

# Ola Mundo

## a) Instalación (creación do modelo,táboa)

### Configuración en Odoo da ruta onde crearemos os módulos

Imos configurar Odoo para facilitarlle a ruta onde teremos os nosos módulos coa finalidade de instalalos en Odoo.

Para elo:

- Editamos o arquivo de configuración /etc/odoo/odoo.conf e engadimos, no meu caso, a ruta /home/antonio/PycharmProjects
- Despois reiniciamos o servizo de Odoo

```
root@voyager:/etc/odoo# ls -l
total 4
-rw-r----- 1 odoo odoo 1390 Xuñ 15 12:04 odoo.conf
root@voyager:/etc/odoo# nano odoo.conf
root@voyager:/etc/odoo# cat odoo.conf | grep addons
addons_path = /usr/lib/python3/dist-packages/odoo/addons,/home/antonio/PycharmProjects
root@voyager:/etc/odoo# ls -l
total 4
-rw-r----- 1 odoo odoo 1390 Xuñ 15 12:04 odoo.conf
root@voyager:/etc/odoo# systemctl restart odoo
root@voyager:/etc/odoo#
```

### Creación do primeiro módulo

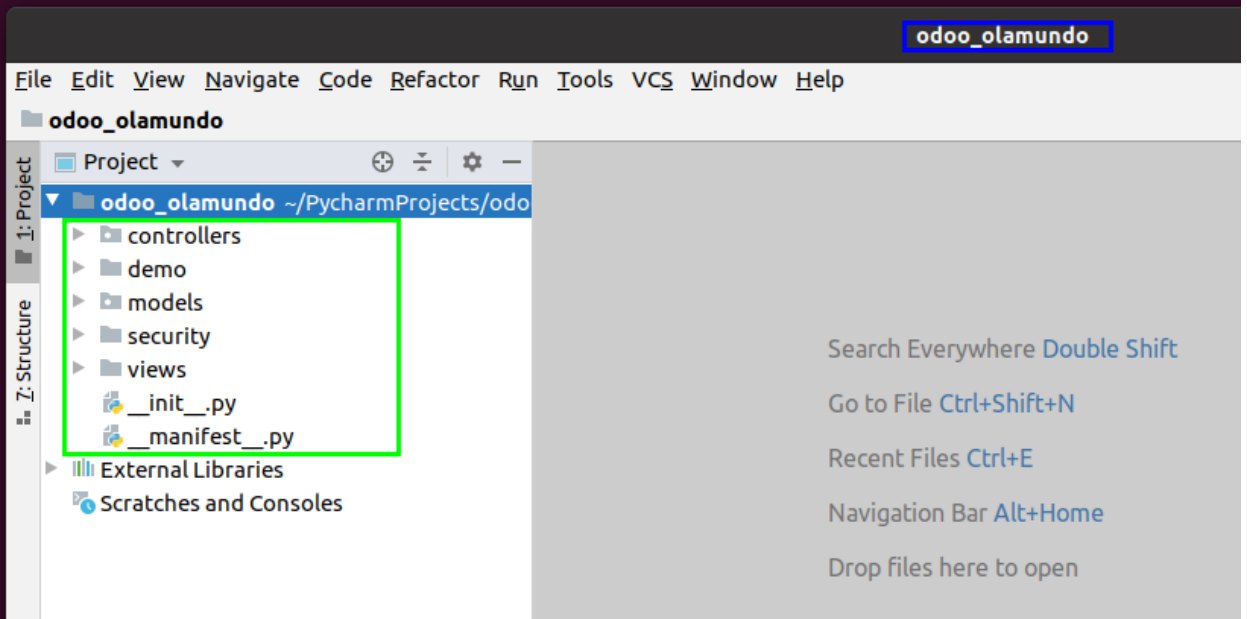
Imos crear o primeiro módulo para instalalo en Odoo:

- Usamos o comando **odoo scaffold nome\_do\_novo\_modulo**
- Vemos como crea o cartafol co nome do módulo e dentro del un esqueleto de cartafóis e documentos
- Abrimos o módulo (proxecto) con pycharm e podemos ver o noso proxecto(módulo) creado

```

antonio@voyager:~/PycharmProjects$ odoo scaffold odoo_olamundo
antonio@voyager:~/PycharmProjects$ ls -l
total 8
drwxrwxr-x 7 antonio antonio 4096 Xuñ 15 12:08 odoo_olamundo
drwxrwxr-x 4 antonio antonio 4096 Xuñ 11 17:33 untitled
antonio@voyager:~/PycharmProjects$ ls -l odoo_olamundo/
total 28
drwxrwxr-x 2 antonio antonio 4096 Xuñ 15 12:08 controllers
drwxrwxr-x 2 antonio antonio 4096 Xuñ 15 12:08 demo
-rw-rw-r-- 1 antonio antonio 71 Xuñ 15 12:08 __init__.py
-rw-rw-r-- 1 antonio antonio 919 Xuñ 15 12:08 __manifest__.py
drwxrwxr-x 2 antonio antonio 4096 Xuñ 15 12:08 models
drwxrwxr-x 2 antonio antonio 4096 Xuñ 15 12:08 security
drwxrwxr-x 2 antonio antonio 4096 Xuñ 15 12:08 views
antonio@voyager:~/PycharmProjects$

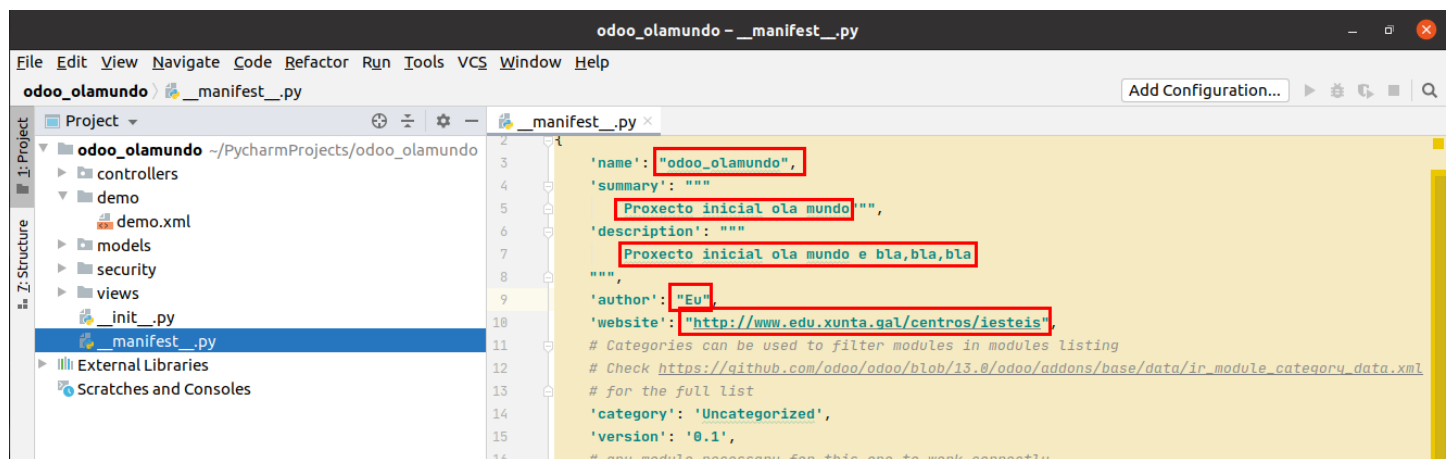
```



