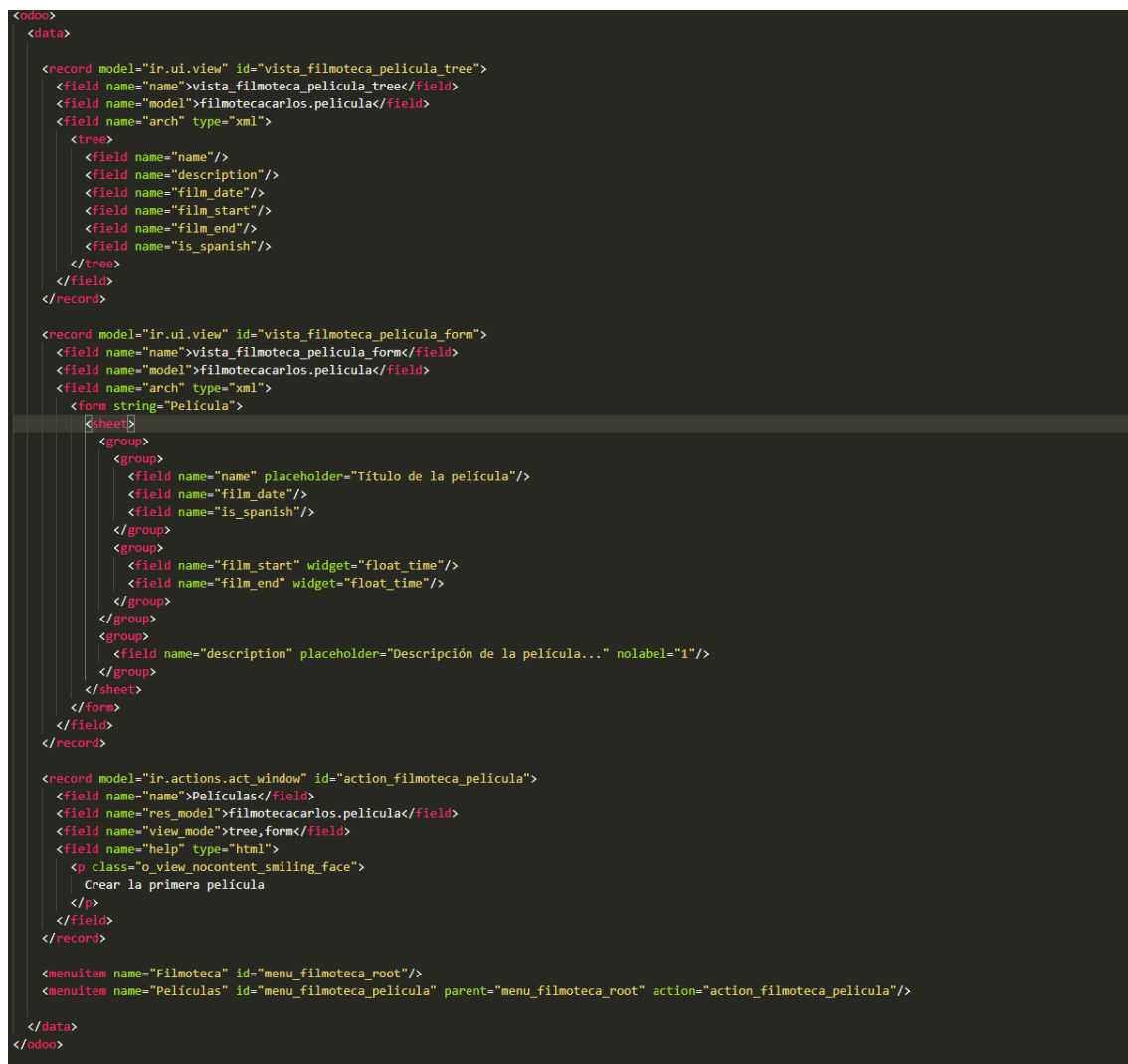
NO ESTUVE EN CLASE Y HE TENIDO QUE REALIZAR LA PRACTICA DE LA PELICULA PRIMERO



```

1 from django.db import models, fields, api
2
3
4 class Pelicula(models.Model):
5     name = 'filmoteca.pelicula'
6     description = 'una clase peli para la filmoteca'
7
8     name = fields.CharField(max_length=100, required=True, help_text="Introduce todo...el nombre")
9     description = fields.TextField()
10    film_date = fields.DateField()
11    film_start = fields.DateTimeField()
12    film_end = fields.DateTimeField()
13    is_spanish = fields.BooleanField()

