

# ***Mòduls Odoo***

**Miguel Navarro Bellido**  
**SGE 2ºDAM**

# Índex

1. Introducció a la creació d'un mòdul .....	pág3
2. Activitat 1 .....	pág5
3. Activitat 2 .....	pág8
4. Activitat 3 .....	pág11

## Introducció a la creació d'un mòdul

Per crear un mòdul, no ho farem al directori on hi ha els mòduls oficials, perquè és més difícil de mantenir, així que el crearem en un directori diferent, és interessant per evitar problemes.

El crearem amb l'usuari odoo, perquè odoo pugui modificar els arxius, és més còmode treballar amb l'usuari odoo que amb l'usuari root.

Per a això, primer creem una contrasenya per a l'usuari odoo perquè pugui entrar.

Un cop creada la contrasenya, al cmd ens col·loquem com a usuari odoo amb el següent comandament:

**su odoo**

Ens demanarà la contrasenya i una vegada sent l'usuari odoo, ens col·loquem en el directori arrel, si entrem i no ens col·loca per defecte ho farem amb el següent comandament:

**cd ~**

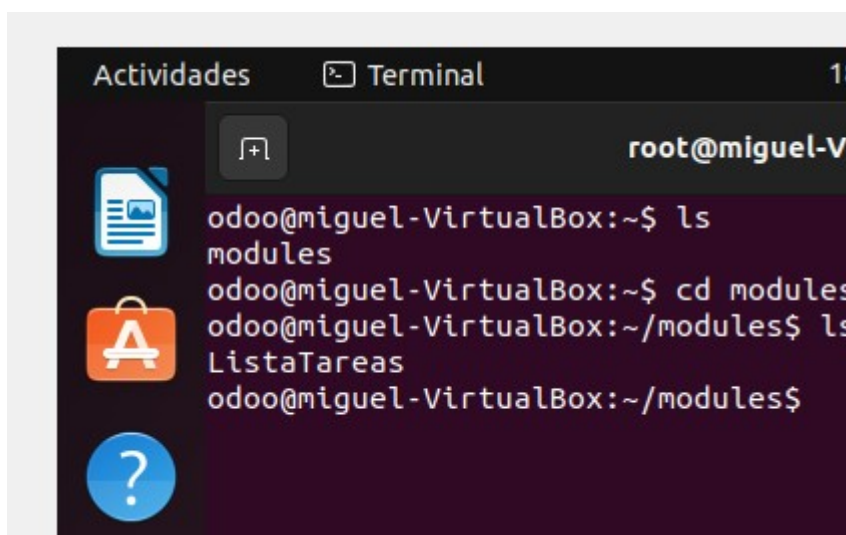
Ara crearem el directori a on col·locarem els mòduls que creem:

**mkdir modules**

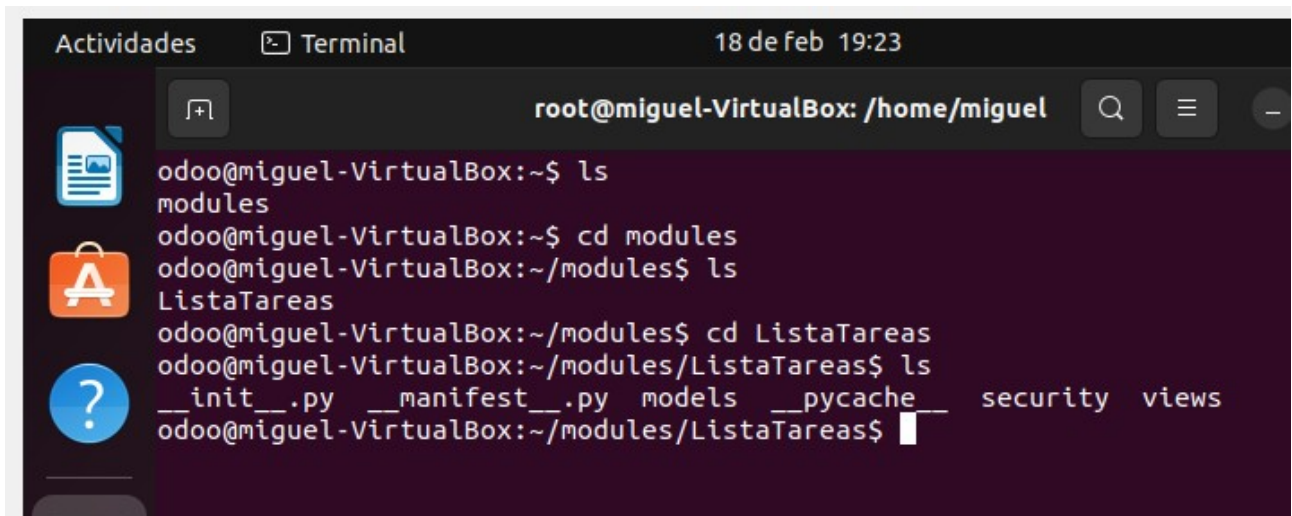
Dins del directori "modules" crearem el mòdul ListaTareas con scaffold.

**Odoo scaffold listaTareas**

Comprobem que s'ha creat.



Dins de lista\_tareas vorem els següents subdirectororis i fitxers.