## Arquivo \_\_manifest\_\_.py

Como xa comentamos, imos programar en **python**, por tanto o primeiro arquivo que imos ver é `__manifest__.py`.

Onde entre outras cousas podemos configurar a información do noso módulo.



```

17     'depends': ['base'],
18     # always loaded
19     'data': [
20         # 'security/ir.model.access.csv',
21         'views/views.xml',
22         'views/templates.xml',
23     ],
24     # only loaded in demonstration mode
25     'demo': [
26         'demo/demo.xml',
27     ],
28 ]

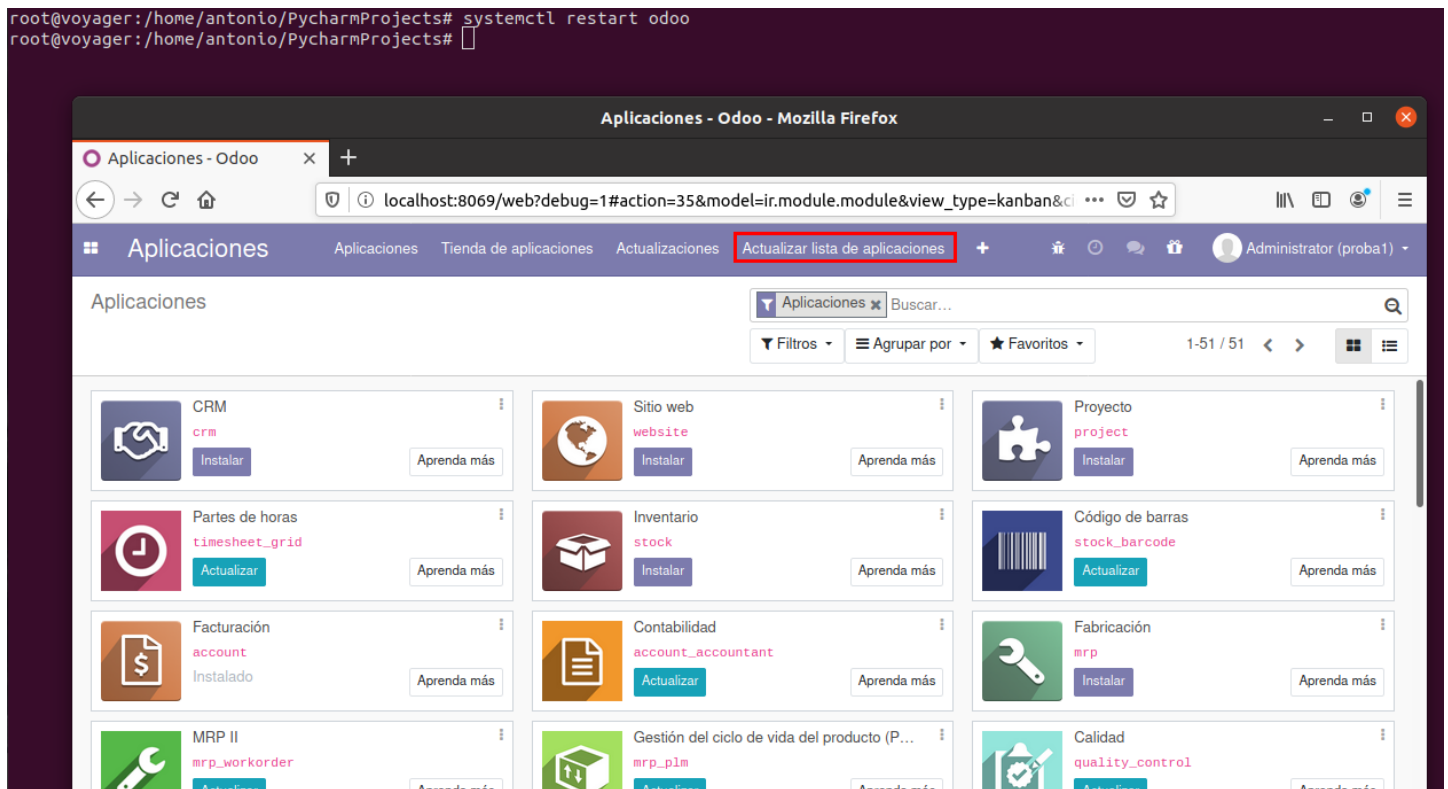
```

author

Despois de gardar os cambios realizados imos ver o noso módulo dende Odoo.

