

Práctica 4 - Desarrollo de módulos de Odoo: Modelo y Vista

Sistemas de Gestión Empresarial

Cristian Fernández

15 de diciembre de 2025

Índice

| | |
|--|----|
| 1. Introducción | 2 |
| 2. Actividad 01 - Lista de tareas | 3 |
| 3. Actividad 02 - Biblioteca de cómics | 10 |
| 4. Actividad 03 - Hospital | 22 |
| 5. Actividad 04 - Instituto | 31 |

1. Introducción

El objetivo de esta práctica es aplicar los conocimientos adquiridos en ejercicios anteriores sobre modelos, relaciones y vistas en el desarrollo de aplicaciones. Para ello, nos ocuparemos de la implementación de cuatro actividades que abarcan diferentes contextos, como la gestión de tareas, bibliotecas de cómics, pacientes y médicos, y ciclos formativos.

Habrà que tener en cuenta la revisión de los ejemplos proporcionados en el siguiente repositorio. <https://github.com/sergarb1/OdooModulosEjemplos>

Una vez que verifiquemos y probemos el funcionamiento de cada ejemplo en nuestro contenedor de Odoo debemos:

- Revisar los manifests.
- Revisar los modelos de datos.
- Revisar las vistas.

Entender y probar estos ejemplos nos facilitará la comprensión sobre la arquitectura de Odoo y su sistema de modular y, por tanto, la realización la práctica.

Se reutilizará el repositorio creado en las prácticas anteriores, con objeto de tener todo el trabajo bien organizado y disponible para una posible visualización en cualquier momento.

Cada actividad tendrá una explicación breve del proceso de creación y se demostrará su funcionamiento mediante capturas. Si fuese necesario también se comentarán los problemas encontrados y las soluciones empleadas.

2. Actividad 01 - Lista de tareas

En esta primera actividad modificaremos el módulo de la práctica anterior **‘lista_tareas’** para que:

- Tenga una nueva vista para ver las tareas en formato **Kanban**.
- Modificar las tareas para que tengan una fecha asignada (implementado en la práctica anterior) y crear una nueva vista para visualizarlas en un formato **Calendario** según esa fecha.

Las primera tanda de capturas servirán para mostrar el código creado/modificado e intentarán ser lo suficientemente autoexplicativas para una correcta comprensión del mismo.

Echémosle un vistazo a la vista principal del módulo.

```

views.xml M X
addons > lista_tareas > views > views.xml
1 <odoo>
2   <data>
3     <!-- DEFINICIÓN DE VISTAS -->
4
5     <!-- Listado de tareas principal -->
6     <record model="ir.ui.view" id="lista_tareas.list">
7       <field name="name">lista_tareas list</field>
8       <field name="model">lista_tareas.lista_tareas</field>
9       <field name="arch" type="xml">
10         <list>
11           <field name="tipo"/>
12           <field name="fecha_limite"/>
13           <field name="pendiente"/>
14           <field name="realizada"/>
15           <field name="fallida"/>
16         </list>
17       </field>
18     </record>
19
20     <!-- Listado de tareas formato Kanban -->
21     <record id="lista_tareas.view_kanban" model="ir.ui.view">
22       <field name="name">lista_tareas kanban</field>
23       <field name="model">lista_tareas.lista_tareas</field>
24       <field name="arch" type="xml">
25         <kanban class="o_kanban_mobile" sample="1" quick_create="false">
26           <progressbar field="activity_state"
27             colors='{ "planned": "success", "today": "warning", "overdue": "danger" }' />
28           <templates>
29             <t t-name="card">
30               <div class="d-flex mb-2" style="justify-content: space-between;">
31                 <field name="tipo" class="fw-bolder fs-5" />
32                 <field name="fecha_limite" class="ms-1 text-muted fs-5"/>
33               </div>
34               <footer>
35                 <div class="d-flex text-muted">
36                   <field name="description"/>
37                 </div>
38               </footer>
39             </t>
40           </templates>
41         </kanban>
42       </field>
43     </record>

```

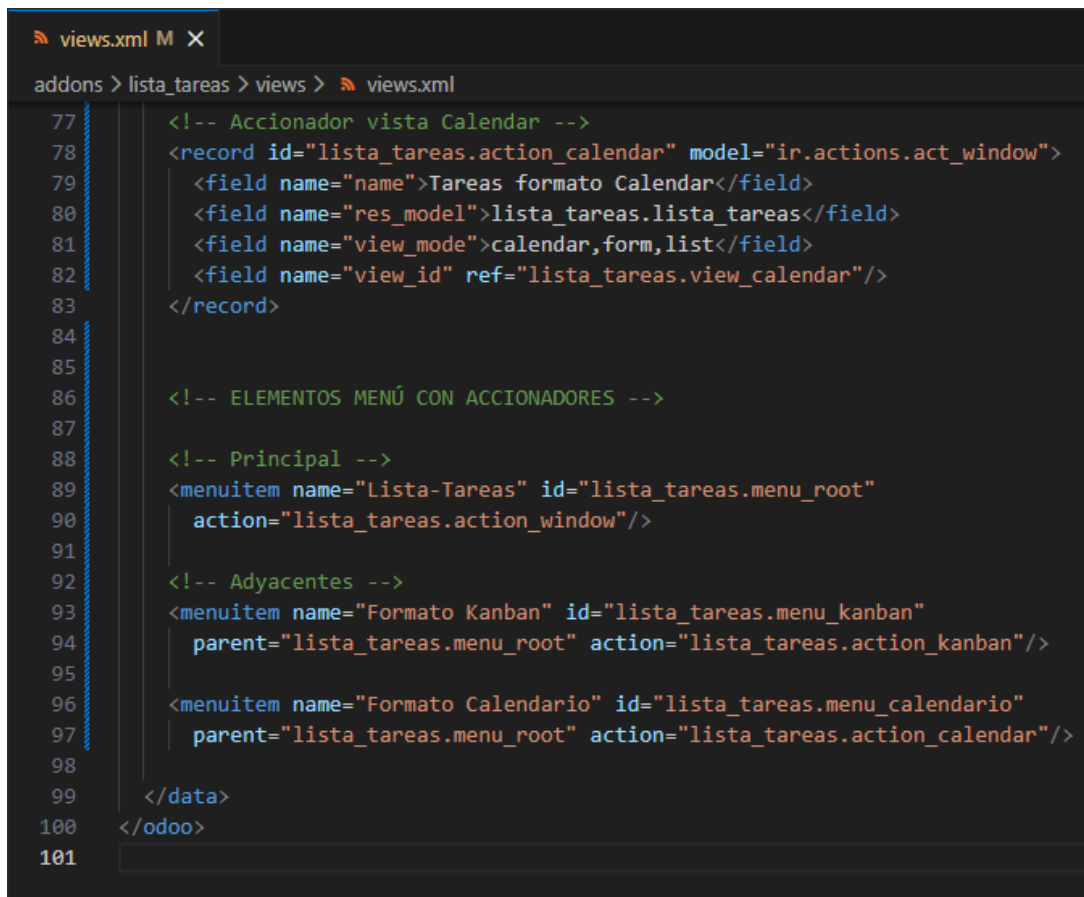
Figura 1: Vista listado principal y la nueva adición, la vista Kanban

```

views.xml M X
addons > lista_tareas > views > views.xml
45      <!-- Listado de tareas formato Calendar -->
46      <record id="lista_tareas.view_calendar" model="ir.ui.view">
47          <field name="name">lista_tareas calendar</field>
48          <field name="model">lista_tareas.lista_tareas</field>
49          <field name="arch" type="xml">
50              <calendar string="Tareas" date_start="fecha_limite" color="tipo">
51                  <field name="tipo"/>
52                  <field name="description"/>
53                  <field name="pendiente"/>
54                  <field name="realizada"/>
55              </calendar>
56          </field>
57      </record>
58
59
60      <!-- ACCIONES DE VENTANA -->
61
62      <!-- Accionador vista Lista Tareas -->
63      <record id="lista_tareas.action_window" model="ir.actions.act_window">
64          <field name="name">Listado principal</field>
65          <field name="res_model">lista_tareas.lista_tareas</field>
66          <field name="view_mode">list,form</field>
67      </record>
68
69      <!-- Accionador vista Kanban -->
70      <record id="lista_tareas.action_kanban" model="ir.actions.act_window">
71          <field name="name">Tareas formato Kanban</field>
72          <field name="res_model">lista_tareas.lista_tareas</field>
73          <field name="view_mode">kanban,form,list</field>
74          <field name="view_id" ref="lista_tareas.view_kanban"/>
75      </record>

```

Figura 2: Segunda adición (vista Calendar) y accionadores listado principal y Kanban



```
77 <!-- Accionador vista Calendar -->
78 <record id="lista_tareas.action_calendar" model="ir.actions.act_window">
79   <field name="name">Tareas formato Calendar</field>
80   <field name="res_model">lista_tareas.lista_tareas</field>
81   <field name="view_mode">calendar,form,list</field>
82   <field name="view_id" ref="lista_tareas.view_calendar"/>
83 </record>
84
85
86 <!-- ELEMENTOS MENÚ CON ACCIONADORES -->
87
88 <!-- Principal -->
89 <menuitem name="Lista-Tareas" id="lista_tareas.menu_root"
90   action="lista_tareas.action_window"/>
91
92 <!-- Adyacentes -->
93 <menuitem name="Formato Kanban" id="lista_tareas.menu_kanban"
94   parent="lista_tareas.menu_root" action="lista_tareas.action_kanban"/>
95
96 <menuitem name="Formato Calendario" id="lista_tareas.menu_calendario"
97   parent="lista_tareas.menu_root" action="lista_tareas.action_calendar"/>
98
99 </data>
100 </odoo>
101
```

Figura 3: Accionador Calendar y elementos de menú

Y este es todo el código generado para esta actividad. Ambas vistas fueron añadidas en el archivo xml principal *views.xml* por pura simplificación de código.