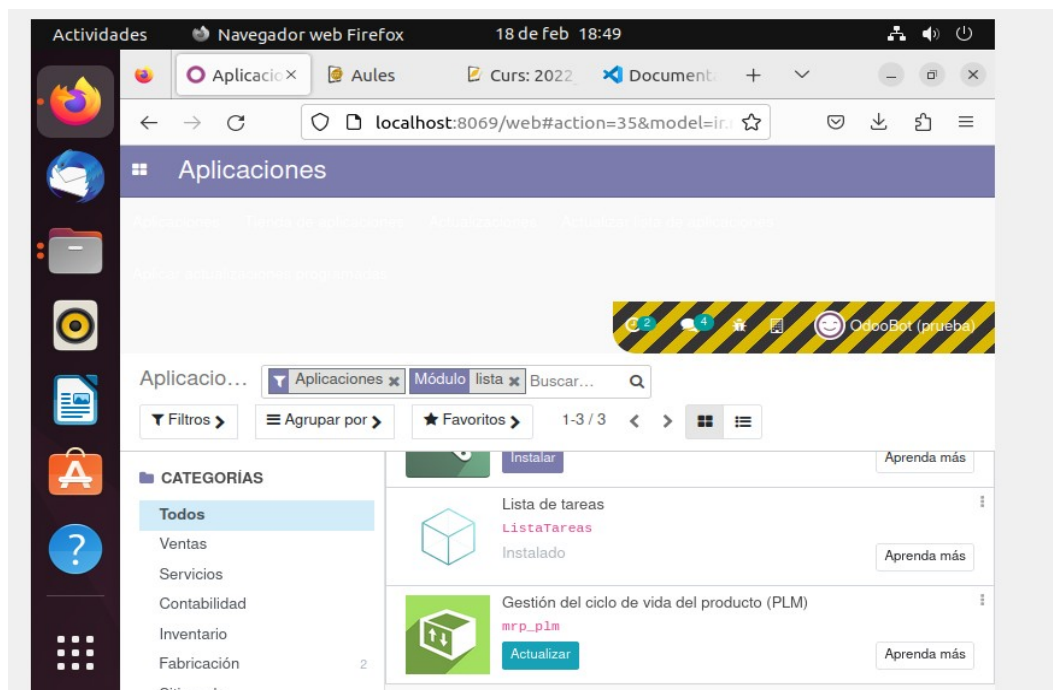


```
root@miguel-VirtualBox: /home/miguel
odoo@miguel-VirtualBox:~$ ls
modules
odoo@miguel-VirtualBox:~$ cd modules
odoo@miguel-VirtualBox:~/modules$ ls
ListaTareas
odoo@miguel-VirtualBox:~/modules$ cd ListaTareas
odoo@miguel-VirtualBox:~/modules/ListaTareas$ ls
__init__.py __manifest__.py models __pycache__ security views
odoo@miguel-VirtualBox:~/modules/ListaTareas$
```

Situats a la carpeta "modulos" utilitzem el següent comandament per donar permisos totals a l'usuari "odoo" sobre lista\_tareas.

**Sudo chmod 777 ./listaTareas**

Un cop creat, ens vam odoo, posem el mode desenvolupador, i actualitzem la llista d'aplicacions. Un cop realitzat busquem el mòdul llista de tasques i l'instal·lem.



### Exercici 04.01: Modifica l'exemple més senzill de la llista de tasques de forma que:

- Tinga una nova vista per veure les tasques en format "Kanban".

En primer lloc modifiquem la vista perquè les tasques es vegin en format Kanban, jo he utilitzat com a ide **Visual studio code**.

Per realitzar-lo ens anem a l'arxiu views.xml i el modifiquem afegint una acció, un menú ítem i una vista kanban.

```
12 <field name="name">Listado de tareas pendientes</field>
13 <field name="res_model">lista_tareas.lista</field>
14 <field name="view_mode">tree,form</field>
15 </record>
16
17 <record model="ir.actions.act_window" id="action_lista_tareas K
18 <!-- -->
19 <field name="name">Listado de tareas pendientes vista Kanban<
20 <field name="res_model">lista_tareas.lista</field>
21 <field name="view_mode">kanban</field>
22 </record>
23
24
```

```
33
34 <menuitem name="Opciones Lista Tareas" id="lista_tareas_me
35
36 <!-- actions -->
37
38 <menuitem name="Mostrar formato kanban" id="lista_tareas_m
39
40 <menuitem name="Mostrar lista" id="lista tareas menu 1 lis
```

