

UT5 DESARROLLO DE MÓDULOS EN ODOO

MÓDULO LIBRERÍA (Relaciones entre modelos)

Tabla de contenido

1. EXPLICACIÓN DEL MÓDULO.....	2
2. AMDAMIAJE PARA EL MÓDULO	2
3. FICHERO MANIFEST	3
4. MODELOS	4
5. FICHERO DE ACCESOS	6
6. VISTAS Y ACCIONES	6
7. CARGAR DATOS DEMO	10
8. ACTUALIZAR LOS CAMBIOS DEL MÓDULO	13
9. TABLAS CREADAS.	13
10. AMPLIACIÓN.....	14

1. EXPLICACIÓN DEL MÓDULO

En este módulo completo vamos a ver la gestión de una librería donde tendremos libros, autores y géneros.

Este módulo nos servirá para aprender a usar las relaciones más habituales que podemos encontrar entre modelos y en que tablas y relaciones se convierten.

Explicación:

- **Relación Autor-Libro:** Un autor puede tener varios libros, pero cada libro tiene un único autor. (En el ejemplo anterior vimos como un libro podía tener muchos autores por usar por primera vez una relación)
- **Relación Libro-Género:** Un libro puede pertenecer a varios géneros, y un género puede estar asociado a múltiples libros. Esto se implementará mediante la tabla intermedia transparente para el programador que veremos más adelante.

Si pensáramos en el modelo relacional veríamos claramente las siguientes tablas :

1. **Autor**

- id (PK)
- nombre (Nombre del autor)

2. **Género**

- id (PK)
- nombre (Nombre del género)

3. **Libro**

- id (PK)
- titulo (Título del libro)
- autor_id (FK → Autor.id)

4. **Libro_Género** (tabla intermedia para la relación muchos-a-muchos entre libros y géneros)

- id (PK)
- libro_id (FK → Libro.id)
- genero_id (FK → Género.id)

2. AMDAMIAJE PARA EL MÓDULO

Te aconsejo que uses el comando aprendido en el primer módulo para crear todos los archivos y carpetas necesarias `odoo scaffold`. A partir de la estructura creada vamos a modificar algunos archivos y crear otros.

3. FICHERO MANIFEST

En nuestro archivo `manifest` vamos a completar los datos más básicos y nos vamos a fijar en:

- La línea del fichero de accesos que debe estar descomentada.
- Vamos a definir las vistas y el menú según los modelos para tenerlas separadas.
- Vamos a cargar un fichero `demo.xml` con datos de muestra.

```
# -*- coding: utf-8 -*-
{
    'name': "Libreria Bosco (Relaciones)",
    'summary': "Programa de gestión de libreria IES Juan Bosco",
    'description': """
Programa de gestión de libreria IES Juan Bosco.
Este programa permitirá gestionar tanto los libros, autores y géneros de nuestra biblioteca
""",

    'author': "Fernando del Fresno",
    'website': "https://www.mywebsite.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/libro.xml',
        'views/autor.xml',
        'views/genero.xml',
        'views/menus.xml',
        'views/templates.xml',
    ],
    # only loaded in demonstration mode
    'demo': [
        'demo/demo.xml',
    ],
}
```

4. MODELOS

En el archivo `_init_.py` de la carpeta `model` debemos añadir la importación de los modelos separados en diferentes archivos

```
# -*- coding: utf-8 -*-  
  
from . import autor  
from . import libro  
from . import genero
```

`libro.py`

```
# -*- coding: utf-8 -*-  
from odoo import models, fields, api  
  
class Libro(models.Model):  
    _name = 'libreria.libro'  
    _description = 'Libro'  
  
    name = fields.Char(string='Título', required=True)  
    fecha_publicacion = fields.Date(string='Fecha de Publicación')  
    isbn = fields.Char(string='ISBN')  
    descripcion = fields.Text(string='Descripción')  
    portada = fields.Image(max_width=200, max_height=200)  
  
    # RELACIONES  
    # Un libro pertenece a un autor y muchos libros pueden ser de un autor  
    autor_id = fields.Many2one('libreria.autor', string='Autor')  
    # Un libro puede tener varios géneros y un género tiene muchos libros  
    genero_ids = fields.Many2many('libreria.genero', string='Géneros')
```