A continuación unas capturas del módulo *lista_tareas* en funcionamiento.

| <div> <div> <div></div> <div>Lista-Tareas</div> </div> <div> <div>Formato Kanban</div> <div>Formato Calendario</div> </div> </div> <div> <div></div> <div>A</div> </div> | | | | |
|--|---------------------|-------------------------------------|--------------------------|--------------------------|
| <div> <div>New</div> <div>Listado principal</div> <div></div> </div> <div>1-6 / 6</div> <div> <div><</div> <div>></div> </div> | | | | |
| <div> <div>Q</div> <div>Search...</div> <div></div> </div> | | | | |
| <input type="checkbox"/> Tipo | Fecha límite | Pendiente | Realizada | Fallida |
| <input type="checkbox"/> Cita médica | 12/18/2025 17:00:00 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> Hacer compra | 12/16/2025 11:00:00 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> Sacar al perro | 12/15/2025 21:00:00 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> Ir a la peluquería | 12/17/2025 13:00:00 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> Ir a la gestoría | 12/19/2025 16:00:00 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |
| <input type="checkbox"/> Compras navideñas | 12/20/2025 18:00:00 | <input checked="" type="checkbox"/> | <input type="checkbox"/> | <input type="checkbox"/> |

Figura 4: Vista principal

| <div> <div> <div></div> <div>Lista-Tareas</div> </div> <div> <div>Formato Kanban</div> <div>Formato Calendario</div> </div> </div> <div> <div></div> <div>A</div> </div> | | | | |
|---|--|--|--|--|
| <div> <div>New</div> <div>Tareas formato Kanban</div> <div></div> </div> <div>1-6 / 6</div> <div> <div><</div> <div>></div> <div></div> <div></div> </div> | | | | |
| <div> <div>Q</div> <div>Search...</div> <div></div> </div> | | | | |
| <div> <div> <div>Cita médica</div> <div>12/18/2025 17:00:00</div> <div>Vacuna contra la gripe</div> </div> <div> <div>Hacer compra</div> <div>12/16/2025 11:00:00</div> <div>Leche, patatas, huevos.</div> </div> </div> | | | | |
| <div> <div> <div>Sacar al perro</div> <div>12/15/2025 21:00:00</div> <div>El cánido debe evacuar satisfactoriamente antes de finalizar el día.</div> </div> <div> <div>Ir a la peluquería</div> <div>12/17/2025 13:00:00</div> <div>Hay que arreglar este desastre.</div> </div> </div> | | | | |
| <div> <div> <div>Ir a la gestoría</div> <div>12/19/2025 16:00:00</div> <div>Gestionar asuntos.</div> </div> <div> <div>Compras navideñas</div> <div>12/20/2025 18:00:00</div> <div>Urgente, no olvidarse!</div> </div> </div> | | | | |