Com que la màquina virtual amb la qual estic treballant em dona un error en intentar instal·lar les guest additions no puc realitzar les captures a pantalla completa, per la qual cosa poso el codi de la imatge a continuació.

```
<menuitem name="Mostrar formato kanban" id="lista_tareas_menu_2"
parent="lista_tareas_menu_1" action="action_lista_tareas_Kanban"/>
```

```

68 <!--Para kanban-->
69 <record model="ir.ui.view" id="lista_tareas_kanban">
70   <field name="name">Lista de tareas vista kanban</field>
71   <field name="model">lista_tareas.lista</field>
72   <field name="arch" type="xml">
73     <kanban>
74       <field name="tarea"/>
75       <field name="prioridad"/>
76       <field name="urgente"/>
77       <field name="realizada"/>
78     <templates>
79       <t t-name="kanban-box">
80         <div class="oe_kanban_global_click">
81           <div>
82             <field name="tarea"/>
83           </div>
84           <div class="oe_kanban_details">
85             <div>
86               | Prioridad: <field name="prioridad"/>
87             </div>
88             <div>
89               | Realizada: <field name="realizada"/>
90             </div>
91           </div>
92         </div>
93       </t>
94     </templates>
95   </kanban>
96 </field>
97 </record>

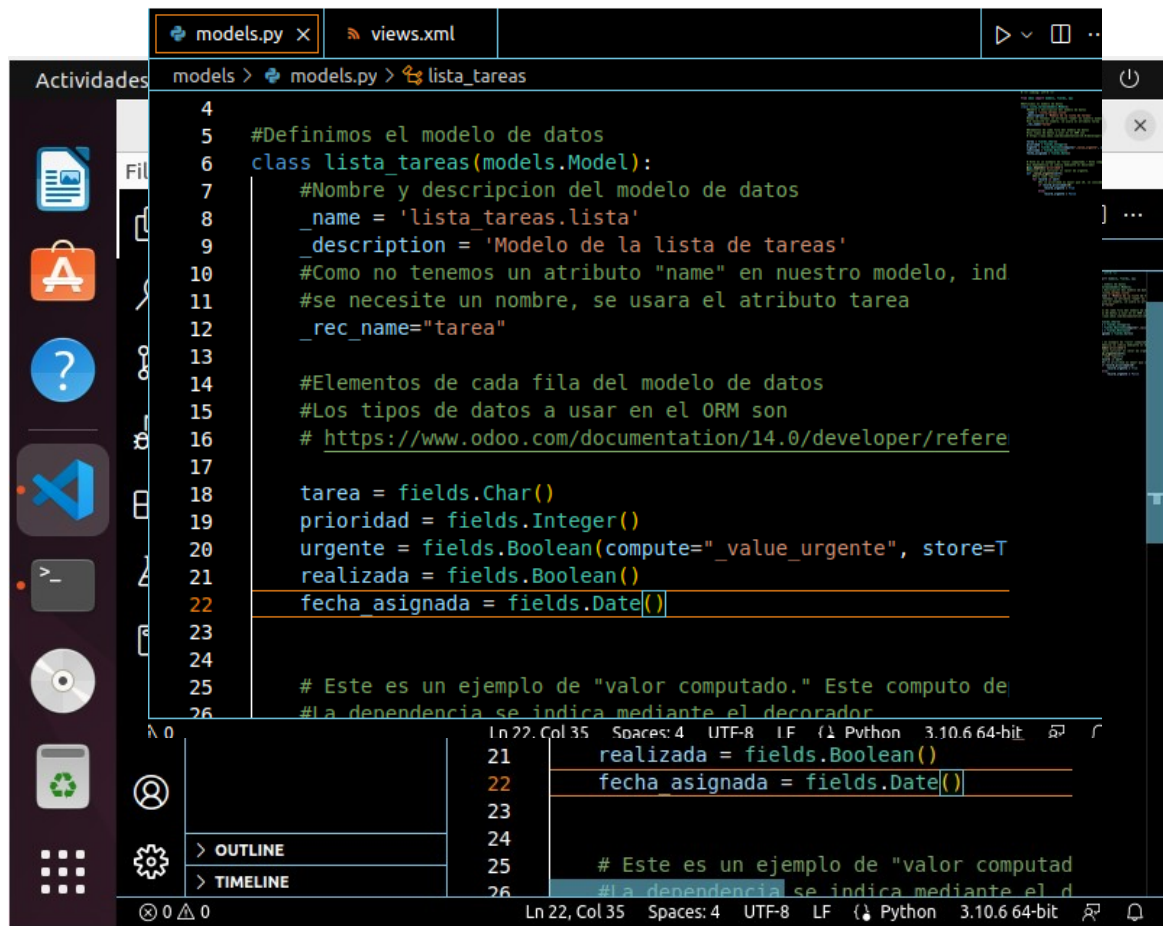
```

Go to Line/Column

- Modifica les tasques per tindre una data assignada i crea una nova vista per veure en una vista "Calendar" la data assignada.

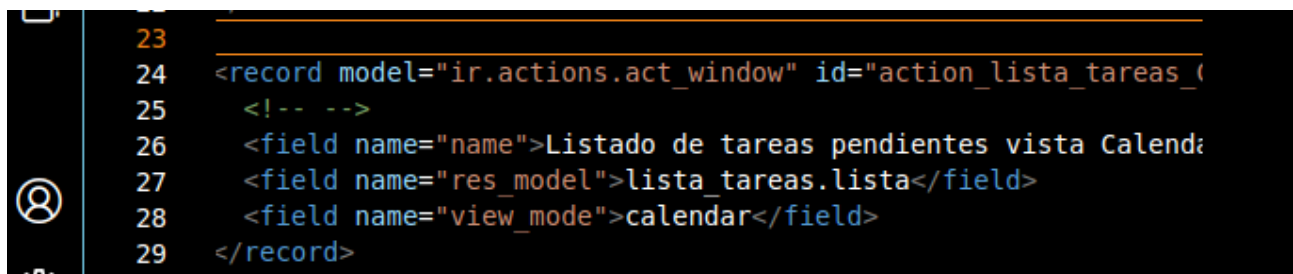
En primer lloc modifiquem model.py.

En aquest cas assignem a la variable "fecha\_asignada" un camp de tipus data.



```
4
5 #Definimos el modelo de datos
6 class lista_tareas(models.Model):
7     #Nombre y descripcion del modelo de datos
8     _name = 'lista_tareas.lista'
9     _description = 'Modelo de la lista de tareas'
10    #Como no tenemos un atributo "name" en nuestro modelo, ind
11    #se necesite un nombre, se usara el atributo tarea
12    _rec_name="tarea"
13
14    #Elementos de cada fila del modelo de datos
15    #Los tipos de datos a usar en el ORM son
16    # https://www.odoo.com/documentation/14.0/developer/refere
17
18    tarea = fields.Char()
19    prioridad = fields.Integer()
20    urgente = fields.Boolean(compute="_value_urgente", store=T
21    realizada = fields.Boolean()
22    fecha_asignada = fields.Date()
23
24
25    # Este es un ejemplo de "valor computado." Este computo de
26    #La dependencia se indica mediante el decorador
27
28    realizada = fields.Boolean()
29    fecha_asignada = fields.Date()
30
31    # Este es un ejemplo de "valor computad
32    #La dependencia se indica mediante el d
```

