

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	2ª Evaluación
<b>P5</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Albano Díez de Paulino</b>	<b>Fecha</b>	21/02/2024

Diseñar y crear un módulo de Odoo que nos permita gestionar un aspecto empresarial.

El módulo deberá contener, al menos, dos modelos. Los modelos deberán estar relacionados entre sí y cada uno de ellos tendrá, al menos, cuatro atributos. En la documentación deberás incluir una descripción del módulo y un diagrama UML que permitan la comprensión del mismo.

Se crearán vistas (árbol, formulario), menús y acciones que posibiliten mantener las entidades.

Deberás incluir también una demostración de uso del módulo diseñado, instalando y comprobando el funcionamiento de cada elemento.

## Normas

Crea la documentación, que muestre todos los pasos realizados incluyendo capturas de pantalla y la explicación de la realización de los pasos. Entrega, también, la carpeta con la aplicación.

**Todas las capturas de pantalla, se realizarán a pantalla completa. (No recortar)**

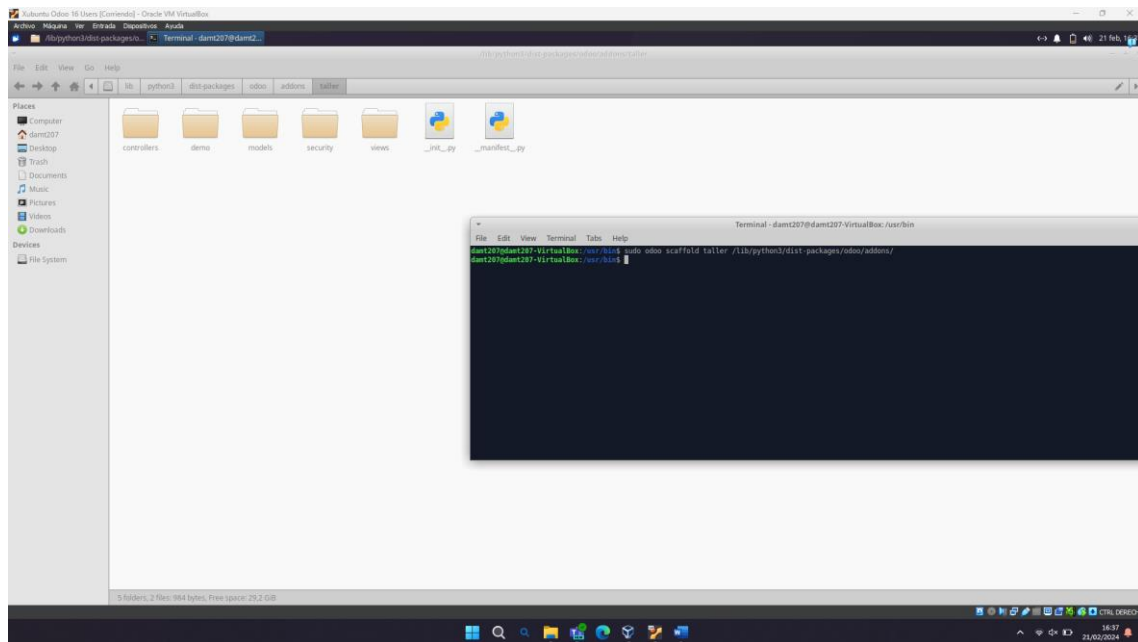
El documento a entregar se llamará: apellido1\_nombre\_p5.pdf

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>2ª Evaluación</b>
<b>P5</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Albano Díez de Paulino</b>	<b>Fecha</b>	<b>21/02/2024</b>