Figura 5: Vista Kanban

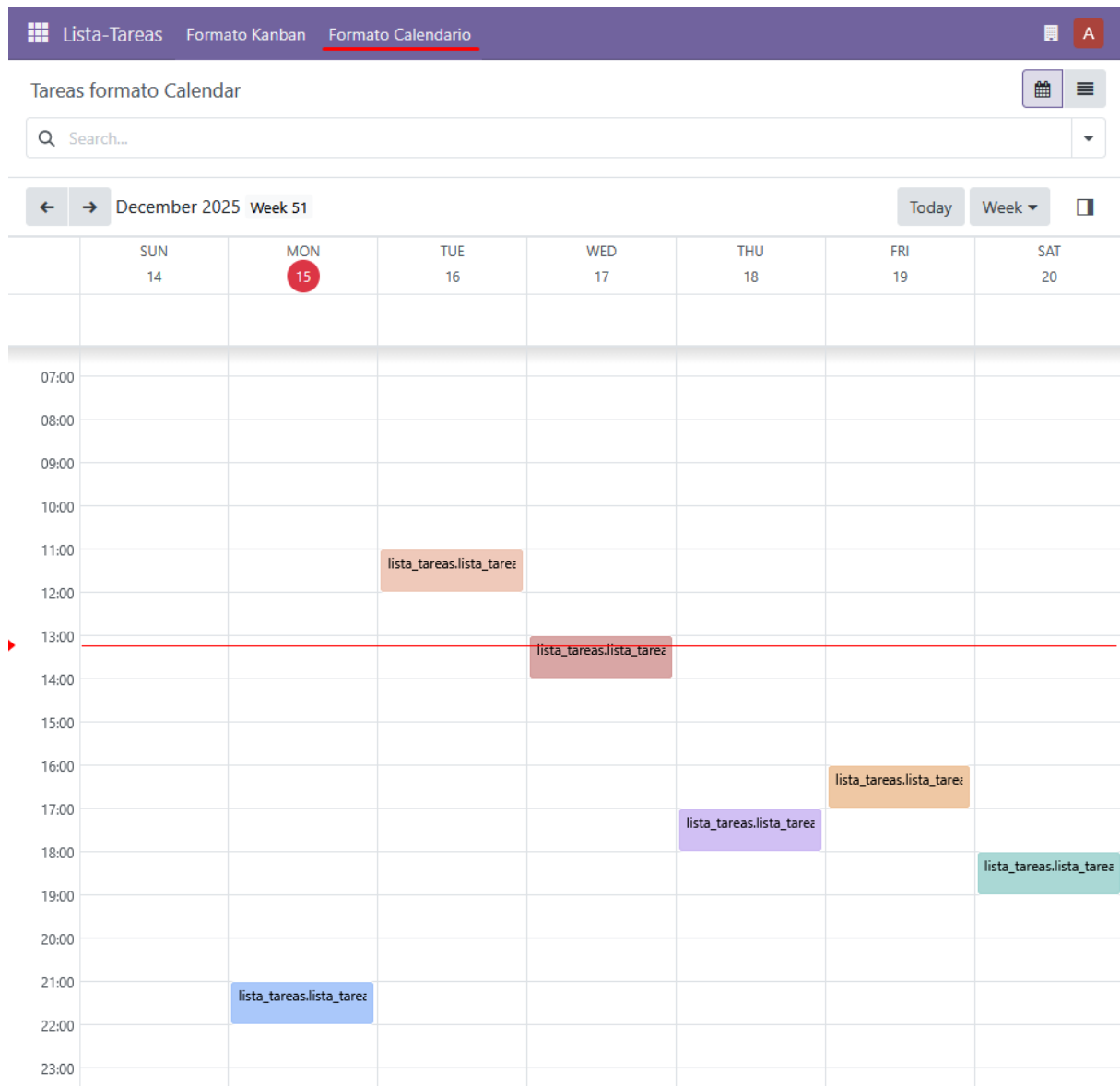


Figura 6: Vista Calendar parte 1

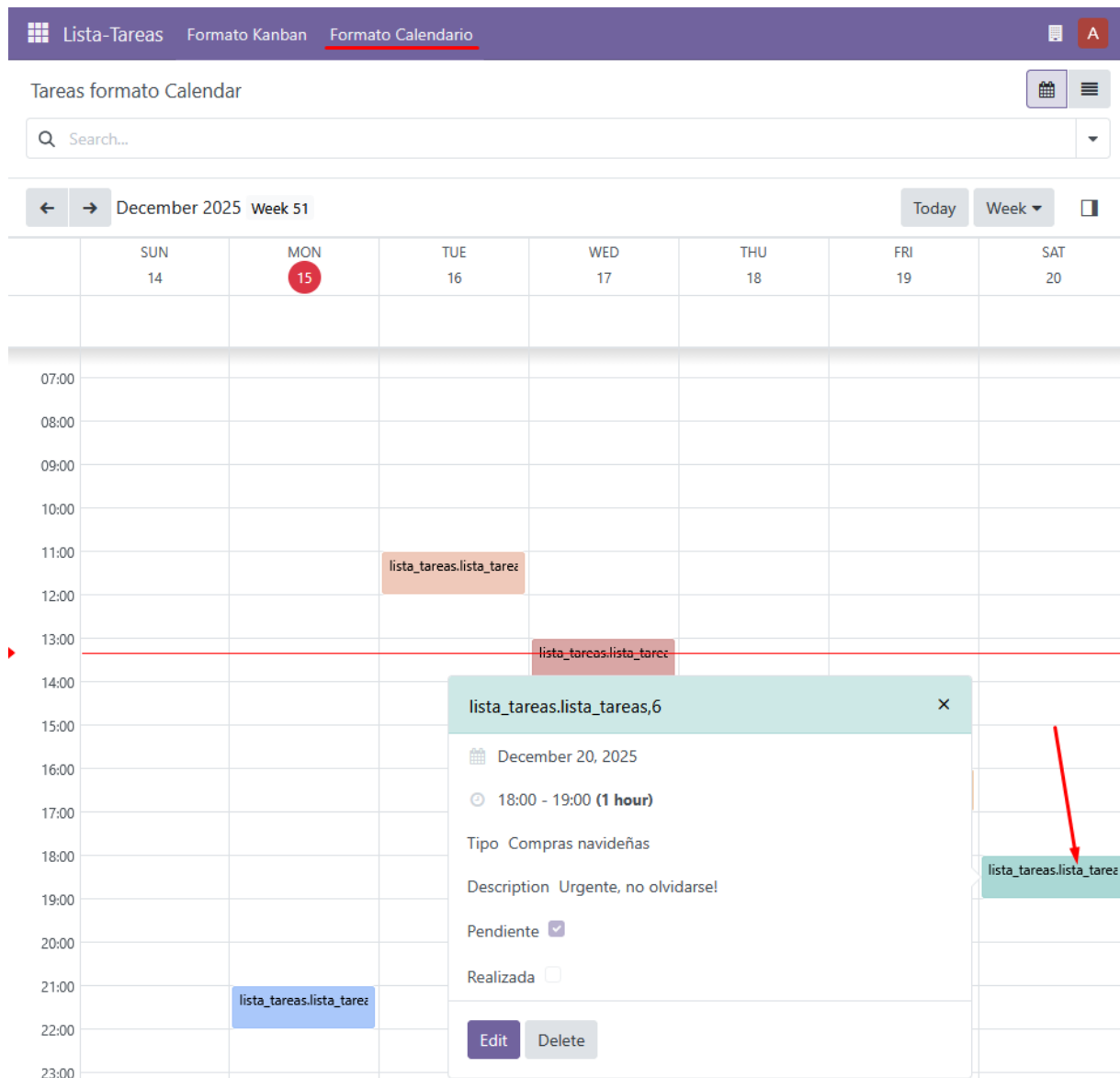


Figura 7: Vista Calendar parte 2 (clic en tarea)

3. Actividad 02 - Biblioteca de cómics

En esta segunda actividad ampliaremos el módulo de prueba ‘**EJ03-ComicsSimple**’ con objeto de:

- Incluir la posibilidad de gestionar socios, almacenando su nombre, apellido e identificador.
- Introducir al opción de que existan ejemplares de cómics para préstamo.
 - El modelo actual de cómic servirá como referencia de la información del cómic.
 - Se creará un nuevo modelo para gestionar los ejemplares prestables de cada cómic.
- Estos ejemplares de préstamo deben registrar:
 - A quién están prestados (socio).
 - La fecha de inicio y fecha de fin del préstamo.
 - No es necesario mantener un histórico de préstamos, sólo quién tiene el cómic actualmente, cuándo se prestó y la fecha prevista de devolución.
- Restricciones:
 - La fecha de préstamo no puede ser posterior al día actual.
 - La fecha prevista de devolución no puede ser anterior al día actual.

A continuación se mostrarán las capturas del código realizado y las capturas con el módulo en funcionamiento.

```
biblioteca_comic.py 2, M X socio.py 1, U ejemplar_prestamo.py 1, U
addons > EJ03-ComicsSimple > models > biblioteca_comic.py > ...
64 #Fecha de publicación
65 fecha_publicacion = fields.Date('Fecha publicación')
66
67 #Precio del libro
68 precio = fields.Float('Precio')
69 #Numero de paginas.
70 paginas = fields.Integer('Numero de páginas',
71     #Hace que este atributo este disponible para este grupo de seguridad
72     #Que en este caso son todos los usuarios
73     groups='base.group_user',
74     #Establece que si el estado es perdido, el numero de paginas no se puede cambiar
75     estados={'perdido': [('readonly', True)]},
76     #Texto a mostrar en la ayuda de la interfaza al dejar el ratón encima
77     help='Total numero de paginas',
78     #Si se pone a true, indica que si este atributo se aplica a distintas empresas en Odoo
79     #para cada empresa ponga un valor distintos
80     #Esta colocado con fin didactico a false
81     company_dependent=False)
82
83 #Valoración lector, indicando como son los datos
84 valoracion_lector = fields.Float(
85     'Valoración media lectores',
86     digits=(14, 4), # Precision opcional (total, decimales),
87 )
88 # Relación muchos a muchos de autores utilizando un "partner"
89 # de Odoo (Es un elemento que puede ser empresa o individuo)
90 # https://stackoverflow.com/questions/22927605/what-is-res-partner
91 autor_ids = fields.Many2many('res.partner', string='Autores')
92
93
94 # RELACIÓN 1-N (Un cómic tiene muchos ejemplares)
95 ejemplar_ids = fields.One2many('ejemplar.prestamo', 'comic_id', string='Ejemplares Prestables')
96
97
98 #Constraints de SQL del modelo
99 #Util cuando la constraint se puede definir con sintaxis SQL
100 _sql_constraints = [
101     ('name_uniq', 'UNIQUE (nombre)', 'El titulo Comic debe ser único.'),
102     ('positive_page', 'CHECK(paginas>0)', 'El comic debe tener al menos una página')
103 ]
```

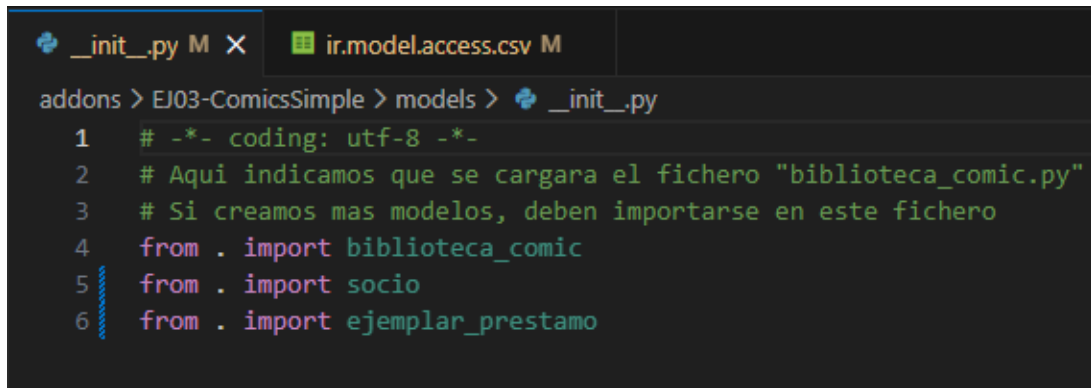
Figura 8: Única modificación del modelo principal

```
addons > EJ03-ComicsSimple > models > socio.py > ...
1  from odoo import models, fields, api
2
3  class Socio(models.Model):
4      # Nombre y descripción del modelo
5      _name = 'socio.biblioteca'
6      _description = 'Socio de biblioteca'
7
8      # Este atributo privado mostrará el nombre del socio en la relación con ejemplar.
9      _rec_name = 'nombre_completo'
10
11     # Propiedades
12     nombre = fields.Char('Nombre de pila', required=True)
13     apellidos = fields.Char('Apellidos', required=True)
14     id_socio = fields.Char('DNI', required=True, index=True)
15     # Para mostrar el nombre completo de socio en la relación usaremos un campo calculado
16     nombre_completo = fields.Char('Nombre Completo', compute='_compute_nombre_completo', store=True, index=True)
17
18     # RELACIÓN 1-N (Un socio puede tener muchos ejemplares prestados)
19     ejemplar_ids = fields.One2many('ejemplar.prestamo', 'socio_id', string='Ejemplares Prestados')
20
21
22     # Campo calculado para concatenar nombre y apellido
23     @api.depends('nombre', 'apellidos')
24     def _compute_nombre_completo(self):
25         for record in self:
26             record.nombre_completo = f"{record.nombre} {record.apellidos}"
27
```

Figura 9: Nuevo modelo *socio.py*

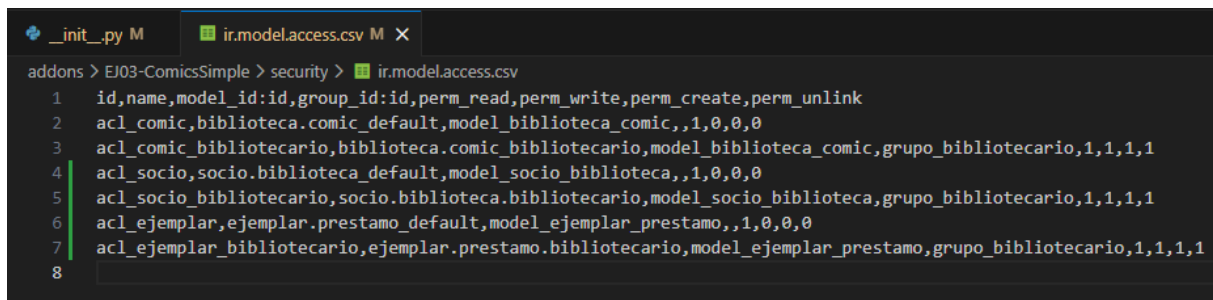
```
addons > EJ03-ComicsSimple > models > ejemplar_prestamo.py > ...
1  from odoo import models, fields, api
2
3  class EjemplarPrestamo(models.Model):
4      # Nombre y descripción del modelo
5      _name = 'ejemplar.prestamo'
6      _description = 'Ejemplar de cómic prestable'
7
8      # Relación N:1 (muchos ejemplares para un cómic)
9      comic_id = fields.Many2one('biblioteca.comic', string='Comic de Referencia', required=True, ondelete='cascade')
10     # Relación N:1 (muchos ejemplares para un socio)
11     socio_id = fields.Many2one('socio.biblioteca', string='Prestado a Socio')
12
13     # Fechas de entrega y devolución
14     fecha_inicio = fields.Date('Fecha de entrega')
15     fecha_fin = fields.Date('Fecha de devolución')
16
17
18     # Comprobación fecha de inicio
19     @api.constrains('fecha_inicio')
20     def _check_fecha_inicio(self):
21         for record in self:
22             if record.fecha_inicio and record.fecha_inicio > fields.Date.today():
23                 raise models.ValidationError('La fecha de entrega no puede ser posterior al día actual.')
24
25     #Comprobación fecha de devolución
26     @api.constrains('fecha_fin')
27     def _check_fecha_fin(self):
28         for record in self:
29             if record.fecha_fin and record.fecha_fin < fields.Date.today():
30                 raise models.ValidationError('La fecha de devolución no puede ser anterior al día actual.')
31
```