En segon lloc, realitzem els mateixos passos que en la vista kanban, anem a l'arxiu views.xml i el modifiquem afegint una acció, un menú ítem i una vista calendar.



```
23
24 <record model="ir.actions.act_window" id="action_lista_tareas_"
25     <!-- -->
26     <field name="name">Listado de tareas pendientes vista Calend
27     <field name="res_model">lista_tareas.lista</field>
28     <field name="view_mode">calendar</field>
29 </record>
```

```

42  </actions>
43
44  <menuitem name="Mostrar formato kanban" id="lista_tareas_menu_2"
45
46  <menuitem name="Mostrar formato calendar" id="lista_tareas_menu_3"
47

```

<menuitem name="Mostrar formato calendar" id="lista\_tareas\_menu\_3"  
parent="lista\_tareas\_menu\_1" action="action\_lista\_tareas\_calendar"/>

```

97  </record>
98  <!--Para Calendar-->
99  <record model="ir.ui.view" id="lista_tareas_calendario">
100    <field name="name">Lista de tareas vista calendario</field>
101    <field name="model">lista_tareas.lista</field>
102    <field name="arch" type="xml">
103      <calendar date_start="fecha_asignada" color="fecha_asignada"
104        <field name="fecha_asignada"/>
105      </calendar>
106    </field>
107  </record>
108  data>

```

## Exercici 04.02: Amplia el mòdul de l'exemple de biblioteca de còmics de forma que:

- Incloure la possibilitat de gestionar socis, emmagatzemant nom, cognom e identificador.

Per a aquest cas ho vaig fer tant en visual studio code de la màquina virtual com en visual studio code del host perquè les captures surtin més decents.

El primer pas és crear una classe per als socis de la biblioteca. Ho farem a l'arxiu biblioteca\_comic.py.

```

111  #Si procede, lanzamos una excepcion
112  raise models.ValidationError('La fecha de lanzamiento debe ser anterior a la actual')
113  #Creamos la clase BibliotecaSocio para poder gestionar los socios
114  class BibliotecaSocio(models.AbstractModel):
115      #Nombre y descripción del modelo
116      _name='biblioteca.socio'
117      nombre = fields.Char(string="Nombre", required=True)
118      apellido = fields.Char(string="Apellido", required=True)
119      identificador = fields.Char(string="Identificador", required=True)
120

```



El següent serà modificar l'arxiu biblioteca\_comic.xml afegint una acció, un menú ítem i una vista.

```
13
14     <!--Para el socio-->
15     <record id='biblioteca_socio_action' model='ir.actions.act_window'>
16         <field name="name">Socio Biblioteca</field>
17         <!-- Indicamos a que modelo aplica -->
18         <field name="res_model">biblioteca.socio</field>
19         <!-- Indicamos que los socios pueden verse en tree para el listado,
20          y en form para la creación/edición -->
21         <field name="view_mode">tree,form</field>
22     </record>
23
```

```
24
25     <!-- Simple menu item, sin más utilidad que la didáctica -->
26     <menuitem name="Mi biblioteca (Simple)" id="biblioteca_base_menu" />
27     <!--Añadimos un menú para poder seleccionar entre comic o socio-->
28     <menuitem name="Comics" id="biblioteca_opciones_menu" parent="biblioteca_base_menu"/>
29     <menuitem name="Comics" id="biblioteca_comic_menu" parent="biblioteca_opciones_menu" action="biblioteca_comic_action"/>
30     <!--Añadimos el menuitem del socio-->
31     <menuitem name="Comics" id="biblioteca_socio_menu" parent="biblioteca_opciones_menu" action="biblioteca_socio_action"/>
32
```

```
103
104     <!--Para el socio-->
105     <record id="biblioteca_socio_view_tree" model="ir.ui.view">
106         <field name="name">Búsqueda de Socios en la biblioteca</field>
107         <field name="model">biblioteca.socio</field>
108         <field name="arch" type="xml">
109             <form string="Partner">
110                 <sheet>
111                     <group>
112                         <field name="nombre"/>
113                         <field name="apellido"/>
114                         <field name="identificador"/>
115                     </group>
116                 </sheet>
117             </form>
118         </field>
119     </record>
120
```

- Introduïu la possibilitat que hi haja exemplars de còmics per prestar. Aquests exemplars de préstec hauran de controlar només a qui estan prestats i la data d'inici i data de final del préstec. No cal tindre un històric de préstecs, només qui té el còmic en cada moment, quan se l'ha prestat i la data prevista de tornada. (la data de préstec no pot ser posterior al dia de hui i la data prevista de tornada no pot ser anterior al dia de hui)

Per poder prestar comics cal relacionar la classe BibliotecaComic amb la classe BibliotecaSocio, a més, per saber si un còmic està disponible o no afegirem un camp booleà.

```
50     _rec_name = nombre
51
52     socio = fields.Many2one('biblioteca.socio' , string='prestamo')
53     disponible = fields.Boolean(['Disponible'])
54
55     #Atributo nombre
56     nombre = fields.Char('Titulo', required=True, index=True)
```

Ara s'afegeix al archiu biblioteca\_comic.xml la vista.

```
75
76     <record id="biblioteca_comic_prestamo_view_form" model="ir.ui.view">
77         <field name="name">vista de prestamo de la biblioteca</field>
78         <field name="model">biblioteca.comic</field>
79         <field name="arch" type="xml">
80             <form string="prestamo">
81                 <group>
82                     <field name="nombre"/>
83                     <field name="disponible"/>
84                     <field name="socio"/>
85                 </group>
86             </form>
87         </field>
88     </record>
```

### Exercici 04.03: Crea un mòdul per a gestionar pacients i metges d'un hospital.

Per crear el mòdul segueix els mateixos passos que en l'exercici 1 amb listaTareas, però aquesta vegada ho crec en el directori addons, amb scaffold.

Col·locant-nos en el directori addons:

**odoo scaffold hospital**

i li donarem permís a l'usuari odoo perquè pugui fer canvis al seu interior.

**Sudo chmod 777 ./hospital**

Per cada pacient, tindrem un model amb les següents dades:

- Nom i cognoms del pacient.
- Síntomes.