```



```

<odoo>
<data>

<record model="ir.ui.view" id="vista_filmoteca_pelicula_tree">
    <field name="name">vista_filmoteca_pelicula_tree</field>
    <field name="model">filmotecacarlos.pelicula</field>
    <field name="arch" type="xml">
        <tree>
            <field name="name"/>
            <field name="description"/>
            <field name="film_date"/>
            <field name="film_start"/>
            <field name="film_end"/>
            <field name="is_spanish"/>
        </tree>
    </field>
</record>

<record model="ir.ui.view" id="vista_filmoteca_pelicula_form">
    <field name="name">vista_filmoteca_pelicula_form</field>
    <field name="model">filmotecacarlos.pelicula</field>
    <field name="arch" type="xml">
        <form string="Película">
            <sheet>
                <group>
                    <group>
                        <field name="name" placeholder="Título de la película"/>
                        <field name="film_date"/>
                        <field name="is_spanish"/>
                    </group>
                    <group>
                        <field name="film_start" widget="float_time"/>
                        <field name="film_end" widget="float_time"/>
                    </group>
                </group>
                <group>
                    <field name="description" placeholder="Descripción de la película..." nolabel="1"/>
                </group>
            </sheet>
        </form>
    </field>
</record>

<record model="ir.actions.act_window" id="action_filmoteca_pelicula">
    <field name="name">Películas</field>
    <field name="res_model">filmotecacarlos.pelicula</field>
    <field name="view_mode">tree,form</field>
    <field name="help" type="html">
        <p class="o_view_nocontent_smiling_face">
            Crear la primera película
        </p>
    </field>
</record>

<menuitem name="Filmoteca" id="menu_filmoteca_root"/>
<menuitem name="Películas" id="menu_filmoteca_pelicula" parent="menu_filmoteca_root" action="action_filmoteca_pelicula"/>

</data>
</odoo>

```

```
'name': "filmotecacarlos",

'summary': "Una filмотeca",

'description': """
Long description of module's purpose
""",

'author': "My Company",
'website': "https://www.yourcompany.com",

# 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/pelicula.xml',
    'views/views.xml',
    'views/templates.xml',
],
# only loaded in demonstration mode
'demo': [
    'demo/demo.xml',
],
```

Odoo facilita el desarrollo de **módulos** porque, además de un ERP, es un framework de programación.

El ORM (Object-Relational Mapping) de Odoo es uno de los componentes más importantes del framework, ya que permite interactuar con bases de datos de manera sencilla a través de clases y objetos de Python, sin necesidad de escribir directamente consultas SQL.

En Odoo, cada **tabla de la BBDD** → **clase Python** que hereda de **models.Model**.

### Clases → modelos

En el ejemplo que estamos desarrollando: el modelo **"película"** define una tabla llamada **"filмотeca.película"** en la BBDD.

**DENTRO DE UN MÓDULO HAY MODELOS:** en nuestro caso el módulo es **filмотeca** y el modelo **película**.

Los campos **name**, **description**, **film\_date**... representan las columnas de la tabla

```
from odoo import models, fields, api

class pelicula(models.Model):
    _name = 'filмотeca.película'
    _description = 'filмотeca.película'

    name = fields.Char(string="Nombre", readonly = False, required=True, help="Introduzca el nombre")
    description = fields.Text()
    film_date = fields.Date()
    start_date = fields.Datetime()
    end_date = fields.Datetime()
    is_spanish = fields.Boolean()
```

Este sistema modular funciona porque, cada vez que se reinicia el servidor o se actualiza un módulo, se interpretan los ficheros Python que definen los modelos y el ORM mapea las novedades en la base de datos. Además, se cargan los datos de los ficheros XML en la base de datos y se actualizan los datos que han cambiado.

Cuando se crea un módulo con el comando scaffold:

**\_\_manifest\_\_.py** → informa a Odoo de qué hace el módulo y cómo interactuar con otros módulos

**\_\_init\_\_.py** → indica a Odoo qué archivos se deben cargar para instalar el módulo o ejecutarlo

Usando los apuntes de la Unidad 4, realizar lo siguiente:

1. En el módulo **filmotecanombre**, añadir un nuevo modelo, que se llame “**genero**” (sin acentos). Como mínimo, debe contener los atributos `_name` y `_description` y los campos `name` donde se va a guardar el género de la película (intriga, infantil...) y `description`. **Crear en un fichero independiente llamado genero.py**
2. Realizar todo el proceso para crear una vista para añadir los géneros de las películas (vista formulario) y otra para poder visualizarlos en un listado (vista árbol). En las vistas, añadir los 2 campos del modelo (`name` y `description`). **Crear en un fichero independiente llamado genero.xml**
3. Crear un submenú por debajo del raíz, llamado “Listado de Géneros” y una entrada en ese menú que se llame “Consultar géneros”.