Figura 10: Nuevo modelo *ejemplar_prestamo.py*



```
__init__.py M X ir.model.access.csv M X
addons > EJ03-ComicsSimple > models > __init__.py
1  # -*- coding: utf-8 -*-
2  # Aquí indicamos que se cargara el fichero "biblioteca_comic.py"
3  # Si creamos mas modelos, deben importarse en este fichero
4  from . import biblioteca_comic
5  from . import socio
6  from . import ejemplar_prestamo
```

Figura 11: Se debe actualizar el `__init__.py` con los nuevos modelos



```
__init__.py M ir.model.access.csv M X
addons > EJ03-ComicsSimple > security > ir.model.access.csv
1  id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2  acl_comic,biblioteca.comic_default,model_biblioteca_comic,,1,0,0,0
3  acl_comic_bibliotecario,biblioteca.comic_bibliotecario,model_biblioteca_comic,grupo_bibliotecario,1,1,1,1
4  acl_socio,socio.biblioteca_default,model_socio_biblioteca,,1,0,0,0
5  acl_socio_bibliotecario,socio.biblioteca.bibliotecario,model_socio_biblioteca,grupo_bibliotecario,1,1,1,1
6  acl_ejemplar,ejemplar.prestamo_default,model_ejemplar_prestamo,,1,0,0,0
7  acl_ejemplar_bibliotecario,ejemplar.prestamo.bibliotecario,model_ejemplar_prestamo,grupo_bibliotecario,1,1,1,1
8
```

Figura 12: Se deben actualizar las reglas de acceso para los modelos nuevos, sino no se podrán visualizar ni utilizar

```
biblioteca_comic.xml M X
addons > EJ03-ComicsSimple > views > biblioteca_comic.xml

60 <!-- Definición de la vista busqueda-->
61 <record id="biblioteca_comic_view_search" model="ir.ui.view">
62   <field name="name">Búsqueda de Comics en la biblioteca</field>
63   <field name="model">biblioteca.comic</field>
64   <field name="arch" type="xml">
65     <search>
66       <field name="nombre"/>
67       <field name="autor_ids"/>
68       <!-- Indicamos que para si filtramos por los del dominio "autor_ids=false"
69       se muestre el texto "Sin autor"-->
70       <filter string="Sin autor" name="sin_autor" domain="(['autor_ids','=',False)]"/>
71     </search>
72   </field>
73 </record>
74
75 <!-- Definición de la vista Socios -->
76 <record id="biblioteca_socios_view" model="ir.ui.view">
77   <field name="name">Socios de la biblioteca</field>
78   <field name="model">socio.biblioteca</field>
79   <field name="arch" type="xml">
80     <list>
81       <field name="nombre"/>
82       <field name="apellidos"/>
83       <field name="id_socio"/>
84     </list>
85   </field>
86 </record>
87
88 <!-- Definición de la vista Prestamos -->
89 <record id="biblioteca_prestamos_view" model="ir.ui.view">
90   <field name="name">Ejemplares prestables</field>
91   <field name="model">ejemplar.prestamo</field>
92   <field name="arch" type="xml">
93     <list>
94       <field name="comic_id"/>
95       <field name="socio_id"/>
96       <field name="fecha_inicio"/>
97       <field name="fecha_fin"/>
98     </list>
99   </field>
100 </record>
101
```

Figura 13: Código añadido a la vista principal parte 1


```
biblioteca_comic.xml M X
addons > EJ03-ComicsSimple > views > biblioteca_comic.xml
101
102 <!-- Definimos como mostramos la vista en el modelo -->
103 <record id='biblioteca_comic_action' model='ir.actions.act_window'>
104   <field name="name">Biblioteca de Comics</field>
105   <!-- Indicamos a que modelo aplica -->
106   <field name="res_model">biblioteca.comic</field>
107   <!-- Indicamos que los comics pueden verse en tree para el listado,
108   y en form para la creación/edición -->
109   <field name="view_mode">list,form</field>
110 </record>
111
112 <!-- Accionador vista Socios -->
113 <record id="biblioteca_socios_action" model="ir.actions.act_window">
114   <field name="name">Socios</field>
115   <field name="res_model">socio.biblioteca</field>
116   <field name="view_mode">list,form</field>
117   <field name="view_id" ref="biblioteca_socios_view"/>
118 </record>
119
120 <!-- Accionador vista Prestamos -->
121 <record id="biblioteca_prestamos_action" model="ir.actions.act_window">
122   <field name="name">Prestamos</field>
123   <field name="res_model">ejemplar.prestamo</field>
124   <field name="view_mode">list,form</field>
125   <field name="view_id" ref="biblioteca_prestamos_view"/>
126 </record>
127
128
129 <!-- Simple menu item, sin más utilidad que la didáctica -->
130 <menuitem name="Mi biblioteca (Simple)" id="biblioteca_base_menu" />
131 <menuitem name="Comics" id="biblioteca_comic_menu" parent="biblioteca_base_menu"
132   action="biblioteca_comic_action"/>
133 <menuitem name="Socios" id="biblioteca_socios_menu" parent="biblioteca_base_menu"
134   action="biblioteca_socios_action"/>
135 <menuitem name="Prestamos" id="biblioteca_prestamos_menu" parent="biblioteca_base_menu"
136   action="biblioteca_prestamos_action"/>
137 </odoo>
138
```

Figura 14: Código añadido a la vista principal parte 2

Y este es todo el código generado para esta actividad. Ambas vistas fueron añadidas en el archivo xml principal *biblioteca_comic.xml* por pura simplificación de código.

A continuación unas capturas del módulo *EJ03-ComicsSimple* en funcionamiento.

| Mi biblioteca (Simple) Comics Socios Prestamos | | | 1-8 / 8 | |
|--|-------------------|------------|---------|--|
| New Biblioteca de Comics | | | | |
| Search... | | | | |
| <input type="checkbox"/> Titulo | Fecha publicación | Estado | | |
| <input type="checkbox"/> BERSERK (ED.MAXIMUM) N°21 | 12/11/2025 | Disponible | | |
| <input type="checkbox"/> DAREDEVIL: DÍA GÉLIDO EN EL INFIERNO | 12/11/2025 | Disponible | | |
| <input type="checkbox"/> EL ASOMBROSO SPIDERMAN 03 | 12/11/2025 | Disponible | | |
| <input type="checkbox"/> EL BAILE DEL VAMPIRO: REDES | 12/11/2025 | Disponible | | |
| <input type="checkbox"/> LA NOBLEZA DE LAS FLORES 11 | 12/11/2025 | Disponible | | |
| <input type="checkbox"/> PRÍNCIPE VALIENTE: 1937-1943. EN TIEMPOS DEL REY ARTURO | 12/11/2025 | Disponible | | |
| <input type="checkbox"/> SUPERLÓPEZ 1988-1990 (Edición Coleccionista) | 12/11/2025 | Disponible | | |
| <input type="checkbox"/> SUPERMAN UNLIMITED 02 | 12/11/2025 | Disponible | | |

Figura 15: Vista cómics con ejemplos creados

| Mi biblioteca (Simple) Comics Socios Prestamos | | | 1-10 / 10 | |
|--|------------------|-----------|-----------|--|
| New Socios | | | | |
| Search... | | | | |
| <input type="checkbox"/> Nombre de pila | Apellidos | DNI | | |
| <input type="checkbox"/> Antonio | Pepito Pérez | 11223344A | | |
| <input type="checkbox"/> María | Gutiérrez García | 12345678Q | | |
| <input type="checkbox"/> Federico | Pérez Olmo | 55667788B | | |
| <input type="checkbox"/> Laura | Gil Tomás | 36123456F | | |
| <input type="checkbox"/> Vanesa | Fernández Vigo | 36998877N | | |
| <input type="checkbox"/> Evaristo | Gil Nieve | 38123451T | | |
| <input type="checkbox"/> Ana | López Muñoz | 34121345R | | |
| <input type="checkbox"/> Alejandro | Rojas Díaz | 45123456M | | |
| <input type="checkbox"/> Cristina | Sosa Romero | 57123456A | | |
| <input type="checkbox"/> Manuel | Molina Estévez | 36612233J | | |