Per realitzar aquesta part modificarem l'arxiu models.py on li canviarem el nom per paciente.py i crearem el model pacient.

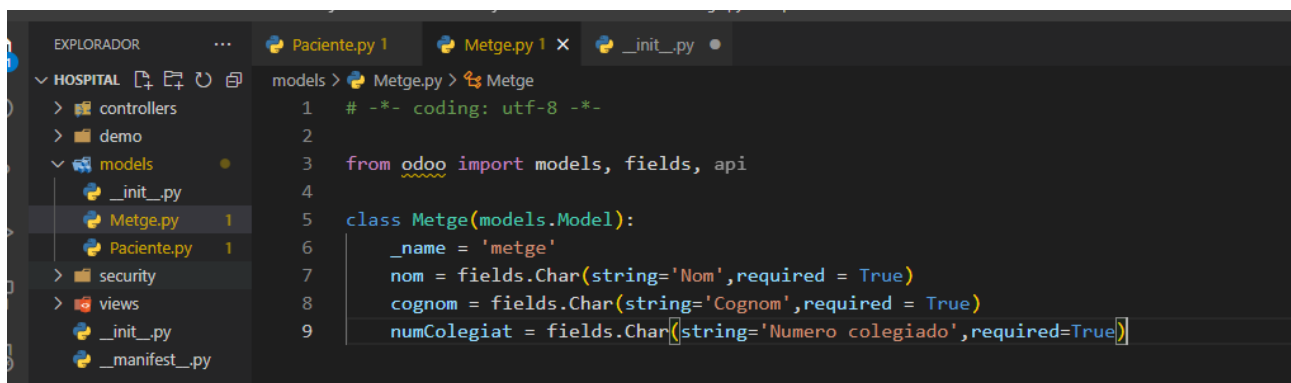


The screenshot shows the VS Code interface with the 'HOSPITAL' module selected in the Explorer. The 'models' directory is expanded, showing files like \_\_init\_\_.py, paciente.py, security, and views. The 'paciente.py' file is open in the editor, displaying the following code:

```
models > paciente.py > paciente
1  # -*- coding: utf-8 -*-
2
3  from odoo import models, fields, api
4
5  class paciente(models.Model):
6      _name = 'paciente'
7      nom = fields.Char(string='Nom',required = True)
8      cognom = fields.Char(string='Cognom',required = True)
9      sintomas = fields.Char(string='Síntomes',required=True)
10
```

Per cada metge, tindrem un model amb les següents dades:

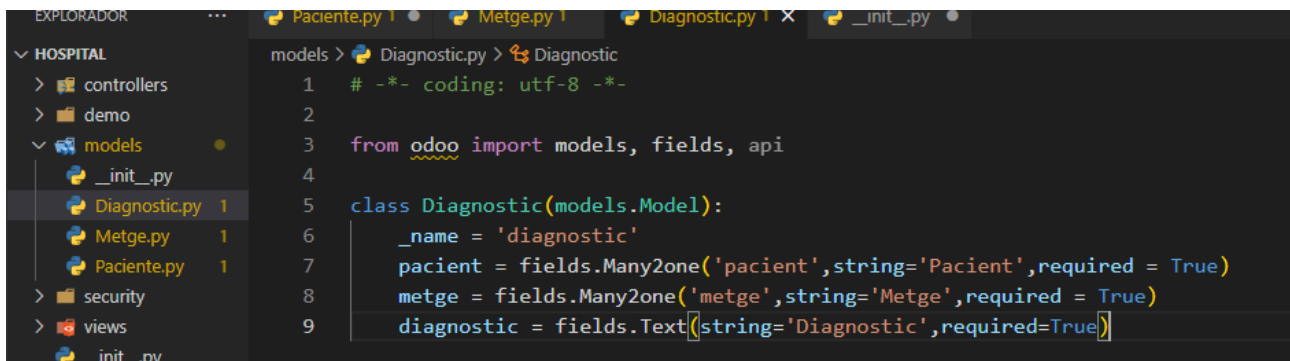
- Nom i cognoms del metge.
- Número de col·legiat.



The screenshot shows the VS Code interface with the 'HOSPITAL' module selected in the Explorer. The 'models' directory is expanded, showing files like \_\_init\_\_.py, Metge.py, and Paciente.py. The 'Metge.py' file is open in the editor, displaying the following code:

```
models > Metge.py > Metge
1  # -*- coding: utf-8 -*-
2
3  from odoo import models, fields, api
4
5  class Metge(models.Model):
6      _name = 'metge'
7      nom = fields.Char(string='Nom',required = True)
8      cognom = fields.Char(string='Cognom',required = True)
9      numColegiat = fields.Char(string='Número colegiado',required=True)
```

Per cada vegada que un metge ha atés a un pacient, tindrem un model indicant el diagnòstic. Un pacient pot haver sigut atés per diversos metges i un metge pot haver atés a diversos pacients.