Para elo:

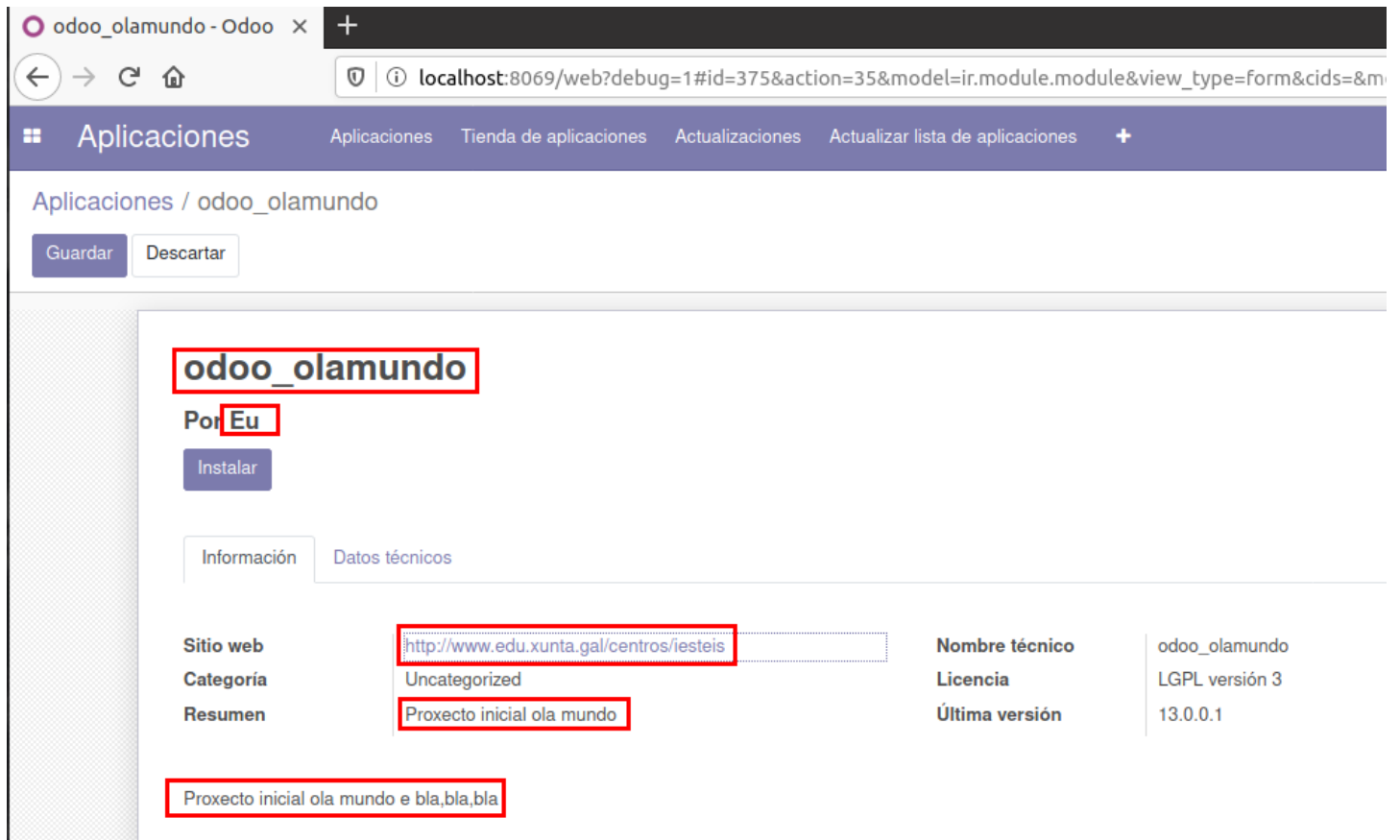
- Reiniciamos o servizo de Odoo
- Activamos "modo desarrollador"
- E no menú "Aplicaciones" pulsamos en "Actualizar lista de aplicaciones"



Despois buscamos o noso módulo por exemplo coa palabra "mundo":

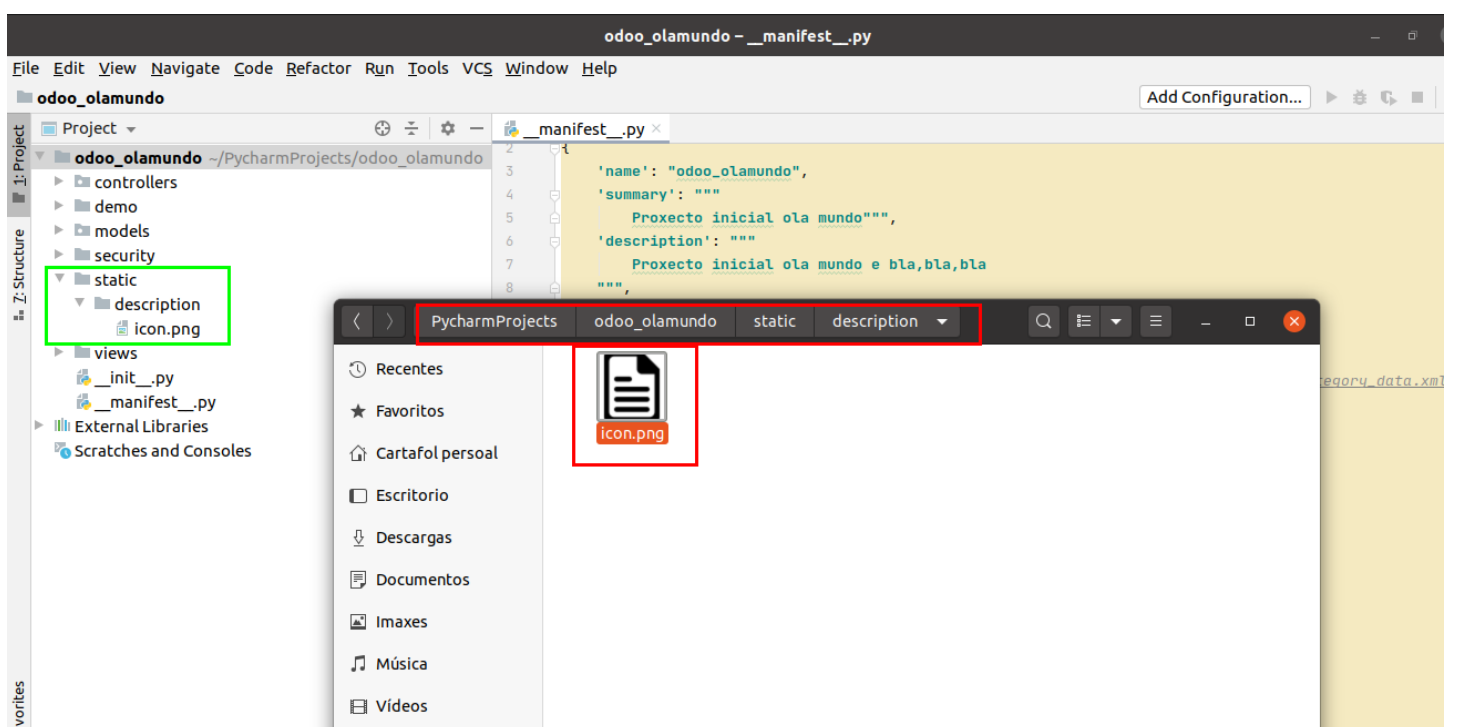


Se pulsamos no menú dos 3 puntitos podemos ver a información do noso módulo:



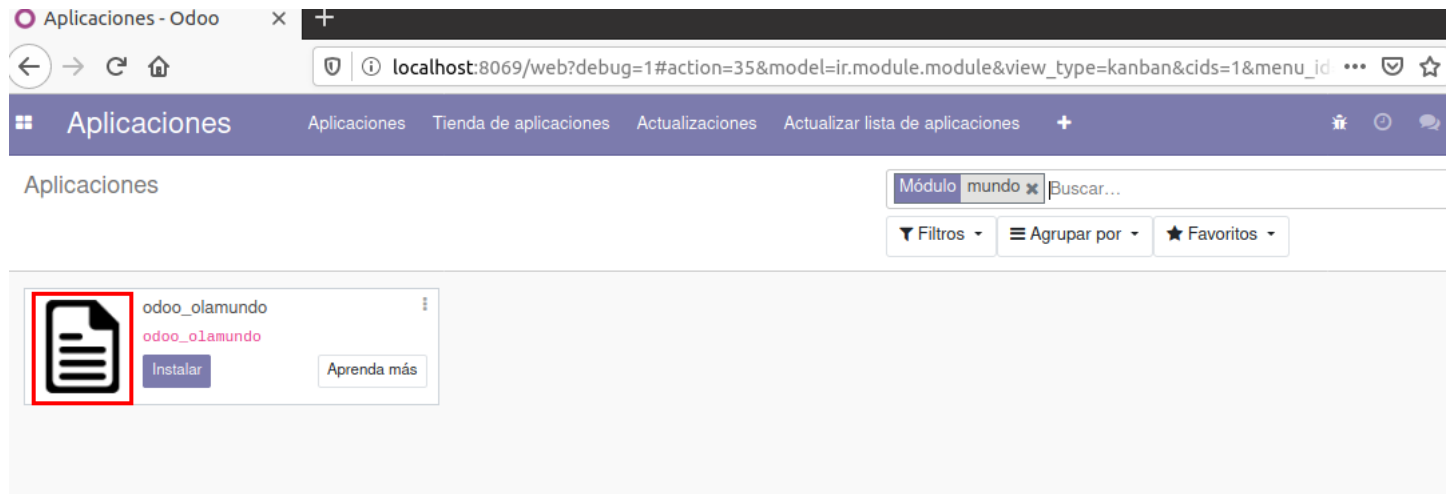
Se queremos cambiar o icono por defecto temos que:

- crear unha ruta **static/description** no noso proxecto e engadir o icono co nome **icon.png**



Despois:

- reiniciar o servizo odoo
- Activar "modo desarrollador"
- E no menú "Aplicaciones" pulsar en "Actualizar lista de aplicaciones"
- E buscar de novo "mundo":



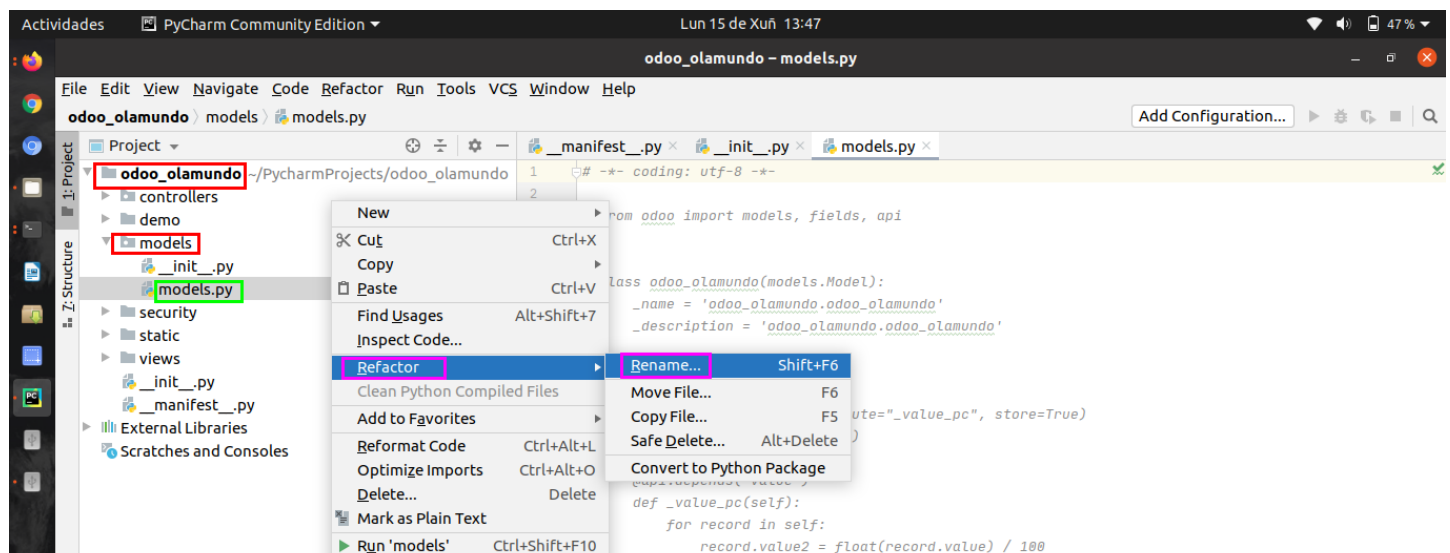
## Instalación do noso módulo

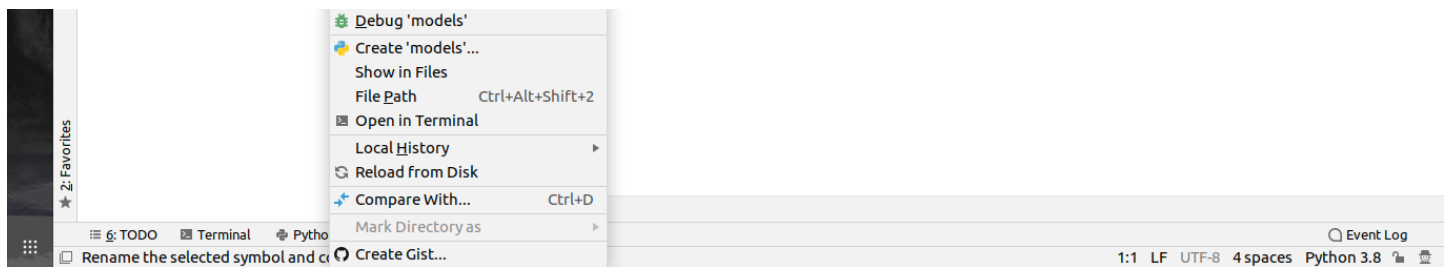
Imos crear o primeiro **modelo** do noso módulo.