Figura 16: Vista socios con ejemplos creados

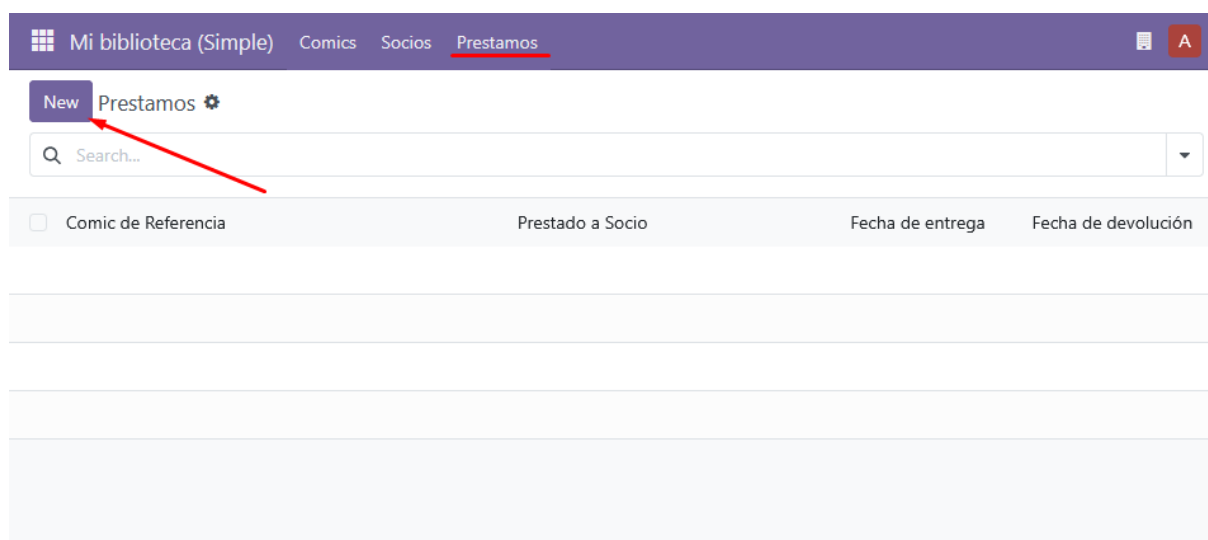


Figura 17: Vista prestamos vacía, creamos uno nuevo

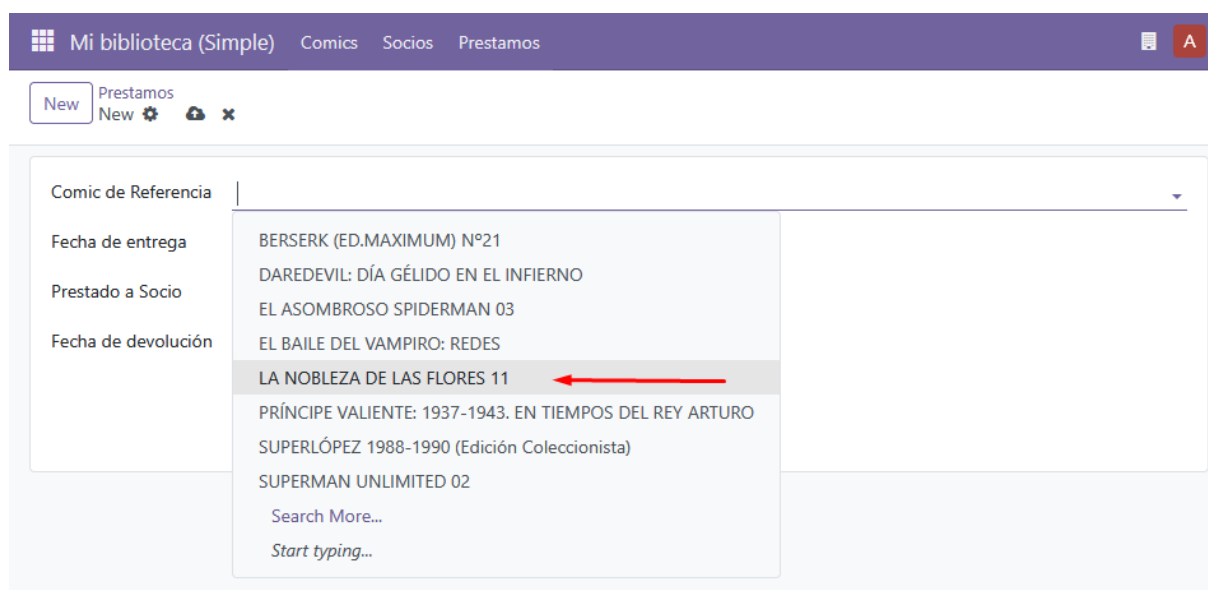


Figura 18: Seleccionamos uno de los ejemplares disponibles

Mi biblioteca (Simple) Comics Socios Prestamos

New Prestamos New [icon] [icon] [icon]

Comic de Referencia LA NOBLEZA DE LAS FLORES 11

Fecha de entrega

Prestado a Socio

Fecha de devolución

- Antonio Pepito Pérez
- María Gutiérrez García
- Federico Pérez Olmo
- Laura Gil Tomás**
- Vanesa Fernández Vigo
- Evaristo Gil Nieve
- Ana López Muñoz
- Alejandro Rojas Díaz
- Search More...
- Start typing...

Figura 19: Seleccionamos un socio

Mi biblioteca (Simple) Comics Socios Prestamos

New Prestamos New [icon] [icon] [icon]

Comic de Referencia LA NOBLEZA DE LAS FLORES 11

Fecha de entrega 12/16/2025

Prestado a Socio Laura Gil Tomás

Fecha de devolución 12/14/2025

Figura 20: Establecemos fechas erróneas para comprobar la validación

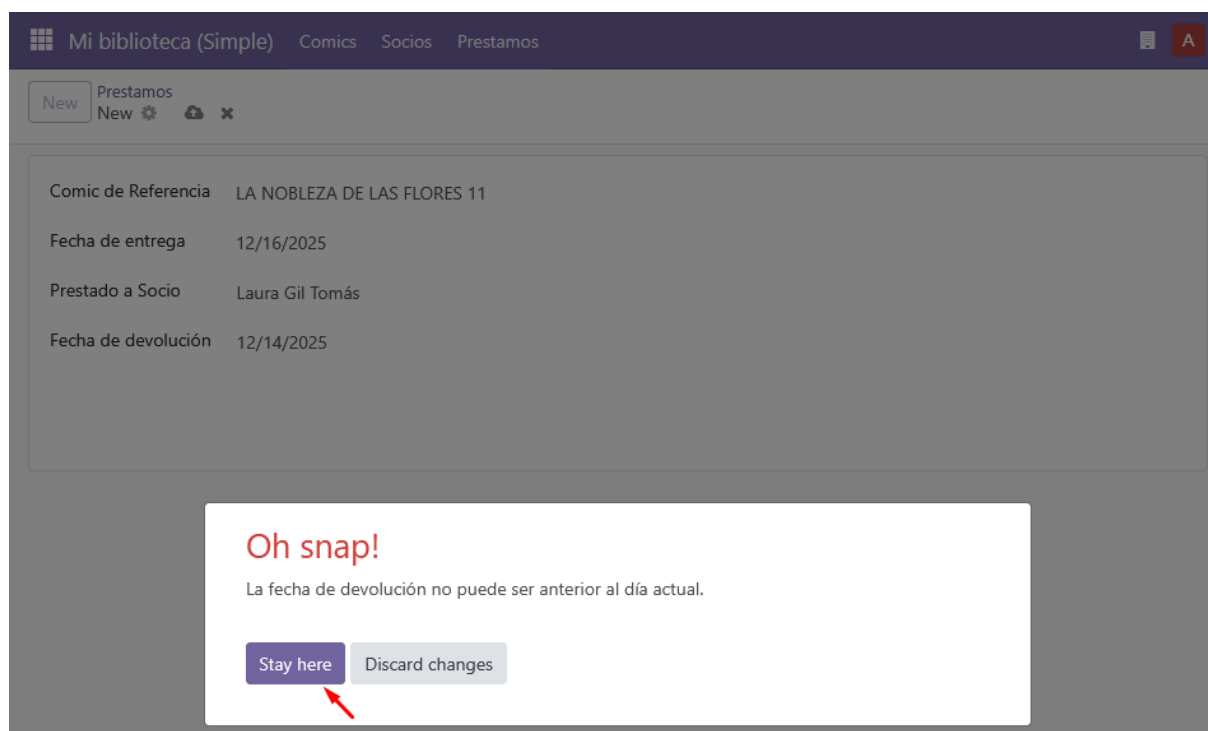


Figura 21: El día actual es el 15 por lo que funciona correctamente

The screenshot shows a web application header with a logo and navigation links: 'Mi biblioteca (Simple)', 'Comics', 'Socios', and 'Prestamos'. On the right, there are icons for a list and a user profile 'A'. Below the header, there's a sub-header with 'New', 'Prestamos', and 'New' with a gear icon. A red arrow points to a cloud icon with an 'x' next to it. The main content area contains a form with the following details:

| | |
|---------------------|-----------------------------|
| Comic de Referencia | LA NOBLEZA DE LAS FLORES 11 |
| Fecha de entrega | 12/15/2025 |
| Prestado a Socio | Laura Gil Tomás |
| Fecha de devolución | 12/16/2025 |

Figura 22: Corregimos las fechas y validamos

The screenshot shows the same web application header. Below it, there's a sub-header with 'New', 'Prestamos', and a gear icon. On the right, there's a pagination control showing '1-1 / 1' and navigation arrows. Below that is a search bar with a magnifying glass icon and the text 'Search...'. The main content area displays a table of loans:

| <input type="checkbox"/> Comic de Referencia | Prestado a Socio | Fecha de entrega | Fecha de devolución |
|--|------------------|------------------|---------------------|
| <input type="checkbox"/> LA NOBLEZA DE LAS FLORES 11 | Laura Gil Tomás | 12/15/2025 | 12/16/2025 |
| | | | |
| | | | |
| | | | |

Figura 23: El préstamo se ha creado correctamente

4. Actividad 03 - Hospital

En esta tercera actividad crearemos el módulo ‘**hospital**’ de cero para gestionar los pacientes y médicos de un centro de salud:

- **Modelo Paciente:**

- Nombre y apellidos del paciente.
- Sintomatología.