The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure: HOSPITAL > controllers, demo, models (containing \_\_init\_\_.py, Diagnostic.py, Metge.py, and Paciente.py), security, and views. The main editor window shows the content of Diagnostic.py:

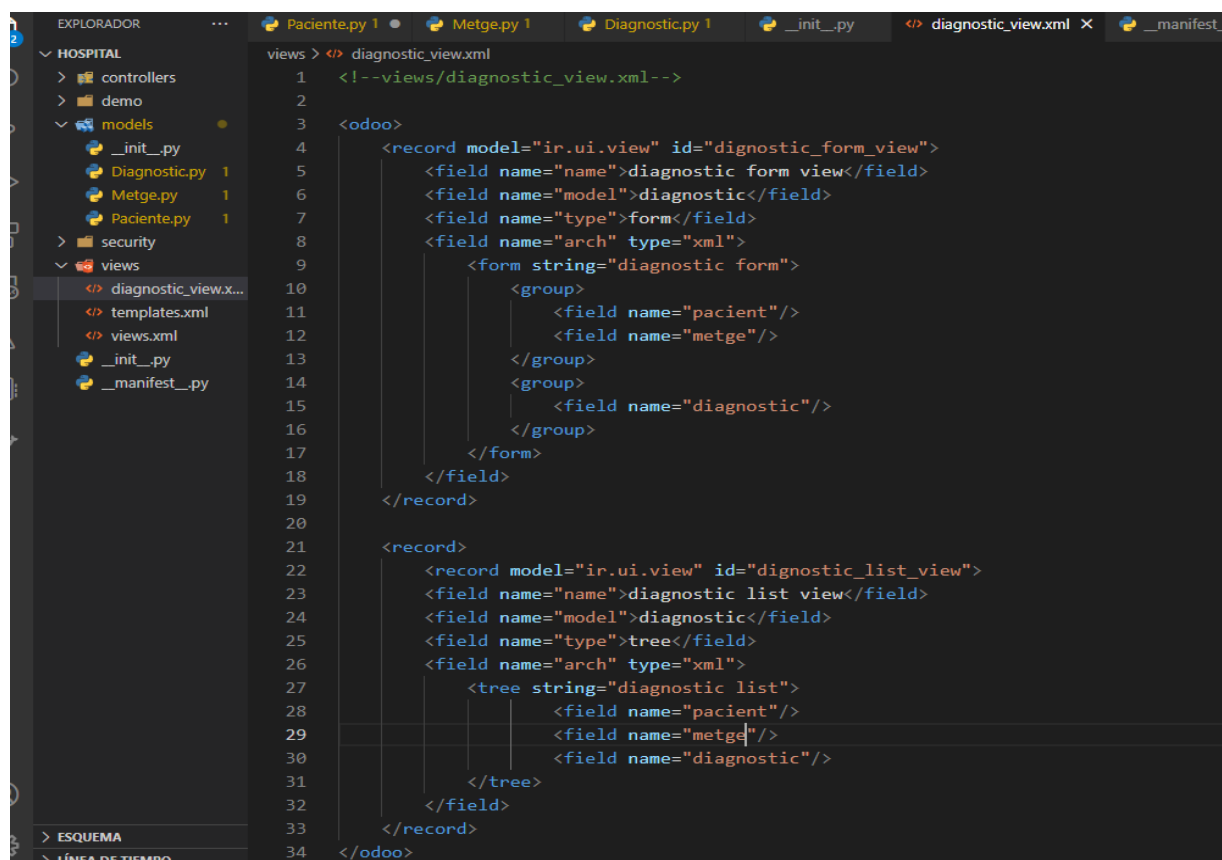
```
1  # -*- coding: utf-8 -*-
2
3  from odoo import models, fields, api
4
5  class Diagnostic(models.Model):
6      _name = 'diagnostic'
7      paciente = fields.Many2one('paciente', string='Pacient', required = True)
8      metge = fields.Many2one('metge', string='Metge', required = True)
9      diagnostic = fields.Text(string='Diagnostic', required=True)
```

Implementa els models i les vistes que cregues adequades per als 3 models.

Un cop ja tenim els models passem a les vistes.

Crearem un arxiu xml per a cadascun dels models.