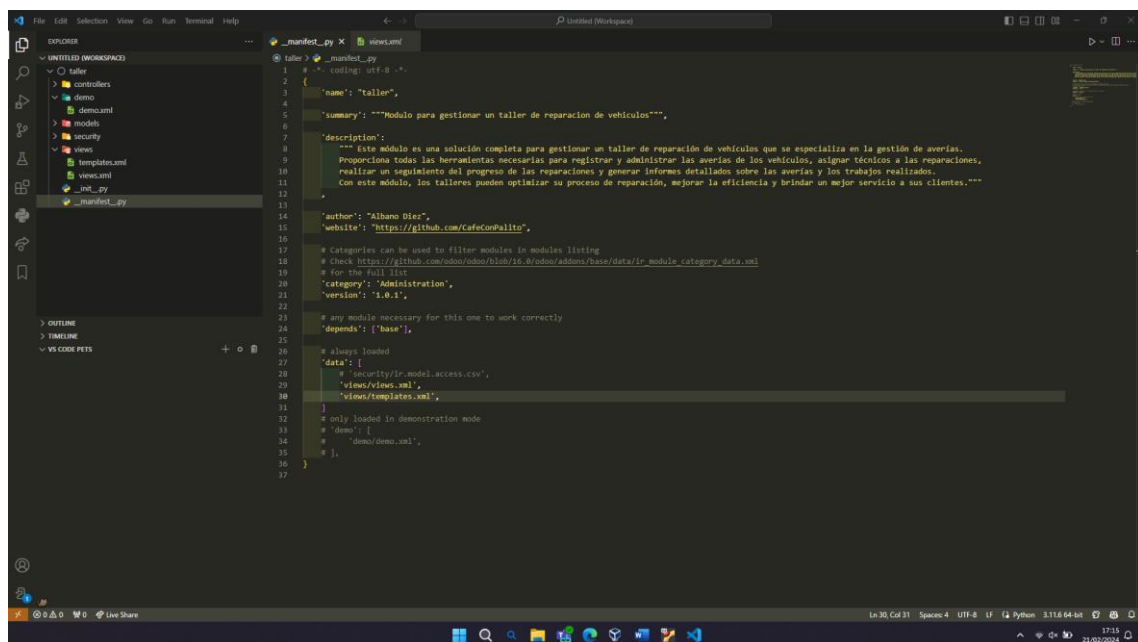
El primer paso para crear el módulo personalizado, en mi caso va a ser un taller que tenga coches y estos a su vez estas averías, para ello hay que ejecutar el siguiente comando para crear una plantilla que lo modificare.

Comando: `odoo scaffold taller /lib/python3/dist-packages/odoo/addons/`

Si estas en Ubuntu hay que ejecutar el comando desde la siguiente ruta `/usr/bin`



Teniendo la plantilla el primer paso hay que modificar el `__manifest__.py`



<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	2ª Evaluación
<b>P5</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Albano Díez de Paulino</b>	<b>Fecha</b>	21/02/2024

Tras modificar el `__manifest__.py` hay que crear los modelos en el fichero `models.py`, en mi caso son los siguientes:

Modelo Coche:

- Marca
- Modelo
- Matricula
- Color

```

1 # -*- coding: utf-8 -*-
2
3 from odoo import models, fields, api
4
5 class Coche(models.Model):
6     _name = "modulo.coche"
7     _description = "modulo.coche"
8
9
10     marca = fields.Char(
11         string="Marca",
12         store=True,
13         readonly=False,
14         required=True,
15         translate=False)
16
17     modelo = fields.Char(
18         string="Modelo",
19         store=True,
20         readonly=False,
21         required=True,
22         translate=False)
23
24     matricula = fields.Char(
25         string="Matricula",
26         store=True,
27         readonly=False,
28         required=True,
29         translate=False)
30
31     color = fields.Selection([('1','Rojo'),('2','Azul'),('3','Verde')],
32                             string="Color",
33                             store=True,
34                             readonly=False,
35                             required=True,
36                             translate=False)
37
38     # value = fields.Integer()
39     # value2 = fields.Float(compute='_value_pc', store=True)
40     # description = fields.Text()
41
42     # @api.depends('value')
43     # def _value_pc(self):
44     #     for record in self:
45     #         record.value2 = float(record.value) / 100
46
47
48

```

Modelo Averias:

- Importe
- Coche
- Tipo
- Fecha

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>2ª Evaluación</b>
<b>P5</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Albano Díez de Paulino</b>	<b>Fecha</b>	<b>21/02/2024</b>

```

41 # description = fields.Text()
42
43 # api.depends('value')
44 # def value_pc(self):
45 #     for record in self:
46 #         record.value2 = float(record.value) / 100
47
48
49 class averia(models.Model):
50     name = fields.Char('Nombre')
51     description = fields.Text()
52
53     coche = fields.Many2one('modulo.coche',
54                             string='coche',
55                             store=True,
56                             readonly=False,
57                             required=True,
58                             translate=False)
59
60     fecha = fields.Date(string='Fecha',
61                        default=fields.Date.now,
62                        store=True,
63                        readonly=False,
64                        required=True,
65                        translate=False)
66
67     precio = fields.Float(
68         string='Precio',
69         store=True,
70         readonly=False,
71         required=True,
72         translate=False)
73
74     tipo = fields.Char(
75         string='tipo',
76         store=True,
77         readonly=False,
78         required=True,
79         translate=False)
80
81
82
83
84

```

El siguiente paso es crear las vistas, las acciones para lanzar las vistas y el menú para poder acceder, si se hiciese desde la interfaz de odoo cada cosa se hace por separado, pero como se está haciendo por código se hace todo junto en el fichero views.xml

Lo primero que he creado ha sido son las vistas de tipo tree de los dos modelos.

```

1 <?xml version="1.0"?>
2 <coche>
3     <!-- explicit list view definition -->
4
5     <record model="ir.ui.view" id="coche.list">
6         <field name="name">Lista de coches</field>
7         <field name="model">modulo.coche</field>
8         <field name="arch" type="xml">
9             <tree>
10                 <field name="marca"/>
11                 <field name="modelo"/>
12                 <field name="matricula"/>
13                 <field name="color"/>
14             </tree>
15         </field>
16     </record>
17
18     <record model="ir.ui.view" id="averia.list">
19         <field name="name">Lista de averias</field>
20         <field name="model">modulo.averia</field>
21         <field name="arch" type="xml">
22             <tree>
23                 <field name="coche"/>
24                 <field name="fecha"/>
25                 <field name="precio"/>
26                 <field name="tipo"/>
27             </tree>
28         </field>
29     </record>
30
31     <record model="ir.ui.view" id="coche.form">
32         <field name="name">Inserta un coche</field>
33         <field name="model">modulo.coche</field>
34         <field name="arch" type="xml">
35             <form>
36                 <group name="coche">
37                     <div>Matricula: <field name="matricula"/></div>
38                     <div>
39                         <field name="modelo" label="Modelo:" />
40                         <field name="marca" label="Marca:" />
41                         <field name="color" label="Color:" />
42                     </div>
43                 </group>
44             </form>
45         </field>
46     </record>
47
48     <record model="ir.ui.view" id="averia.form">