Os modelos definímoslos con **python**, mediante as clases (**class**) podemos ter un arquivo **py** con todas as clases pero queda moito mais claro se temos un arquivo co nome de cada clase.

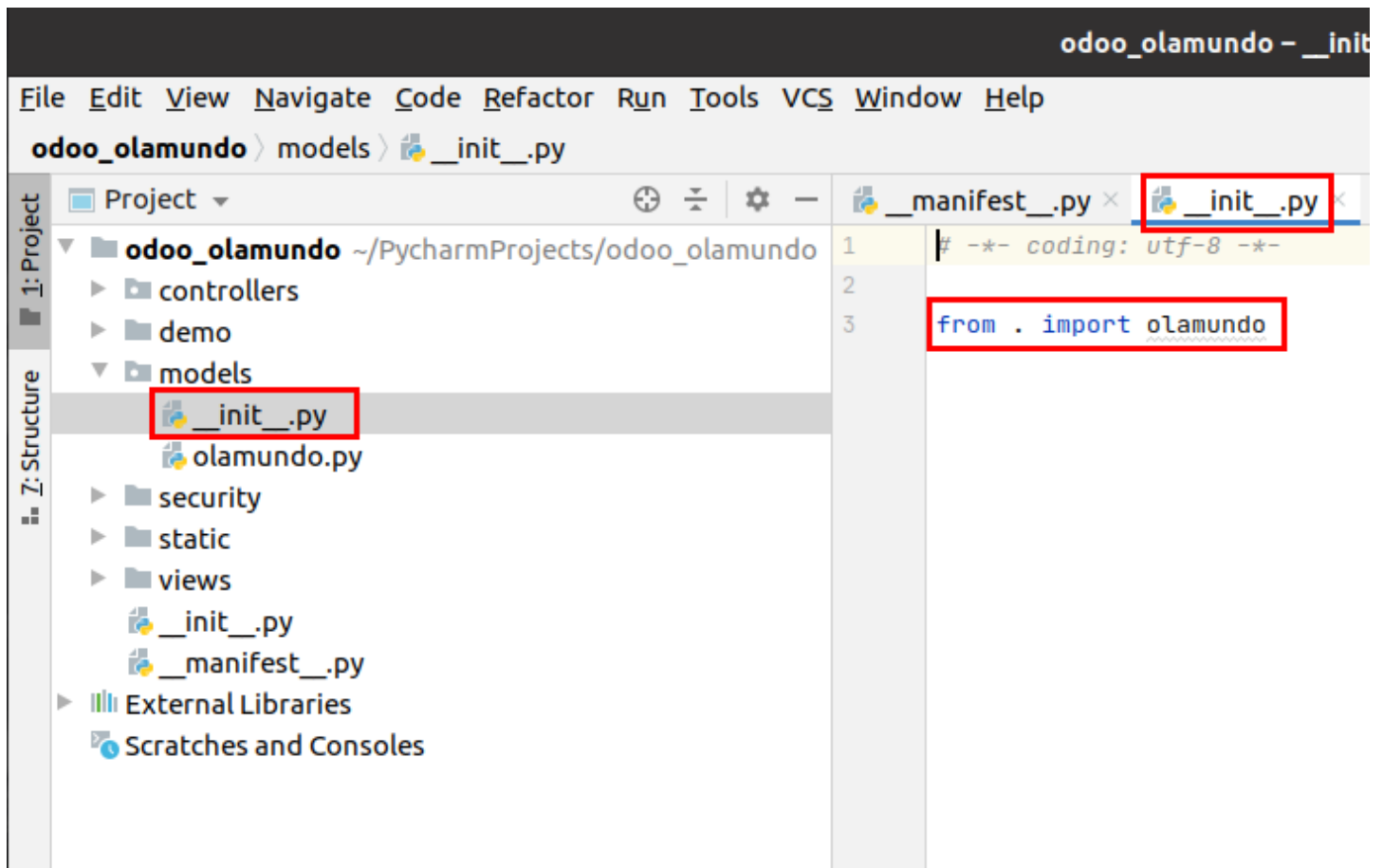
Estos arquivos almacénanse no cartafol **models** do noso proxecto (módulo).

Por tanto imos **renomear** o arquivo **models.py** como **olamundo.py**:





Os arquivos **py** que usamos para definir modelos estarán no cartafol **models** e teñen que estar referenciados no arquivo **\_\_init\_\_.py** que tamén está no mesmo cartafol:



Despois de renomear o arquivo podemos definir o modelo (clase en python):

- Referenciamos as librerías que imos usar
- Definimos a **class** olamundo
- Con **\_name** (azul) determinamos o nome que terá a táboa que Odoo creará na Base de Datos PostgreSQL. No noso caso **odoo\_olamundo\_olamundo**
  - Poñemos **odoo\_olamundo.olamundo**.
    - Onde **odoo\_olamundo** é o nome do **módulo**
    - **.** (separador un punto)
    - **olamundo** é o nome da **clase**
  - A táboa que Odoo creará chamarase **odoo\_olamundo\_olamundo** (Onde se substitue o punto separador entre o módulo e a clase por un guión baixo)

• Mediante **name** creamos o primeiro campo da nosa táboa

• mediante **name** creamos o primeiro campo da nosa táboa

- Que será de tipo **char**
- E en pantalla aparecerá co **literal** "Ola Mundo:"