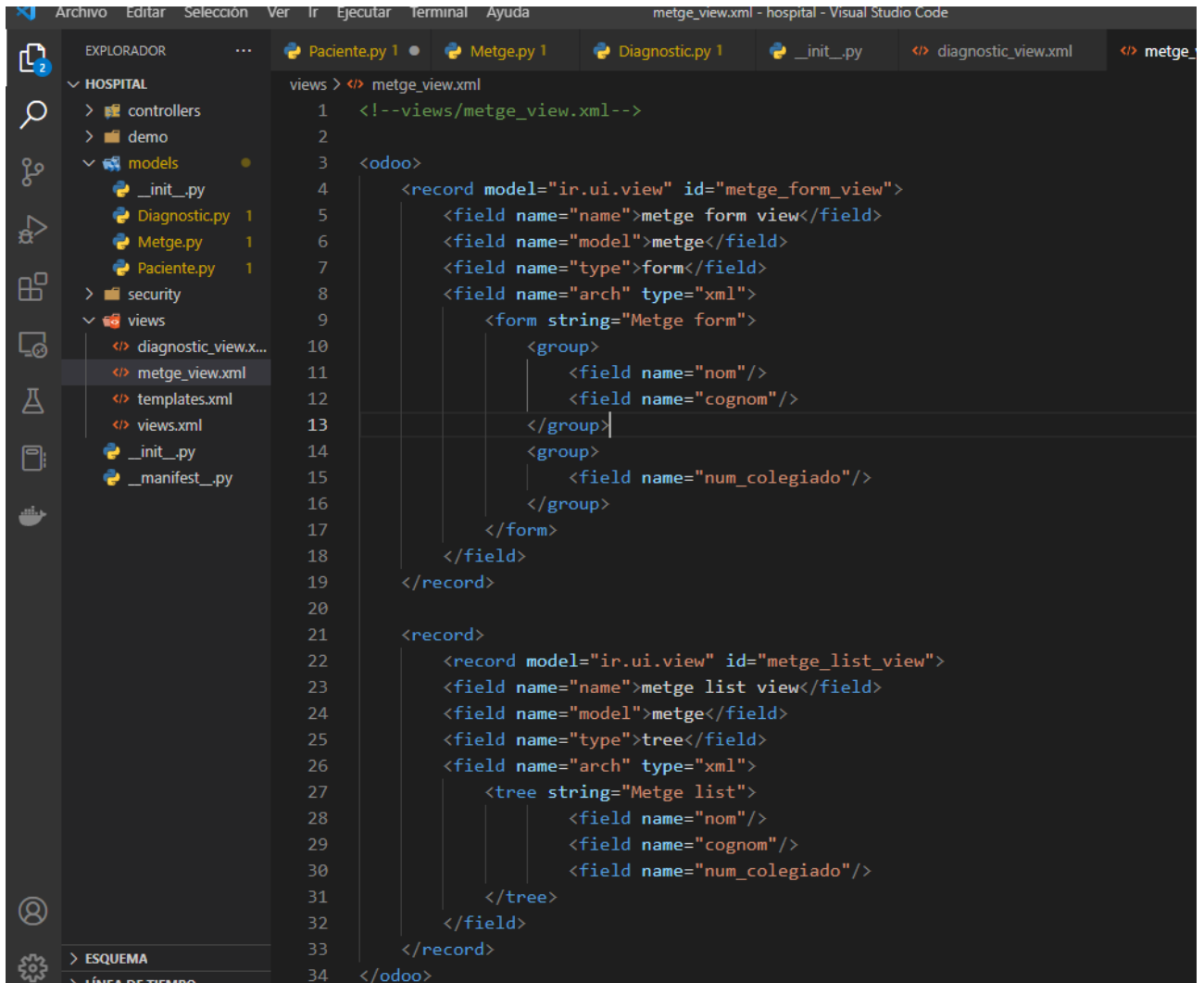
## Diagnostic



The screenshot shows the VS Code editor with the file explorer on the left displaying the project structure, including the views directory. The main editor window shows the content of diagnostic\_view.xml:

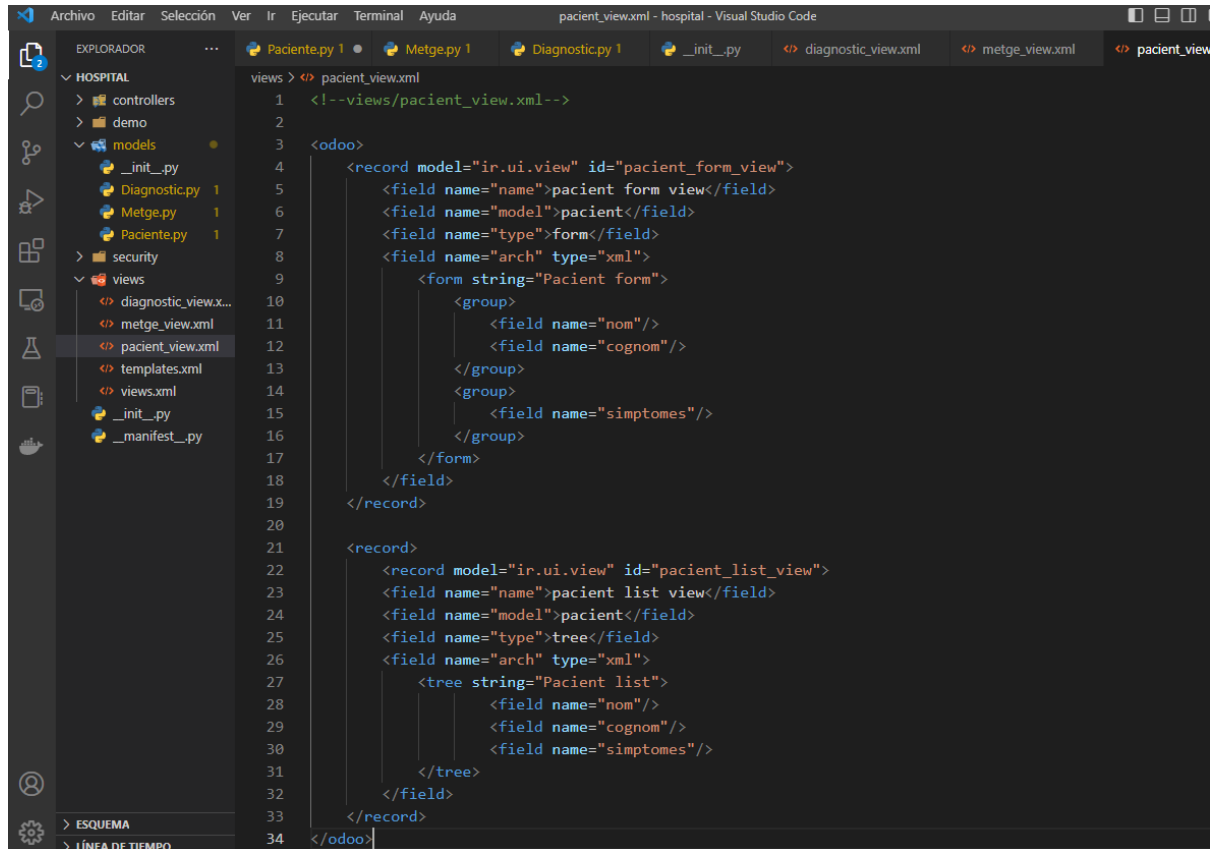
```
1  <!--views/diagnostic_view.xml-->
2
3  <odoo>
4      <record model="ir.ui.view" id="dignostic_form_view">
5          <field name="name">diagnostic form view</field>
6          <field name="model">diagnostic</field>
7          <field name="type">form</field>
8          <field name="arch" type="xml">
9              <form string="diagnostic form">
10                  <group>
11                      <field name="paciente"/>
12                      <field name="metge"/>
13                  </group>
14                  <group>
15                      <field name="diagnostic"/>
16                  </group>
17              </form>
18          </field>
19      </record>
20
21      <record>
22          <record model="ir.ui.view" id="dignostic_list_view">
23              <field name="name">diagnostic list view</field>
24              <field name="model">diagnostic</field>
25              <field name="type">tree</field>
26              <field name="arch" type="xml">
27                  <tree string="diagnostic list">
28                      <field name="paciente"/>
29                      <field name="metge"/>
30                      <field name="diagnostic"/>
31                  </tree>
32              </field>
33          </record>
34      </odoo>
```