- **Modelo Médico:**

- Nombre y apellidos del médico.
- Número de colegiado.

- **Modelo Consulta:**

- Registro de cada vez que un médico atiende a un paciente, incluyendo el diagnóstico.
- Relaciones:
 - Un paciente puede haber sido atendido por varios médicos.
 - Un médico puede haber atendido a varios pacientes.

- Implementar los modelos y las vistas que se consideren oportunos.

A continuación se mostrarán las capturas del código realizado y las capturas con el módulo en funcionamiento.

```
paciente.py 1, U X  medico.py 1, U  consulta.py 1, U
addons > hospital > models > paciente.py > ...
1  from odoo import models, fields, api
2
3  class Paciente(models.Model):
4      # Nombre y descripción del modelo
5      _name = 'hospital.paciente'
6      _description = 'Paciente del hospital'
7
8      # Este atributo privado mostrará el nombre del socio en la relación con ejemplar.
9      _rec_name = 'nombre_completo'
10
11     # Propiedades
12     nombre = fields.Char('Nombre', required=True)
13     apellidos = fields.Char('Apellidos', required=True)
14     sintomatologia = fields.Text('Síntomas', required=True)
15     # Para mostrar el nombre completo del paciente en la relación usaremos un campo calculado
16     nombre_completo = fields.Char('Nombre Completo', compute='_compute_nombre_completo', store=True, index=True)
17
18     # RELACIÓN 1-N (para saber qué consultas tuvo este paciente)
19     consulta_ids = fields.One2many('hospital.consulta', 'paciente_id', string='Historial de Consultas')
20
21
22     # Campo calculado para concatenar nombre y apellido
23     @api.depends('nombre', 'apellidos')
24     def _compute_nombre_completo(self):
25         for record in self:
26             record.nombre_completo = f"{record.nombre} {record.apellidos}"
27
```

Figura 24: Modelo Paciente


```
paciente.py 1, U  medico.py 1, U X  consulta.py 1, U
addons > hospital > models > medico.py > Medico
1  from odoo import models, fields, api
2
3  class Medico(models.Model):
4      # Nombre y descripción del modelo
5      _name = 'hospital.medico'
6      _description = 'Médico del hospital'
7
8      # Este atributo privado mostrará el nombre del socio en la relación con ejemplar.
9      _rec_name = 'nombre_completo'
10
11     # Propiedades
12     nombre = fields.Char('Nombre', required=True)
13     apellidos = fields.Char('Apellidos', required=True)
14     numero_colegiado = fields.Char("Nº colegiado", required=True)
15     # Para mostrar el nombre completo del paciente en la relación usaremos un campo calculado
16     nombre_completo = fields.Char('Nombre Completo', compute='_compute_nombre_completo', store=True, index=True)
17
18     # RELACIÓN 1-N (Un médico puede haber atendido a muchos pacientes)
19     consulta_ids = fields.One2many('hospital.consulta', 'medico_id', string='Consultas Atendidas')
20
21
22     # Campo calculado para concatenar nombre y apellido
23     @api.depends('nombre', 'apellidos')
24     def _compute_nombre_completo(self):
25         for record in self:
26             record.nombre_completo = f"{record.nombre} {record.apellidos}"
27
```

Figura 25: Modelo Medico

```
paciente.py 1, U  medico.py 1, U  consulta.py 1, U X
addons > hospital > models > consulta.py > Consulta > diagnostico
1  from odoo import models, fields
2
3  class Consulta(models.Model):
4      # Nombre y descripción del modelo
5      _name = 'hospital.consulta'
6      _description = 'Registro de consultas realizadas'
7
8      # Propiedades
9      fecha_atencion = fields.Datetime(string='Fecha y Hora', default=fields.Datetime.now, required=True)
10     diagnostico = fields.Text(string='Diagnóstico')
11
12     # Relaciones N:1
13     paciente_id = fields.Many2one('hospital.paciente', string='Paciente', required=True)
14     medico_id = fields.Many2one('hospital.medico', string='Médico', required=True)
15
```

Figura 26: Modelo Consulta

```
__init__.py U X
addons > hospital > models > __init__.py
1  # -*- coding: utf-8 -*-
2
3  from . import consulta
4  from . import medico
5  from . import paciente
6
```

Figura 27: *__init__.py*

```
ir.model.access.csv U X
addons > hospital > security > ir.model.access.csv
1  id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2  access_paciente,hospital.paciente_default,model_hospital_paciente,,1,0,0,0
3  access_paciente_gestor,hospital.paciente_gestor,model_hospital_paciente,base.group_user,1,1,1,1
4  access_medico,hospital.medico_default,model_hospital_medico,,1,0,0,0
5  access_medico_gestor,hospital.medico_gestor,model_hospital_medico,base.group_user,1,1,1,1
6  access_consulta,hospital.consulta_default,model_hospital_consulta,,1,0,0,0
7  access_consulta_gestor,hospital.consulta_gestor,model_hospital_consulta,base.group_user,1,1,1,1
8
```

Figura 28: Reglas de acceso

```

__manifest__.py U X
addons > hospital > __manifest__.py
1  # -*- coding: utf-8 -*-
2  {
3      'name': "Hospital",
4
5      'summary': "Short (1 phrase/line) summary of the module's purpose",
6
7      'description': """
8  Long description of module's purpose
9  """,
10
11     'author': "My Company",
12     'website': "https://www.yourcompany.com",
13
14     # Categories can be used to filter modules in modules listing
15     # Check https://github.com/odoo/odoo/blob/15.0/odoo/addons/base/data/ir\_module\_category\_data.xml
16     # for the full list
17     'category': 'Uncategorized',
18     'version': '0.1',
19
20     # any module necessary for this one to work correctly
21     'depends': ['base'],
22
23     # always loaded
24     'data': [
25         'security/ir.model.access.csv',
26         'views/views.xml'
27     ],
28     # only loaded in demonstration mode
29     'demo': [
30         'demo/demo.xml',
31     ],
32 }

```

Figura 29: *__manifest__.py*

```

views.xml U X
addons > hospital > views > views.xml
1  <odoo>
2      <!-- Vista Pacientes -->
3      <record id="hospital_pacientes_view" model="ir.ui.view">
4          <field name="name">Pacientes habituales</field>
5          <field name="model">hospital.paciente</field>
6          <field name="arch" type="xml">
7              <list>
8                  <field name="nombre"/>
9                  <field name="apellidos"/>
10                 <field name="sintomatologia"/>
11             </list>
12         </field>
13     </record>
14
15     <!-- Vista Médicos -->
16     <record id="hospital_medicos_view" model="ir.ui.view">
17         <field name="name">Médicos en plantilla</field>
18         <field name="model">hospital.medico</field>
19         <field name="arch" type="xml">
20             <list>
21                 <field name="nombre"/>
22                 <field name="apellidos"/>
23                 <field name="numero_colegiado"/>
24             </list>
25         </field>
26     </record>
27
28     <!-- Vista Consultas -->
29     <record id="hospital_consultas_view" model="ir.ui.view">
30         <field name="name">Consultas realizadas</field>
31         <field name="model">hospital.consulta</field>
32         <field name="arch" type="xml">
33             <list>
34                 <field name="paciente_id"/>
35                 <field name="medico_id"/>
36                 <field name="diagnostico"/>
37                 <field name="fecha_atencion"/>
38             </list>
39         </field>
40     </record>

```

Figura 30: *view.xml* parte 1

```

views.xml U X
addons > hospital > views > views.xml
43 <!-- Accionador vista Pacientes -->
44 <record id='hospital_pacientes_action' model='ir.actions.act_window'>
45   <field name="name">Pacientes</field>
46   <field name="res_model">hospital.paciente</field>
47   <field name="view_mode">list,form</field>
48   <field name="view_id" ref="hospital_pacientes_view"/>
49 </record>
50
51 <!-- Accionador vista Médicos -->
52 <record id="hospital_medicos_action" model="ir.actions.act_window">
53   <field name="name">Médicos</field>
54   <field name="res_model">hospital.medico</field>
55   <field name="view_mode">list,form</field>
56   <field name="view_id" ref="hospital_medicos_view"/>
57 </record>
58
59 <!-- Accionador vista Consultas -->
60 <record id="hospital_consultas_action" model="ir.actions.act_window">
61   <field name="name">Consultas</field>
62   <field name="res_model">hospital.consulta</field>
63   <field name="view_mode">list,form</field>
64   <field name="view_id" ref="hospital_consultas_view"/>
65 </record>
66
67
68 <!-- MENÚES -->
69 <menuitem name="Hospital" id="hospital_base_menu"/>
70 <menuitem name="Pacientes" id="hospital_pacientes_menu" parent="hospital_base_menu" action="hospital_pacientes_action"/>
71 <menuitem name="Médicos" id="hospital_medicos_menu" parent="hospital_base_menu" action="hospital_medicos_action"/>
72 <menuitem name="Consultas" id="hospital_consultas_menu" parent="hospital_base_menu" action="hospital_consultas_action"/>
73 </odoo>
74

```

Figura 31: *view.xml* parte 2

A continuación capturas del módulo en funcionamiento.