`autor.py`

```
# -*- coding: utf-8 -*-  
from odoo import models, fields, api  
  
class Autor(models.Model):  
    _name = 'libreria.autor'  
    _description = 'Autor'  
  
    name = fields.Char(string='Nombre', required=True)  
    #un autor tiene muchos libros y un libro SOLO puede pertenecer a un autor  
    libro_ids = fields.One2many('libreria.libro', 'autor_id', string='Libros')  
  
#campo computados
```

```
numLibros = fields.Integer(string='Número de Libros',
compute='_compute_contar_libros', store=False)
#función para campo computado que cuenta el número de libros del autor
@api.depends('libro_ids')
def _compute_contar_libros(self):
    for autor in self:
        autor.numLibros = len(autor.libro_ids)
```

Aquí vamos a detenernos y fijarnos en esta relación

```
#un autor tiene muchos libros y un libro SÓLO puede pertenecer a un autor
libro_ids = fields.One2many('libreria.libro', 'autor_id', string='Libros')
```

Estamos definiendo los ids de los libros del autor y la relación One2many mediante la variable `autor_id` haremos referencia al modelo libro donde **teníamos esa misma variable** `autor_id`

También hay que prestar atención a un **nuevo tipo de campo llamados computados**. `numLibros` va a almacenar el número total de libros que tiene ese autor.

Explicación del código:

- `numLibros`: Es un campo del modelo que almacena un número entero para almacenar el total de libros del autor.
- `compute='_compute_contar_libros'`: Esto indica el nombre de la función para el cálculo computado del número de libros. Un campo computado es una variable que calcula automáticamente un valor mediante la llamada a una función llamada `_compute_contar_libros`.
- `store=False`: Especifica que el valor **no se guardará en la base de datos**, sino que se calculará **cada vez que se acceda al campo**. Si lo estableciéramos a `True` se guardaría en base de datos. En este caso no nos hace falta. Ponerlo a `False` es útil si no necesitamos persistir el valor y preferimos optimizar espacio en la base de datos.
- `@api.depends`: Es un **decorador** de Odoo que informa al sistema qué campos (o relaciones) deben ser observados para saber **cuándo recalcular el valor** del campo computado. En este caso la función `_compute_contar_libros` se ejecutará **automáticamente** cada vez que el campo `libro_ids` cambie (es decir, cuando se añada o elimine un libro relacionado con el autor).
- `for autor in self`: El bucle recorre cada registro del modelo (`self`) que necesita que se calcule el campo `numLibros`.
- `len(autor.libro_ids)`:

- `libro_ids` es un campo relacional **One2many** (relación con el modelo `libreria.libro`).
- `len()` cuenta cuántos registros (libros) hay en esa relación.
- `autor.numLibros = ...`: Asigna el resultado de la función (`len(libro_ids)`) al campo `numLibros` de cada registro del modelo.

Por último habrá que añadir el campo `numLibros` a la vista `list`, por ejemplo para que muestre el total de libros del autor.

genero.py

```
# -*- coding: utf-8 -*-
from odoo import models, fields, api

class Genero(models.Model):
    _name = 'libreria.genero'
    _description = 'Género Literario'

    name = fields.Char(string='Nombre', required=True)
    descripcion = fields.Text(string='Descripción')
    #un genero puede pertenecer a muchos libros y un libro puede tener varios géneros
    libro_ids = fields.Many2many('libreria.libro', string='Libros')
```

5. FICHERO DE ACCESOS

```
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_libreria_libro,libreria.libro,model_libreria_libro,base.group_user,1,1,1,1
access_libreria_autor,libreria.autor,model_libreria_autor,base.group_user,1,1,1,1
access_libreria_genero,libreria.genero,model_libreria_genero,base.group_user,1,1,1,1
```