## Metge



```
1  <!--views/metge_view.xml-->
2
3  <odoo>
4      <record model="ir.ui.view" id="metge_form_view">
5          <field name="name">metge form view</field>
6          <field name="model">metge</field>
7          <field name="type">form</field>
8          <field name="arch" type="xml">
9              <form string="Metge form">
10                  <group>
11                      <field name="nom"/>
12                      <field name="cognom"/>
13                  </group>
14                  <group>
15                      <field name="num_colegiado"/>
16                  </group>
17              </form>
18          </field>
19      </record>
20
21      <record>
22          <record model="ir.ui.view" id="metge_list_view">
23              <field name="name">metge list view</field>
24              <field name="model">metge</field>
25              <field name="type">tree</field>
26              <field name="arch" type="xml">
27                  <tree string="Metge list">
28                      <field name="nom"/>
29                      <field name="cognom"/>
30                      <field name="num_colegiado"/>
31                  </tree>
32              </field>
33          </record>
34      </odoo>
```

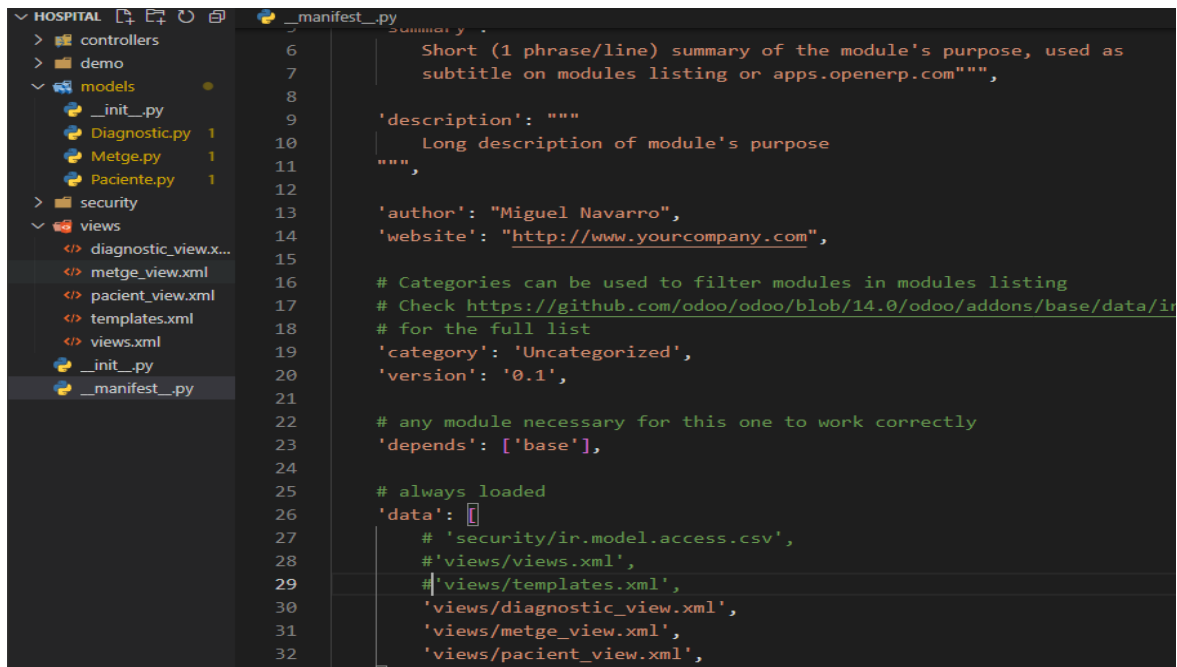
## Pacient



```
1 <!--views/pacient_view.xml-->
2
3 <odoo>
4     <record model="ir.ui.view" id="pacient_form_view">
5         <field name="name">pacient form view</field>
6         <field name="model">pacient</field>
7         <field name="type">form</field>
8         <field name="arch" type="xml">
9             <form string="Pacient form">
10                 <group>
11                     <field name="nom"/>
12                     <field name="cognom"/>
13                 </group>
14                 <group>
15                     <field name="simptome"/>
16                 </group>
17             </form>
18         </field>
19     </record>
20
21     <record>
22         <record model="ir.ui.view" id="pacient_list_view">
23             <field name="name">pacient list view</field>
24             <field name="model">pacient</field>
25             <field name="type">tree</field>
26             <field name="arch" type="xml">
27                 <tree string="Pacient list">
28                     <field name="nom"/>
29                     <field name="cognom"/>
30                     <field name="simptome"/>
31                 </tree>
32             </field>
33         </record>
34     </odoo>
```

## \_\_Manifest\_\_.py

L'indiquem les vistes que ha de carregar.



```
6
7     Short (1 phrase/line) summary of the module's purpose, used as
8     subtitle on modules listing or apps.openERP.com""",
9
10     'description': """
11         Long description of module's purpose
12     """,
13
14     'author': "Miguel Navarro",
15     'website': "http://www.yourcompany.com",
16
17     # Categories can be used to filter modules in modules listing
18     # Check https://github.com/odoo/odoo/blob/14.0/odoo/addons/base/data/ir
19     # for the full list
20     'category': 'Uncategorized',
21     'version': '0.1',
22
23     # any module necessary for this one to work correctly
24     'depends': ['base'],
25
26     # always loaded
27     'data': [
28         # 'security/ir.model.access.csv',
29         #'views/views.xml',
30         #'views/templates.xml',
31         'views/diagnostic_view.xml',
32         'views/metge_view.xml',
33         'views/pacient_view.xml',
34     ]
```