Hospital

Pacientes

Médicos

Consultas

A

New Pacientes

1-3 / 3

<

>

Q

Search...

| <input type="checkbox"/> | Nombre | Apellidos | Síntomas |
|--------------------------|----------|-----------------|------------------------------------|
| <input type="checkbox"/> | Pepe | Fernandez Tolvo | Le duele el pecho |
| <input type="checkbox"/> | Gabriela | Estévez García | Fuerte dolor en la pierna |
| <input type="checkbox"/> | Agustín | Hernández Soto | Fuerte dolor en el hombro derecho. |
| | | | |
| | | | |

Figura 32: Vista Pacientes con ejemplos creados

Hospital

Pacientes

Médicos

Consultas

A

New

Médicos

1-2 / 2

<

>

Q

Search...

| <input type="checkbox"/> | Nombre | Apellidos | Nº colegiado |
|--------------------------|----------|----------------|--------------|
| <input type="checkbox"/> | Manuel | Estévez García | 112233333 |
| <input type="checkbox"/> | Adelaida | Pino García | 456233333 |
| | | | |
| | | | |
| | | | |

Figura 33: Vista Médicos con ejemplos creados

Hospital

Pacientes

Médicos

Consultas

A

New

Consultas

Search...

Paciente



Médico

Diagnóstico

Fecha y Hora

Figura 34: Vista Consultas vacía, creamos una nueva

Hospital Pacientes Médicos Consultas

New Consultas New  

Fecha y Hora 12/12/2025 18:30:56


Diagnóstico Pulmonía crónica.


Paciente Pepe Fernandez Tolvo

Médico Manuel Estévez García

Figura 35: Rellenamos los campos y validamos

Hospital Pacientes Médicos Consultas

New Consultas  1-1 / 1 < >

 Search...

| <input type="checkbox"/> Paciente | Médico | Diagnóstico | Fecha y Hora |
|---|-----------------------|-------------------|---------------------|
| <input type="checkbox"/> Pepe Fernandez Tolvo | Manuel Estévez García | Pulmonía crónica. | 12/12/2025 18:30:56 |

Figura 36: Resultado de crear una consulta

5. Actividad 04 - Instituto

En esta cuarta y última actividad desarrollaremos un módulo que represente los estudios de ciclos formativos en un instituto:

- **Modelo Ciclo Formativo:**

- Representa un ciclo formativo en el instituto.
- Un ciclo puede tener uno o más módulos asociados.

- **Modelo Módulo**, que relacionará con:

- El ciclo formativo al que pertenece.
- Los alumnos matriculados.
- El profesor que lo imparte.

- **Modelo Alumno:**

- Relacionado con los módulos en los que está matriculado.

- **Modelo Profesor:**

- Relacionado con los módulos que imparte.

- Se deberán implementar las relaciones necesarias y las vistas adecuadas para los cuatro modelos.
- Por último, se creará el rol de usuario *Director* (todos los permisos) y el rol *Profesor* (permiso de sólo lectura).

A continuación se mostrarán las capturas del código realizado y las capturas con el módulo en funcionamiento.


```

addons > instituto > models > alumno.py > ...
1  from odoo import fields, models, api
2
3  class Alumno(models.Model):
4      # Nombre y descripción del modelo
5      _name = 'instituto.alumno'
6      _description = 'Alumno de instituto'
7
8      # Este atributo privado mostrará el nombre del socio en la relación con ejemplar.
9      _rec_name = 'nombre_completo'
10
11     nombre = fields.Char(string='Nombre', required=True)
12     apellidos = fields.Char(string='Apellidos', required=True)
13     nombre_completo = fields.Char(string='Nombre Completo', compute='_compute_full_name', store=True)
14
15     # Relación N:M con Módulos
16     modulo_ids = fields.Many2many(
17         'instituto.modulo', 'modulo_alumno_id', 'alumno_id', 'modulo_id', string='Módulos Matriculados')
18
19     # Campo calculado para concatenar nombre y apellido
20     @api.depends('nombre', 'apellidos')
21     def _compute_nombre_completo(self):
22         for record in self:
23             record.nombre_completo = f"{record.nombre} {record.apellidos}"
24

```

Figura 37: Módulo Alumno

```

addons > instituto > models > profesor.py > ...
1  from odoo import fields, models, api
2
3  class Profesor(models.Model):
4      # Nombre y descripción del modelo
5      _name = 'instituto.profesor'
6      _description = 'Profesor de instituto'
7
8      # Este atributo privado mostrará el nombre del socio en la relación con ejemplar.
9      _rec_name = 'nombre_completo'
10
11     nombre = fields.Char(string='Nombre', required=True)
12     apellidos = fields.Char(string='Apellidos', required=True)
13     nombre_completo = fields.Char(string='Nombre Completo', compute='_compute_full_name', store=True)
14
15     # Relación 1:N con Módulos
16     modulo_ids = fields.One2many('instituto.modulo', 'profesor_id', string='Módulos Impartidos')
17
18     # Campo calculado para concatenar nombre y apellido
19     @api.depends('nombre', 'apellidos')
20     def _compute_nombre_completo(self):
21         for record in self:
22             record.nombre_completo = f"{record.nombre} {record.apellidos}"
23

```

Figura 38: Módulo Profesor

```
alumno.py 1, U  profesor.py 1, U  ciclo.py 1, U X  modulo.py 1, U

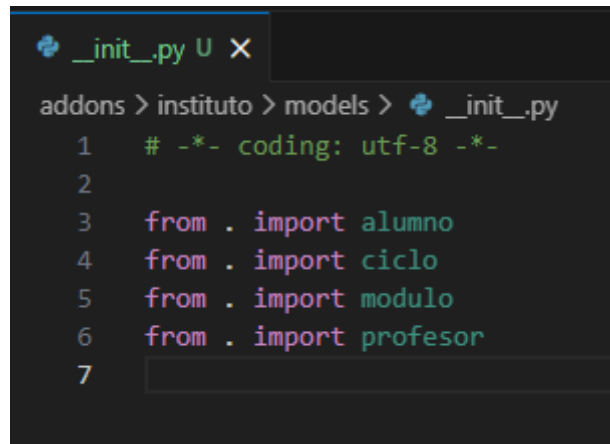
addons > instituto > models > ciclo.py > ...
1  from odoo import fields, models
2
3  class Ciclo(models.Model):
4      _name = 'instituto.ciclo'
5      _description = 'Ciclo Formativo'
6
7      nombre = fields.Char(string='Nombre del Ciclo', required=True)
8      codigo = fields.Char(string='Código', required=True, copy=False)
9
10     # Relación 1:N con Módulos
11     modulo_ids = fields.One2many('instituto.modulo', 'ciclo_id', string='Módulos del Ciclo')
12
```

Figura 39: Módulo Ciclo

```
alumno.py 1, U  profesor.py 1, U  ciclo.py 1, U  modulo.py 1, U X

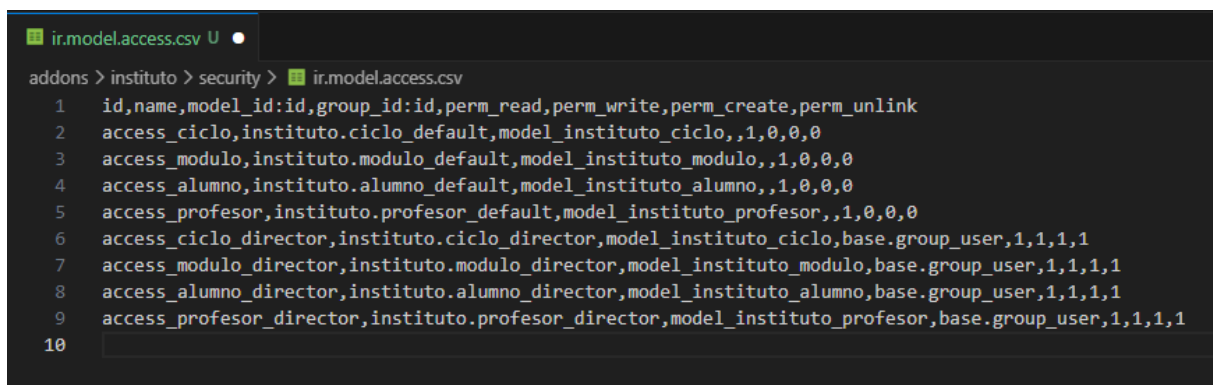
addons > instituto > models > modulo.py > ...
1  from odoo import fields, models
2
3  class Modulo(models.Model):
4      # Nombre y descripción del modelo
5      _name = 'instituto.modulo'
6      _description = 'Módulo Formativo'
7
8      nombre = fields.Char(string='Nombre del Módulo', required=True)
9      horas = fields.Integer(string='Horas Lectivas')
10
11     # Relación N:1 con Ciclo
12     ciclo_id = fields.Many2one('instituto.ciclo', string='Ciclo Formativo', required=True, ondelete='restrict')
13
14     # Relación N:1 con Profesor
15     profesor_id = fields.Many2one('instituto.profesor', string='Profesor')
16
17     # Relación N:M con Alumno
18     alumno_ids = fields.Many2many('instituto.alumno', 'modulo_alumno_id', 'modulo_id', 'alumno_id', string='Alumnos Matriculados')
19
```