```

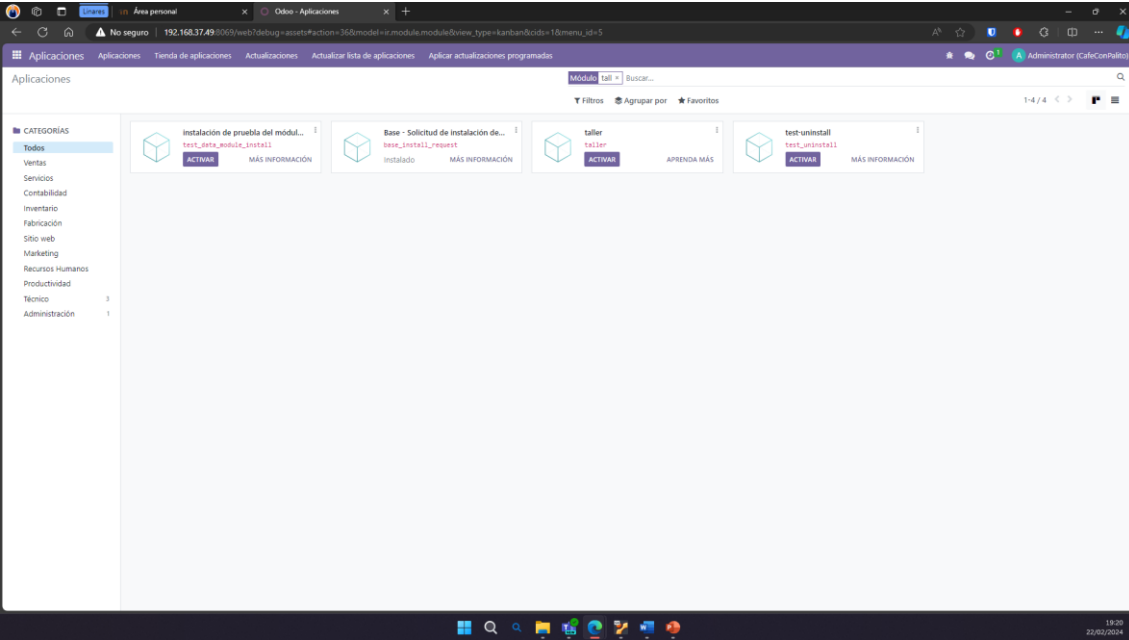
Y luego las vistas de tipo formulario



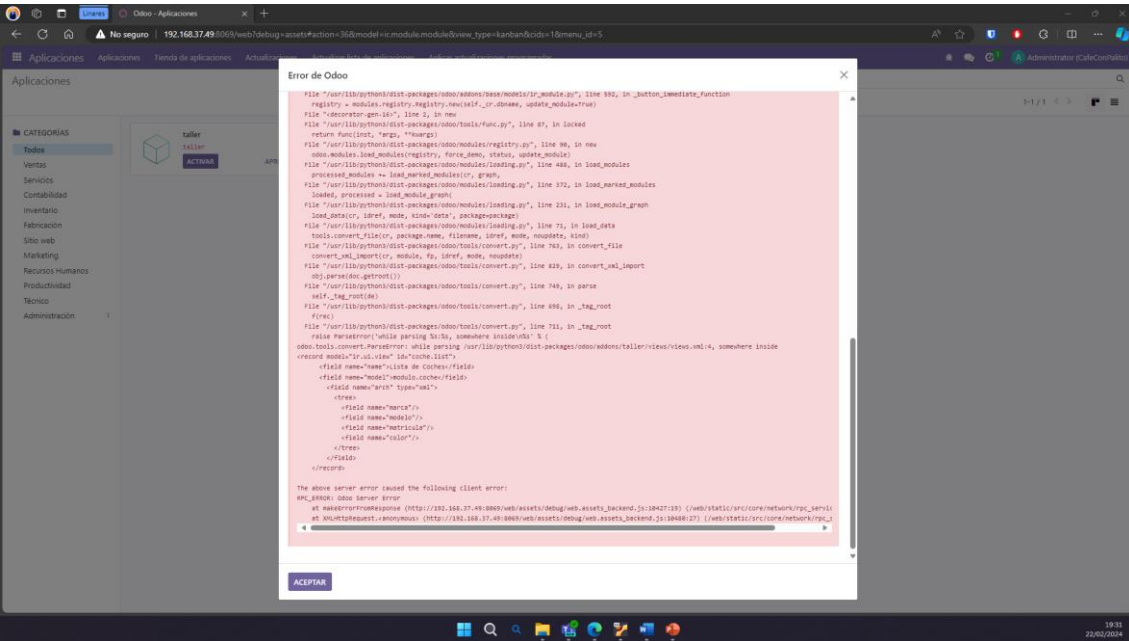


<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>2ª Evaluación</b>
<b>P5</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Albano Díez de Paulino</b>	<b>Fecha</b>	<b>21/02/2024</b>

Ya debería salir el módulo personalizado y se puede activar.



Por desgracia no ha ido todo bien he ha dado este error al intentar instalarlo.



Después de buscar durante bastante tiempo, junto a mis compañeros de clase que tenían el mismo problema, encontramos que el fallo era que no conseguía crear la vistas correctamente porque no ponía el id bien y hay que asociarlo al nombre del módulo que estas creado.