Añadimos todos los modelos al fichero `ir.model.access.csv`

6. VISTAS Y ACCIONES

Para añadir las vistas debemos fijarnos en el archivo `manifest` donde indicábamos los nombres de los archivos `xml` para las vistas.

```
'data': [
    'security/ir.model.access.csv',
    'views/libro.xml',
    'views/autor.xml',
    'views/genero.xml',
    'views/menus.xml',
    #'views/templates.xml',
```

libro.xml

```
<odoo>
<data>
```

```
<!-- explicit list view definition -->
<record model="ir.ui.view" id="libreria.libro_list">
  <field name="name">Listado de Libros</field>
  <field name="model">libreria.libro</field>
  <field name="arch" type="xml">
    <list>
      <field name="portada" widget="image" options="{ 'size': [50, 150] }"
/>
      <field name="name" />
      <field name="isbn" />
      <field name="fecha_publicacion" />
    </list>
  </field>
</record>

<record model="ir.ui.view" id="libreria.esta_mi_vista_tipo_libro_form">
  <field name="name">Formulario Libro</field>
  <field name="model">libreria.libro</field>
  <field name="arch" type="xml">
    <form string="Formulario Libro">
      <sheet>
        <group>
          <field name="portada" widget="image" options="{ 'size': [150,
250] }" />
          <group>
            <field name="name" />
            <field name="autor_id" />
          </group>
          <group>
            <field name="isbn" />
            <field name="fecha_publicacion" />
          </group>
        </group>
        <!-- aquí aparecen las pestañas -->
        <notebook>
          <page string="Géneros del libro">
            <field name="genero_ids" />
          </page>
          <page string="Descripción">
            <field name="descripcion" />
          </page>
        </notebook>
      </sheet>
    </form>
  </field>
</record>

<!-- actions opening views on models -->
<record model="ir.actions.act_window" id="libreria.libro_action_window">
  <field name="name">Listado de Libros</field>
  <field name="res_model">libreria.libro</field>
```

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

</data>
</odoo>
```

autor.xml

```
<odoo>
  <data>

    <!-- explicit list view definition -->
    <record model="ir.ui.view" id="libreria.autor_list">
      <field name="name">Listado de Autores</field>
      <field name="model">libreria.autor</field>
      <field name="arch" type="xml">
        <list>
          <field name="name" />
          <field name="numLibros" />
        </list>
      </field>
    </record>

    <!-- Vista de formulario para autores -->
    <record id="view_autor_form" model="ir.ui.view">
      <field name="name">Formulario Autor</field>
      <field name="model">libreria.autor</field>
      <field name="arch" type="xml">
        <form>
          <sheet>
            <group>
              <field name="name" />
              <field name="numLibros" readonly="1"/>
            </group>
            <notebook>
              <page string="Libros del autor">
                <field name="libro_ids" />
              </page>
            </notebook>
            <!-- añadir las pestañas -->
          </sheet>
        </form>
      </field>
    </record>

    <!-- actions opening views on models -->
    <record model="ir.actions.act_window" id="libreria.autor_action_window">
      <field name="name">Listado de Autores</field>
      <field name="res_model">libreria.autor</field>
```



```
<field name="view_mode">list,form</field>
</record>
</data>
</odoo>
```

genero.xml

```
<odoo>
  <data>
    <!-- explicit list view definition -->
    <record id="libreria.genero_list" model="ir.ui.view">
      <field name="name">Listado de Géneros</field>
      <field name="model">libreria.genero</field>
      <field name="arch" type="xml">
        <list string="Géneros">
          <field name="name" />
        </list>
      </field>
    </record>
    <!-- Vista de formulario para genero -->
    <record id="view_genero_form" model="ir.ui.view">
      <field name="name">Formulario genero</field>
      <field name="model">libreria.genero</field>
      <field name="arch" type="xml">
        <form>
          <sheet>
            <group>
              <field name="name" />
              <field name="descripcion" />
            </group>
            <notebook>
              <page string="Libros">
                <field name="libro_ids" />
              </page>
            </notebook>
          </sheet>
        </form>
      </field>
    </record>
    <!-- actions opening views on models -->
    <record model="ir.actions.act_window"
id="libreria.genero_action_window">
      <field name="name">Géneros</field>
      <field name="res_model">libreria.genero</field>
      <field name="view_mode">list,form</field>
    </record>
  </data>
</odoo>
```

menus.xml