The screenshot shows the PyCharm IDE with the file `olamundo.py` open. The code defines a new Odoo model class `olamundo` that inherits from `models.Model`. The class has the following attributes and methods:

```

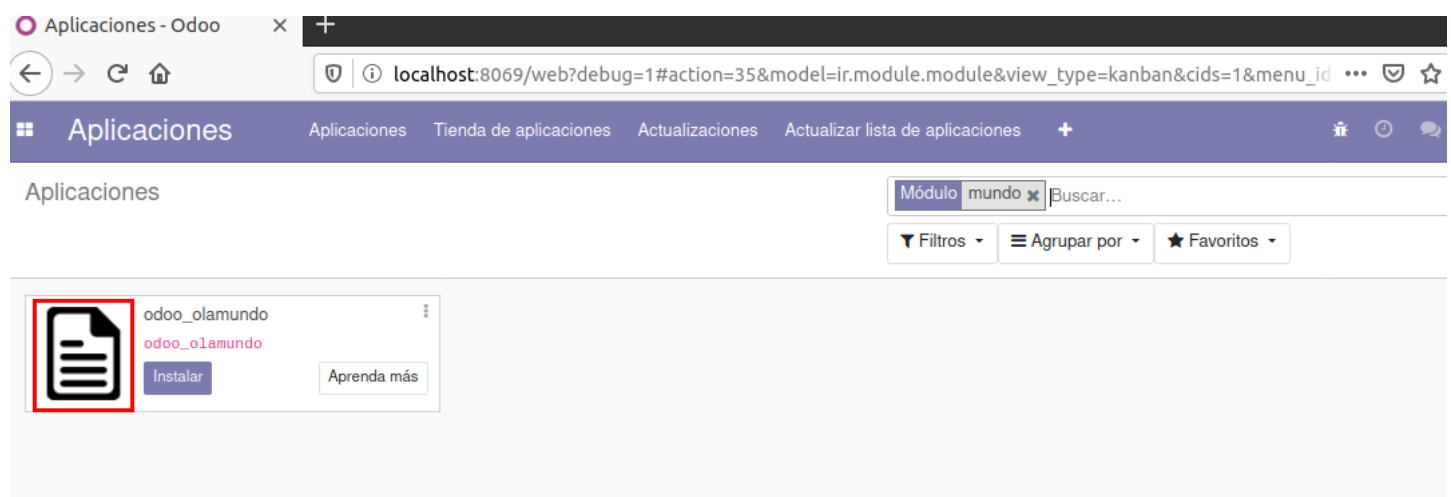
1  # -*- coding: utf-8 -*-
2  from odoo import models, fields, api
3
4
5  class olamundo(models.Model):
6      _name = 'odoo_olamundo.olamundo'
7      _description = 'Exemplo olamundo'
8
9      name = fields.Char(string="Ola Mundo:")
10
11      # value = fields.Integer()
12      # value2 = fields.Float(compute="_value_pc", store=True)
13      # description = fields.Text()
14      # @api.depends('value')
15      # def _value_pc(self):
16      #     for record in self:
17      #         record.value2 = float(record.value) / 100
18

```

In the project structure on the left, the file `olamundo.py` is highlighted under the `models` directory.

Imos **instalar** o noso módulo, para elo, como fixemos cambios no código python, necesitamos **reiniciar o servizo de Odoo** (`systemctl restart odoo`)

Despois instalamos:

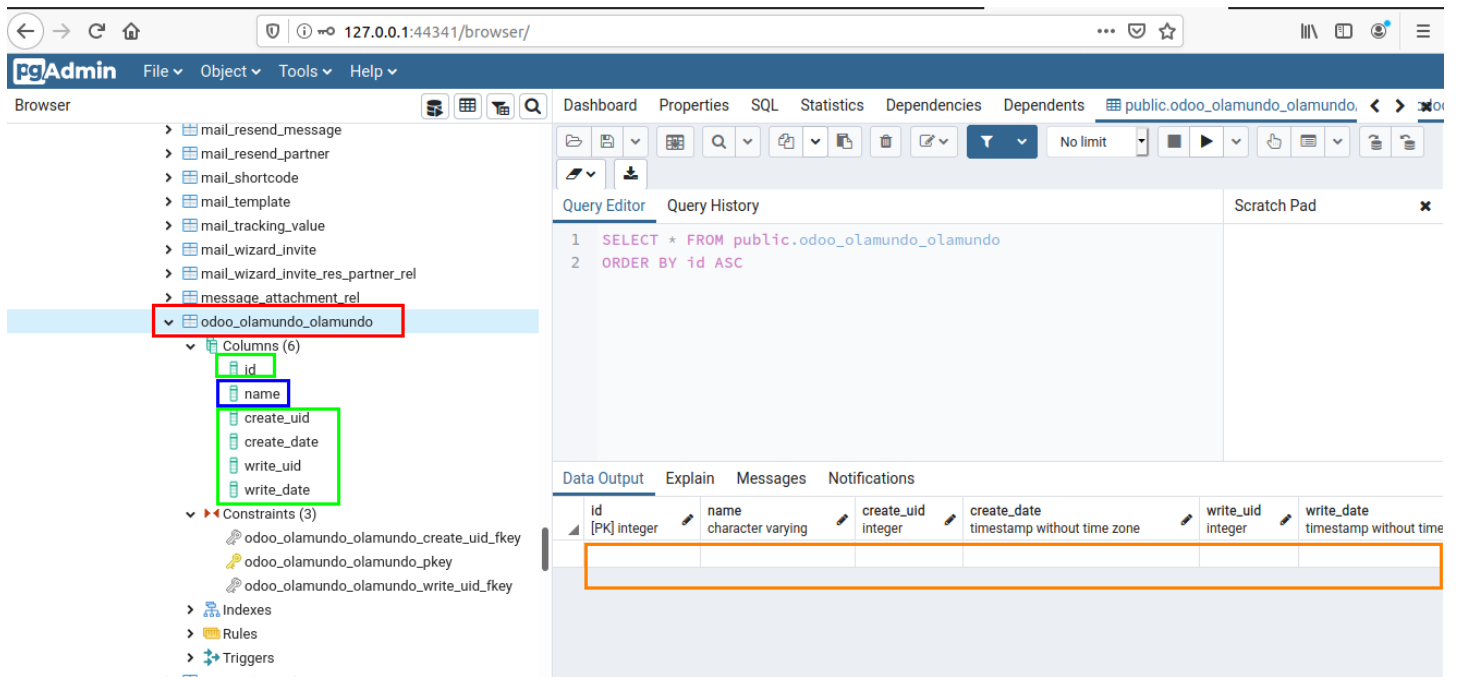


E se non temos ningún erro, podemos comprobar con **pgadmin**:

- A táboa **odoo\_olamundo\_olamundo** que Odoo creou na Base de Datos **proba1**
- Sempre que Odoo crea unha táboa, ésta terá:
  - Os campos que nós definimos. No noso caso **name** (azul)
  - 5 campos (verde): `id`, `create`, `uid`, `create_date`, `write_date`

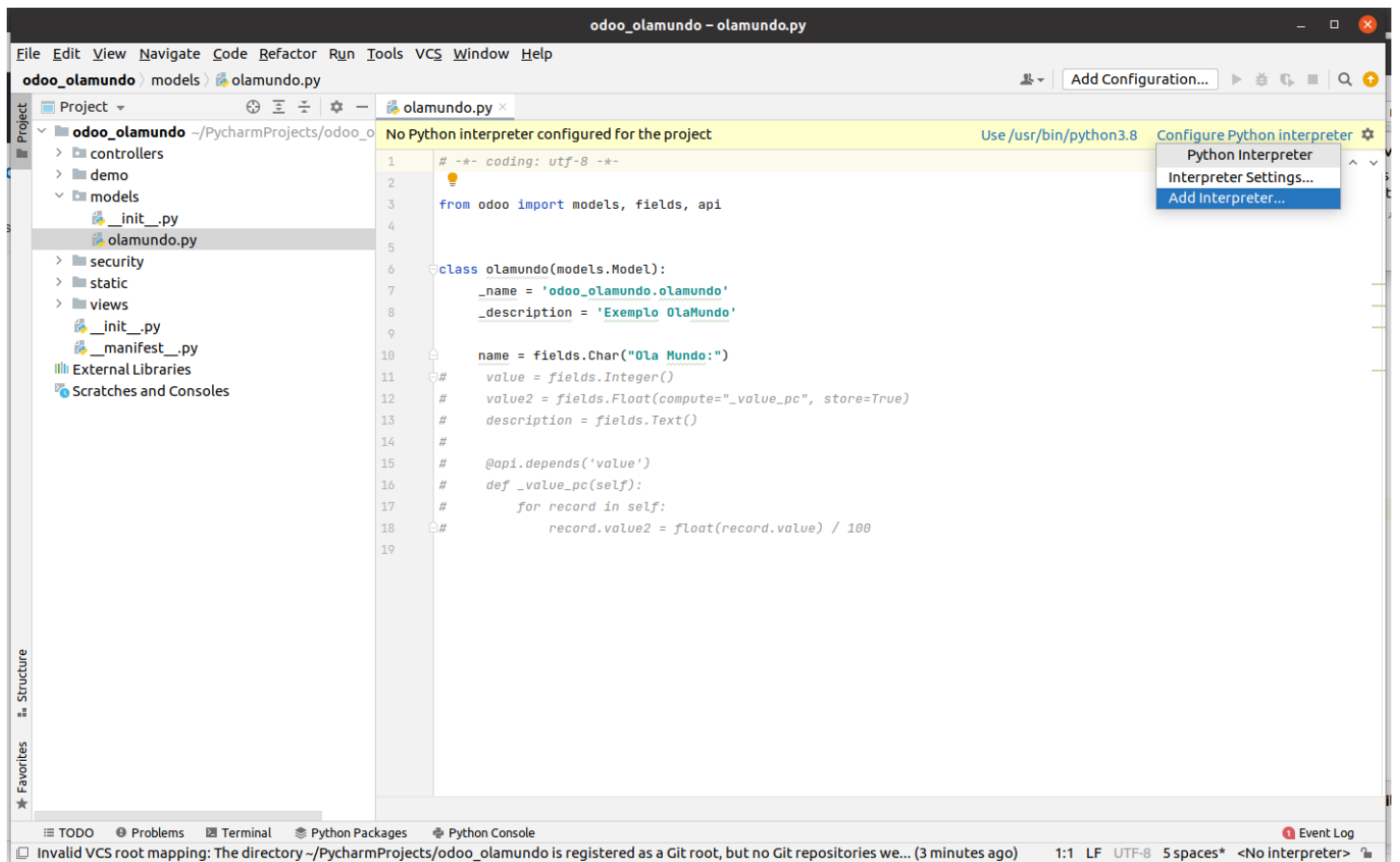


- Vemos que por agora non temos ningún rexistro gravado na táboa (marrón)