<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	2ª Evaluación
<b>P5</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Albano Díez de Paulino</b>	<b>Fecha</b>	21/02/2024

```

1 <{% csrf_token %}>
2 <div>
3     <!-- explicit list view definition -->
4     <record model="ir.ui.view" id="taller.list">
5         <field name="name">Lista de coches</field>
6         <field name="model">modulo.coche</field>
7         <field name="arch" type="xml">
8             <group>
9                 <field name="marca"/>
10                <field name="modelo"/>
11                <field name="matricula"/>
12                <field name="color"/>
13            </group>
14        </field>
15    </record>
16
17    <record model="ir.ui.view" id="taller.list">
18        <field name="name">Lista de averías</field>
19        <field name="model">modulo.averia</field>
20        <field name="arch" type="xml">
21            <group>
22                <field name="coche"/>
23                <field name="fecha"/>
24                <field name="precio"/>
25                <field name="tipo"/>
26            </group>
27        </field>
28    </record>
29
30    <record model="ir.ui.view" id="taller.form">
31        <field name="name">Inserta un coche</field>
32        <field name="model">modulo.coche</field>
33        <field name="arch" type="xml">
34            <group>
35                <field name="coche">
36                    <div>Matricula: <field name="matricula"/></div>
37                    <div>
38                        <field name="modelo" label="Modelo:" />
39                        <field name="marca" label="Marca:" />
40                        <field name="color" label="Color:" />
41                    </div>
42                </field>
43            </group>
44        </field>
45    </record>
46
47    <record model="ir.ui.view" id="taller.form">
48        <field name="name">Inserta una averia</field>

```

Además, los modelos tenían un error en modelo de coche, el campo de color que es de tipo selection y hay que declarar el parámetro que quieres pasar.

```

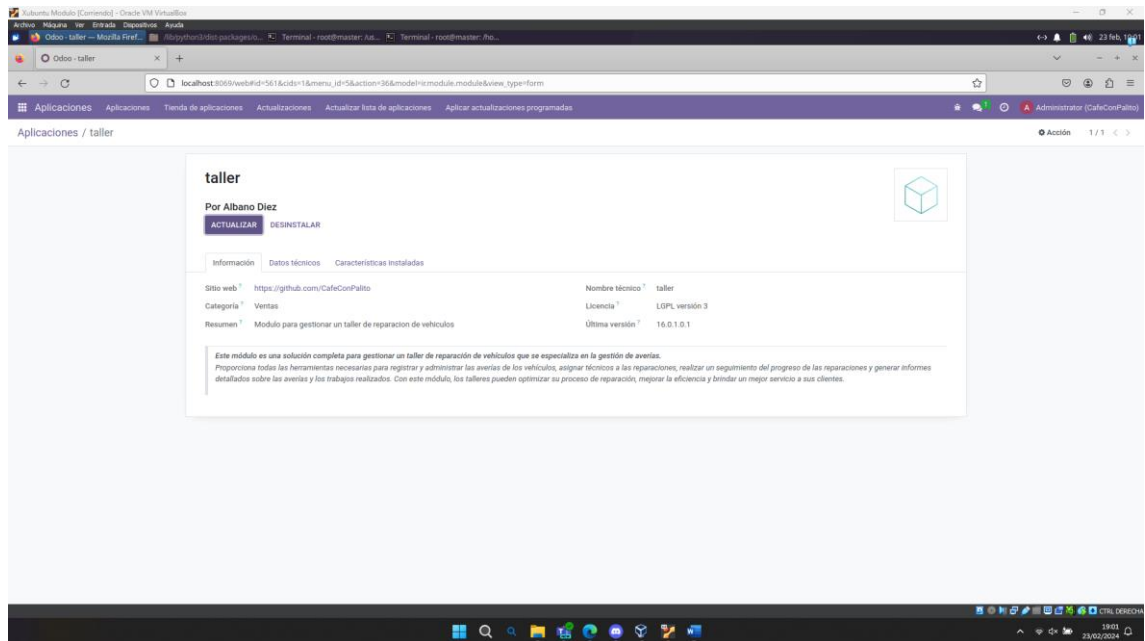
1 # -*- coding: utf-8 -*-
2
3 from django import models, fields, api
4
5 class Coche(models.Model):
6     name = 'modulo.coche'
7     _description = 'modulo.coche'
8
9     marca = fields.CharField(
10         string='Marca',
11         store=True,
12         readonly=False,
13         required=True,
14         translate=False)
15
16     modelo = fields.CharField(
17         string='Modelo',
18         store=True,
19         readonly=False,
20         required=True,
21         translate=False)
22
23     matricula = fields.CharField(
24         string='Matricula',
25         store=True,
26         readonly=False,
27         required=True,
28         translate=False)
29
30     color = fields.Selection(selection=[('1', 'Rojo'), ('2', 'Azul'), ('3', 'Verde')],
31                             string='Color',
32                             store=True,
33                             readonly=False,
34                             required=True,
35                             translate=False)
36
37     # value = fields.IntegerField()
38     # value2 = fields.FloatField(compute='_value_pc', store=True)
39     # description = fields.TextField()
40
41     @api.depends('value')
42     @api.depends('value2')
43     def _value_pc(self):
44         for record in self:
45             record.value2 = float(record.value) / 100
46
47