**Se deben utilizar ficheros independientes para los modelos y para las vistas.**

**Usar la notación que hemos visto en los apuntes de la Unidad 4 para los nombres de moldelos, vistas...**

4. Hacer capturas de lo siguiente:

Código del modelo “genero”	
Contenido fichero <code>__init__.py</code>	
Código vista tree	
Código vista form	
Código de las acciones de las vistas	
Código de los menús de las vistas	

Contenido fichero __manifest__.py	
Contenido fichero ir.model.access.csv	
Formulario vista tree con datos	
Formulario vista form con datos	
Estructura de carpetas del módulo <b>filmotecanombre</b> . Se tiene que ver el nombre del módulo y los ficheros generados en esta práctica (clase y vistas)	

5. Siguiendo con el módulo **filmotecanombre** que estamos haciendo, añadir una serie de campos al modelo **pelicula**: **image** para mostrar el cartel de la película, **language** (desde un conjunto de valores (español, inglés, alemán, francés; valor por defecto: español), **opinion** (desde un conjunto de valores: 0, mala; 1, regular; 2, buena; valor por defecto: regular) y un campo **color** para indicar si es en **blanco y negro o color**. Consultar este enlace:

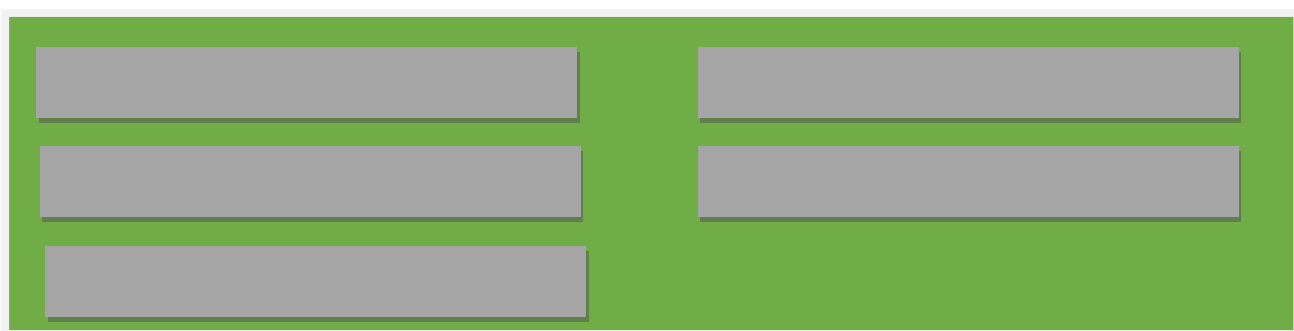
<https://odoo-development.readthedocs.io/en/latest/dev/py/fields.html#>

El campo Image hereda de Binary. Se usa para subir archivos; permite atributos propios de las imágenes (max\_width, max\_height...)

**Reiniciar Odoo** → Navegador → Settings → Con el “Modo Desarrollador” activado (Navegador → Settings → Activar modo desarrollador) → “Actualizar la lista de aplicaciones” → Eliminar los filtros de búsqueda por defecto y buscar “filmoteca”.

**Actualizar el módulo**

6. Después de añadir los campos anteriores al modelo **pelicula**, habría que modificar la vista formulario para mostrar los nuevos campos. Diseñar un formulario de la siguiente forma (2 zonas de tipos columna)



El diagrama muestra un formulario con un fondo verde. Se dividen en dos columnas por una línea vertical. Hay tres filas de campos rectangulares grises. La primera fila tiene un campo en la columna izquierda y uno en la columna derecha. La segunda fila también tiene un campo en cada columna. La tercera fila tiene un solo campo que se extiende por toda la anchura de las dos columnas.

La estructura del diseño elegido sería:

```
<form string="">
  <sheet>
    <group name="group_top">
      <group name="group_left">
        <field name=""/>
        <field name=""/>
      </group>
      <group name="group_right">
        <field name=""/>
        <field name=""/>
      </group>
    </group>
  </sheet>
</form>
```

7. Ahora hay que modificar el fichero **views/pelicula.xml** y añadir los nuevos campos.

**Reiniciar Odoo** → Navegador → Settings → Con el “Modo Desarrollador” activado (Navegador → Settings → Activar modo desarrollador) → “Actualizar la lista de aplicaciones” → Eliminar los filtros de búsqueda por defecto y buscar “filmoteca”.

**Actualizar el módulo**

8. Hacer capturas de lo siguiente:

Código del modelo “película”; con todos los campos” añadidos	
Código vista tree	
Código vista form	
Formulario vista form con datos	

**Formato de entrega:** [fichero pdf en Aula Virtual](#)

**Nombre de fichero:** [Apellido1\\_Apellido2\\_Nombre\\_Practica7.pdf](#)