Figura 40: Módulo Módulo



```
__init__.py U X
addons > instituto > models > __init__.py
1  # -*- coding: utf-8 -*-
2
3  from . import alumno
4  from . import ciclo
5  from . import modulo
6  from . import profesor
7
```

Figura 41: *__init__.py*



```
ir.model.access.csv U
addons > instituto > security > ir.model.access.csv
1  id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2  access_ciclo,instituto.ciclo_default,model_instituto_ciclo,,1,0,0,0
3  access_modulo,instituto.modulo_default,model_instituto_modulo,,1,0,0,0
4  access_alumno,instituto.alumno_default,model_instituto_alumno,,1,0,0,0
5  access_profesor,instituto.profesor_default,model_instituto_profesor,,1,0,0,0
6  access_ciclo_director,instituto.ciclo_director,model_instituto_ciclo,base.group_user,1,1,1,1
7  access_modulo_director,instituto.modulo_director,model_instituto_modulo,base.group_user,1,1,1,1
8  access_alumno_director,instituto.alumno_director,model_instituto_alumno,base.group_user,1,1,1,1
9  access_profesor_director,instituto.profesor_director,model_instituto_profesor,base.group_user,1,1,1,1
10
```

Figura 42: Módulo Módulo

```

__manifest__.py U X
addons > instituto > __manifest__.py
1  # -*- coding: utf-8 -*-
2  {
3      'name': "Instituto",
4
5      'summary': "Short (1 phrase/line) summary of the module's purpose",
6
7      'description': """
8  Long description of module's purpose
9  """,
10
11     'author': "My Company",
12     'website': "https://www.yourcompany.com",
13
14     # Categories can be used to filter modules in modules listing
15     # Check https://github.com/odoo/odoo/blob/15.0/odoo/addons/base/data/ir_module_category_data.xml
16     # for the full list
17     'category': 'Uncategorized',
18     'version': '0.1',
19
20     # any module necessary for this one to work correctly
21     'depends': ['base'],
22
23     # always loaded
24     'data': [
25         'security/ir.model.access.csv',
26         'views/views.xml'
27     ],
28     # only loaded in demonstration mode
29     'demo': [
30         'demo/demo.xml',
31     ],
32 }
33

```

Figura 43: `__manifest__.py`

```
views.xml U X
addons > instituto > views > views.xml
1 <odoo>
2   <!-- Vista Ciclos -->
3   <record id="instituto_ciclos_view" model="ir.ui.view">
4     <field name="name">Ciclos en curso</field>
5     <field name="model">instituto.ciclo</field>
6     <field name="arch" type="xml">
7       <list>
8         <field name="nombre"/>
9         <field name="codigo"/>
10      </list>
11    </field>
12  </record>
13
14  <!-- Vista Módulos -->
15  <record id="instituto_modulos_view" model="ir.ui.view">
16    <field name="name">Módulos del ciclo</field>
17    <field name="model">instituto.modulo</field>
18    <field name="arch" type="xml">
19      <list>
20        <field name="nombre"/>
21        <field name="horas"/>
22        <field name="ciclo_id"/>
23        <field name="profesor_id"/>
24      </list>
25    </field>
26  </record>
27
28  <!-- Vista Alumnos -->
29  <record id="instituto_alumnos_view" model="ir.ui.view">
30    <field name="name">Alumnos matriculados</field>
31    <field name="model">instituto.alumno</field>
32    <field name="arch" type="xml">
33      <list>
34        <field name="nombre"/>
35        <field name="apellidos"/>
36      </list>
37    </field>
38  </record>
```

Figura 44: *view.xml* parte 1

```
views.xml U X
addons > instituto > views > views.xml
40 <!-- Vista Profesores -->
41 <record id="instituto_profesores_view" model="ir.ui.view">
42   <field name="name">Profesores del centro</field>
43   <field name="model">instituto.profesor</field>
44   <field name="arch" type="xml">
45     <list>
46       <field name="nombre"/>
47       <field name="apellidos"/>
48     </list>
49   </field>
50 </record>
51
52 <!-- Accionador vista Ciclos -->
53 <record id='instituto_ciclos_action' model='ir.actions.act_window'>
54   <field name="name">Ciclos</field>
55   <field name="res_model">instituto.ciclo</field>
56   <field name="view_mode">list,form</field>
57   <field name="view_id" ref="instituto_ciclos_view"/>
58 </record>
59
60 <!-- Accionador vista Módulos -->
61 <record id="instituto_modulos_action" model="ir.actions.act_window">
62   <field name="name">Módulos</field>
63   <field name="res_model">instituto.modulo</field>
64   <field name="view_mode">list,form</field>
65   <field name="view_id" ref="instituto_modulos_view"/>
66 </record>
```

Figura 45: *view.xml* parte 2

```

views.xml U X
addons > instituto > views > views.xml
68 <!-- Accionador vista Alumnos -->
69 <record id="instituto_alumnos_action" model="ir.actions.act_window">
70   <field name="name">Alumnos</field>
71   <field name="res_model">instituto.alumno</field>
72   <field name="view_mode">list,form</field>
73   <field name="view_id" ref="instituto_alumnos_view"/>
74 </record>
75
76 <!-- Accionador vista Profesores -->
77 <record id="instituto_profesores_action" model="ir.actions.act_window">
78   <field name="name">Profesores</field>
79   <field name="res_model">instituto.profesor</field>
80   <field name="view_mode">list,form</field>
81   <field name="view_id" ref="instituto_profesores_view"/>
82 </record>
83
84 <!-- MENÚES -->
85 <menuitem name="Instituto" id="instituto_base_menu"/>
86 <menuitem name="Ciclos" id="instituto_ciclos_menu" parent="instituto_base_menu" action="instituto_ciclos_action"/>
87 <menuitem name="Modulos" id="instituto_modulos_menu" parent="instituto_base_menu" action="instituto_modulos_action"/>
88 <menuitem name="Alumnos" id="instituto_alumnos_menu" parent="instituto_base_menu" action="instituto_alumnos_action"/>
89 <menuitem name="Profesores" id="instituto_profesores_menu" parent="instituto_base_menu" action="instituto_profesores_action"/>
90 </odoo>
91

```

Figura 46: *view.xml* parte 3

A continuación capturas del módulo en funcionamiento.

Instituto

Ciclos

Modulos

Alumnos

Profesores

A

New

Ciclos

1-2 / 2

<

>

Q

Search...

| <div><input type="checkbox"/></div> <div>Nombre del Ciclo</div> | Código |
|---|--------|
| <div><input type="checkbox"/></div> <div>DAM</div> | 001 |
| <div><input type="checkbox"/></div> <div>DAW</div> | 002 |

Instituto

Ciclos

Modulos

Alumnos

Profesores

A

New

Módulos

1-1 / 1

search...

Nombre del Módulo

Horas Lec...

Ciclo Formativo

Profesor

Sistemas de Gestión

500

instituto.ciclo,1

Unnamed

| <div> <div> <div></div> <div>Instituto</div> </div> <div> <div>Ciclos</div> <div>Modulos</div> <div><u>Alumnos</u></div> <div>Profesores</div> </div> </div> <div> <div></div> <div>A</div> </div> | |
|--|-----------------|
| <div> <div>New</div> <div>Alumnos</div> <div></div> </div> <div>1-2 / 2</div> <div> <div><</div> <div>></div> </div> | |
| <div> <div>Q</div> <div>Search...</div> <div></div> </div> | |
| <input type="checkbox"/> Nombre | Apellidos |
| <input type="checkbox"/> Pepito | Pérez Fernández |
| <input type="checkbox"/> Manuela | García Olmo |
| | |

| <div> <div> <div></div> <div>Instituto</div> </div> <div> <div>Ciclos</div> <div>Modulos</div> <div>Alumnos</div> <div><u>Profesores</u></div> </div> </div> <div> <div></div> <div>A</div> </div> | |
|--|---------------------|
| <div> <div>New</div> <div>Profesores</div> <div></div> </div> <div>1-2 / 2</div> <div> <div><</div> <div>></div> </div> | |
| <div> <div>Q</div> <div>Search...</div> <div></div> </div> | |
| <input type="checkbox"/> Nombre | Apellidos |
| <input type="checkbox"/> Estaban | Hernández Fernández |
| <input type="checkbox"/> Estefanía | Gutiérrez Hernández |
| | |