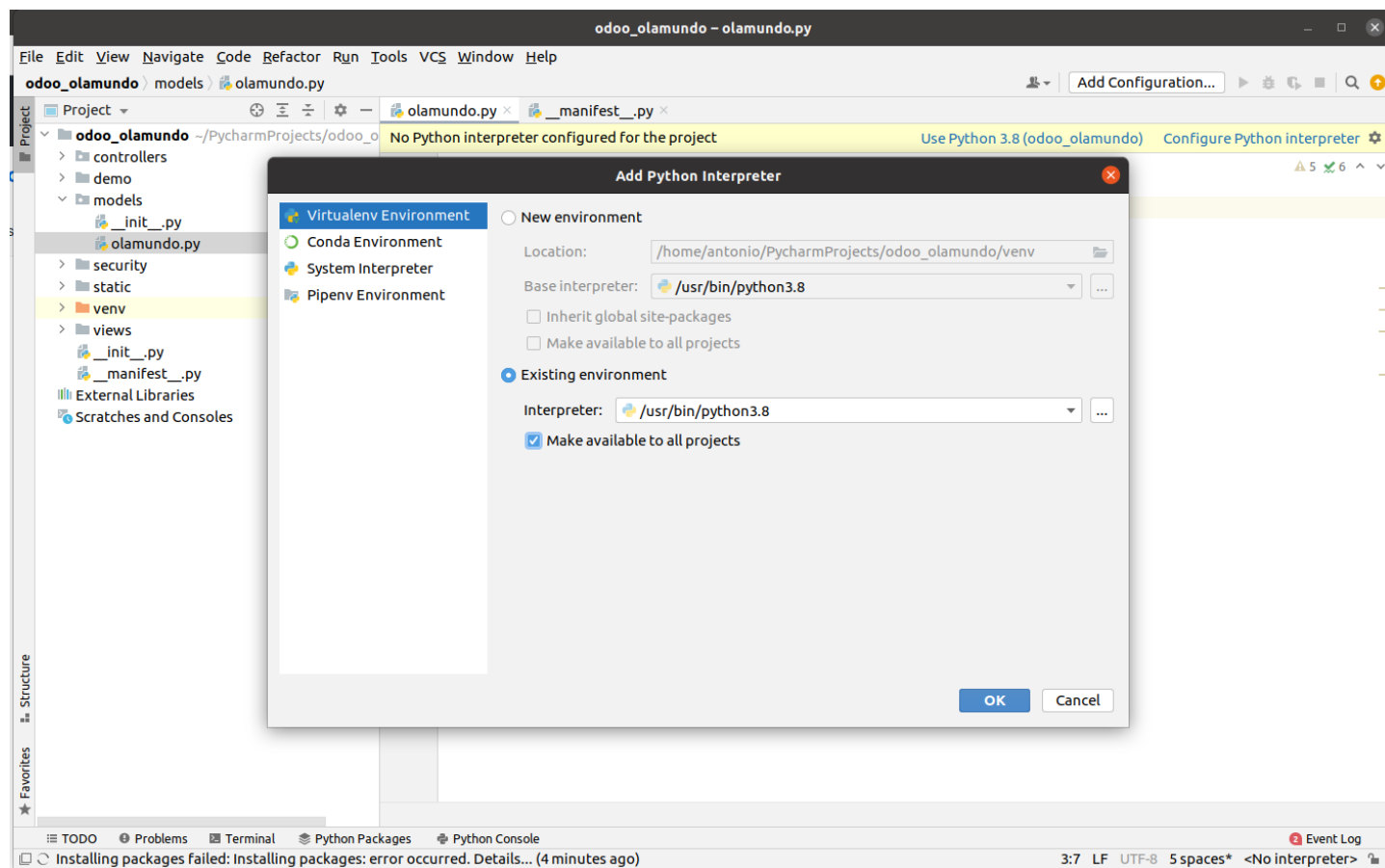
## Configurar o Python Interpreter

Imos engadir o interprete Python. Clickamos en "Add Interpreter"

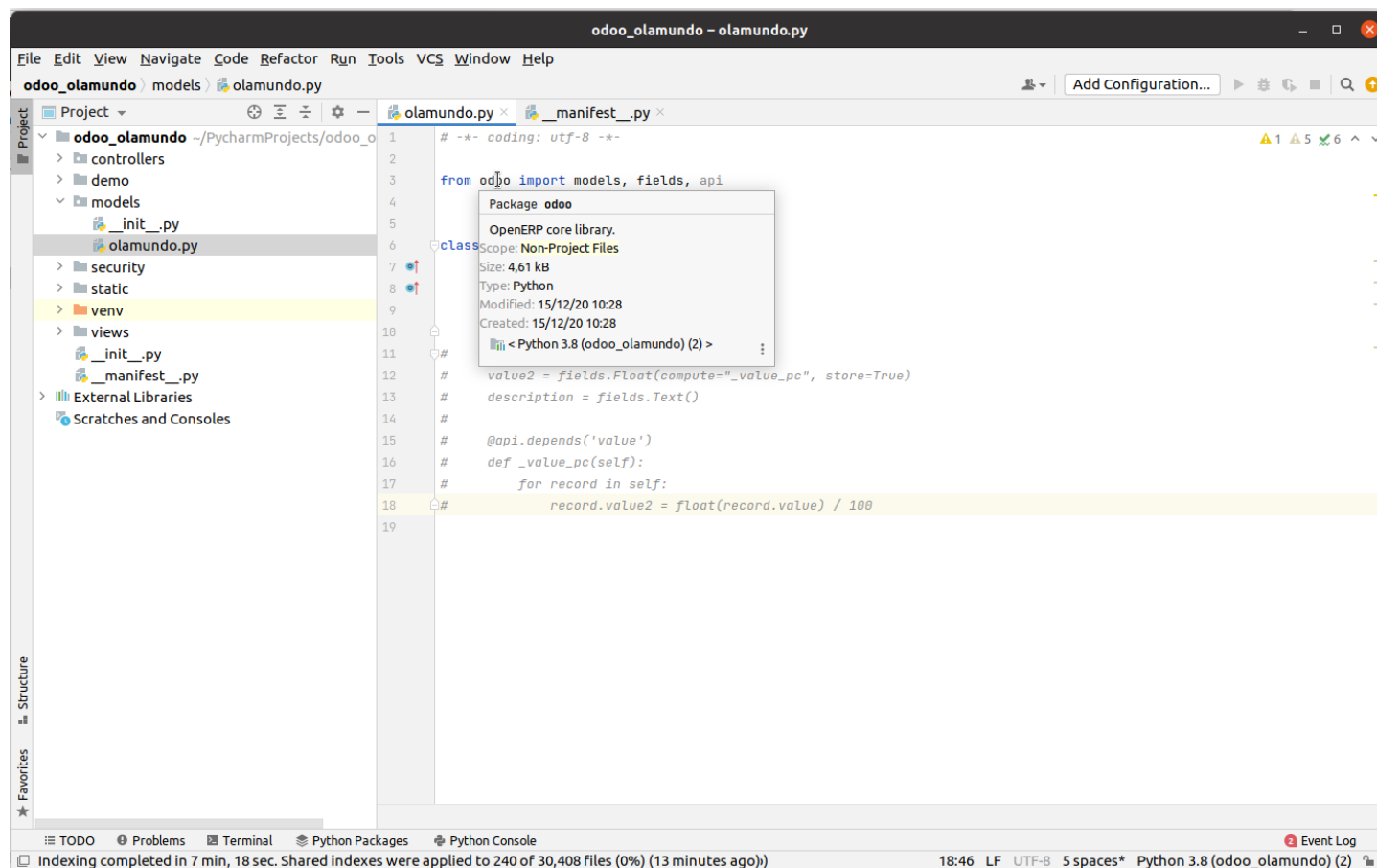




Marcamos en "Existing environment e seleccionamos a ruta /usr/bin/python3.8



e despois de que actualize os índices poderemos ver o interprete configurado:



Obra publicada con Licencia Creative Commons Reconocimiento Compartir igual 4.0