```
<odoo>
  <data>
    <!-- Top menu item -->
    <menuitem name="Librería Bosco" id="libreria.menu_root" />
    <!-- menu categories -->
    <menuitem name="Gestión" id="libreria.menu_1" parent="libreria.menu_root"
  />
    <!-- actions -->
    <menuitem name="Libros" id="libreria.menu_1_list" parent="libreria.menu_1"
      action="libreria.libro_action_window" />
    <menuitem name="Autores" id="libreria.menu_2_list"
parent="libreria.menu_1"
      action="libreria.autor_action_window" />
    <menuitem name="Géneros" id="libreria.menu_3_list"
parent="libreria.menu_1"
      action="libreria.genero_action_window" />
  </data>
</odoo>
```

7. CARGAR DATOS DEMO

En el directorio demo hay un archivo con el nombre demo.xml que nos servirá para cargar datos de ejemplo.

Vamos a cargar unos datos de ejemplo:

```
<odoo>
  <data>

    <record id="genero1" model="libreria.genero">
      <field name="name">Novela de caballerías</field>
    </record>
    <record id="genero2" model="libreria.genero">
      <field name="name">Novela picaresca</field>
    </record>
    <record id="genero3" model="libreria.genero">
      <field name="name">Sátira</field>
    </record>
    <record id="genero4" model="libreria.genero">
      <field name="name">Ficción moderna</field>
    </record>
    <record id="genero5" model="libreria.genero">
      <field name="name">Comedia</field>
    </record>
    <record id="genero6" model="libreria.genero">
      <field name="name">Tragedia</field>
    </record>
    <record id="genero7" model="libreria.genero">
```

```
<field name="name">Histórica</field>
</record>
<record id="genero8" model="libreria.genero">
  <field name="name">Romance</field>
</record>
<record id="genero9" model="libreria.genero">
  <field name="name">Fantasía</field>
</record>
<record id="genero10" model="libreria.genero">
  <field name="name">Ciencia ficción</field>
</record>
<record id="genero11" model="libreria.genero">
  <field name="name">Misterio</field>
</record>
<record id="genero12" model="libreria.genero">
  <field name="name">Terror</field>
</record>

<record id="autor1" model="libreria.autor">
  <field name="name">Miguel de Cervantes</field>
</record>
<record id="autor2" model="libreria.autor">
  <field name="name">William Shakespeare</field>
</record>
<record id="autor3" model="libreria.autor">
  <field name="name">J.R.R. Tolkien</field>
</record>
<record id="autor4" model="libreria.autor">
  <field name="name">Gabriel García Márquez</field>
</record>
<record id="autor5" model="libreria.autor">
  <field name="name">Jane Austen</field>
</record>
<record id="autor6" model="libreria.autor">
  <field name="name">George Orwell</field>
</record>
<record id="autor7" model="libreria.autor">
  <field name="name">Mark Twain</field>
</record>
<record id="autor8" model="libreria.autor">
  <field name="name">Leo Tolstoy</field>
</record>
<record id="autor9" model="libreria.autor">
  <field name="name">Edgar Allan Poe</field>
</record>
<record id="autor10" model="libreria.autor">
  <field name="name">Fyodor Dostoevsky</field>
</record>
<record id="autor11" model="libreria.autor">
  <field name="name">Homer</field>
</record>
```

```
</record>
<record id="autor12" model="libreria.autor">
  <field name="name">Virginia Woolf</field>
</record>

<record id="libro1" model="libreria.libro">
  <field name="name">El Señor de los Anillos 1. La Comunidad del
Anillo</field>
  <field name="isbn">123456789-A</field>
  <field name="descripcion">Ambientada en la Tierra Media, cuenta la
historia del Señor Oscuro Sauron, que está buscando el Anillo Único, el cual
ha acabado en poder del hobbit Frodo Bolsón (Elijah Wood). El destino de la
Tierra Media está en juego mientras Frodo y ocho compañeros que forman la
Compañía del Anillo comienzan un largo y peligroso viaje hacia el Monte del
Destino en la tierra de Mordor, que es el único lugar en el que el anillo
puede ser destruido.</field>
</record>

<record id="libro2" model="libreria.libro">
  <field name="name">El Señor de los Anillos nº 02/03 Las Dos
Torres</field>
  <field name="isbn">123456789-B</field>
  <field name="descripcion">Empieza tu viaje a la Tierra Media. Los
libros que han inspirado la nueva serie EL SEÑOR DE LOS ANILLOS: LOS ANILLOS
DEL PODER. La segunda entrega de la trilogía de J. R. R. Tolkien El Señor de
los Anillos. La Compañía se ha disuelto y sus integrantes emprenden caminos
separados. Frodo y Sam avanzan solos en su viaje a lo largo del río Anduin,
perseguidos por la sombra misteriosa de un ser extraño que también ambiciona
la posesión del Anillo. Mientras, hombres, elfos y enanos se preparan para la
batalla final contra las fuerzas del Señor del Mal</field>
</record>

<record id="libro3" model="libreria.libro">
  <field name="name">La historia Interminable</field>
  <field name="isbn">9876854-CD</field>
  <field name="descripcion">Se trata de una novela que a pesar de
haber sido calificada como «un nuevo clásico de la literatura juvenil»,3
siempre fue defendida por su autor como una novela que se extiende más allá de
la mera narración para convertirse en una crítica. Por ejemplo, Ende defendió
que la novela expresa el deseo de encontrar la realidad que nos rodea a través
de recorrer el camino inverso, es decir, la parte interna de cada uno que
reside en su imaginación</field>
</record>
</data>
</odoo>
```

8. ACTUALIZAR LOS CAMBIOS DEL MÓDULO

Cada vez que modifiquemos el módulo podremos actualizar los cambios desde 'Actualizar lista de Aplicaciones' y en nuestro módulo pulsaremos sobre 'Actualizar' pero hay 2 maneras de actualizar más rápidamente:

1. Lanzar los cambios desde la terminal del sistema operativo.

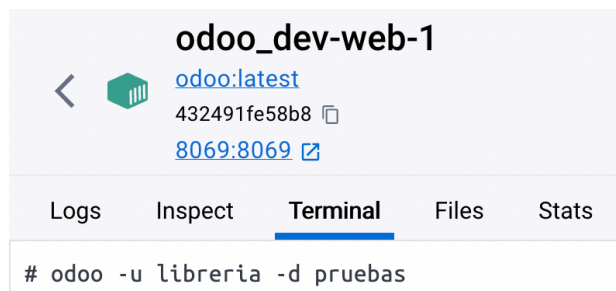
```
docker exec -it odoo_dev-web-1 odoo -u libreria -d pruebas --dev=reload
```

Esto lanzará en nuestro contenedor `odoo_dev-web-1` la actualización del módulo `libreria` en nuestra base de datos `-d pruebas` aplicándose los cambios y limpiando la caché con `--dev=reload`.

2. Lanzar los cambios desde la terminal del contenedor.

Puedes usar este comando que hará exactamente lo mismo pero desde dentro de la terminal del propio contenedor de Odoo.

```
odoo -u libreria -d pruebas --dev=reload
```



CONSEJO: Si después de instalar, actualizar los módulos varias veces aparecen más opciones en el menú con el mismo nombre, tablas repetidas, datos que se mantienen puedes borrarlo todo desde el menú Ajustes -> Técnico.

9. TABLAS CREADAS.

Encuentra las tablas que se han creado a partir de los modelos creados. Presta especial atención a la tabla many2many entre libro y género.

Puedes usar esta consulta:

```
SELECT * FROM INFORMATION_SCHEMA.tables  
WHERE TABLE_NAME LIKE '%libreria%'
```

10. AMPLIACIÓN

EJERCICIO. Crea un campo computado para calcular el tiempo que hace que se publicó ese libro. Debe indicarlo en años, meses y días desde la fecha de publicación a la fecha actual.

Para un correcto cálculo desde la fecha de publicación tendremos que validar si la fecha de publicación no está vacía.