```

Tras solucionar esos problemas ya he podido cargar el módulo.

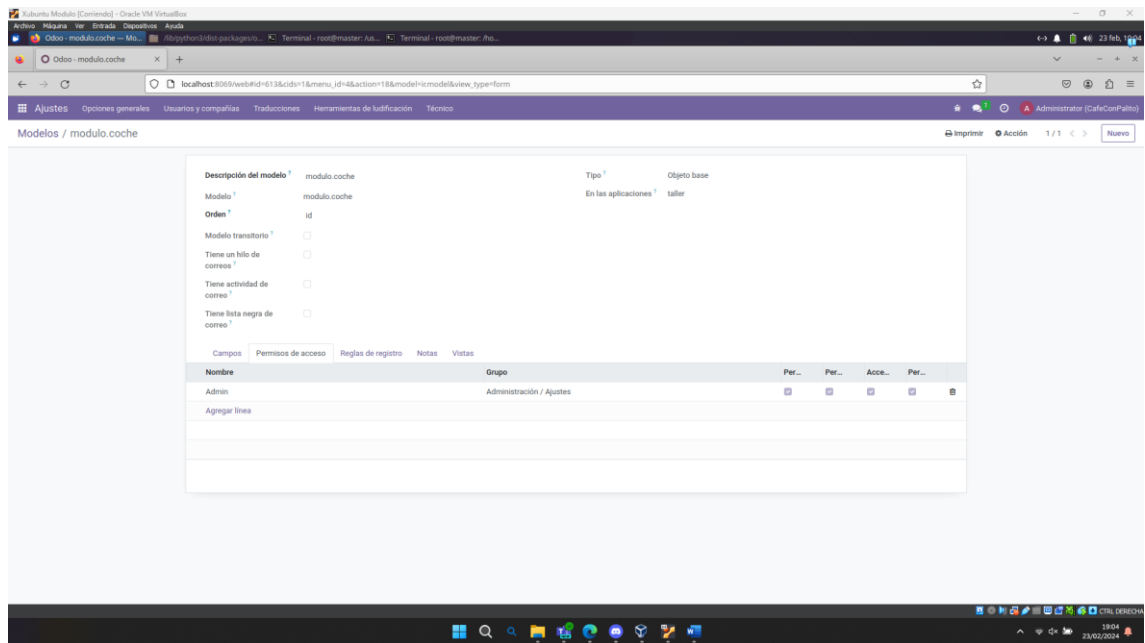


<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>2ª Evaluación</b>
<b>P5</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Albano Díez de Paulino</b>	<b>Fecha</b>	<b>21/02/2024</b>



Pero hay un nuevo problema, que no he cargado los permisos por código por lo que hay que implementarlos, para ello con dársele a los módulos.

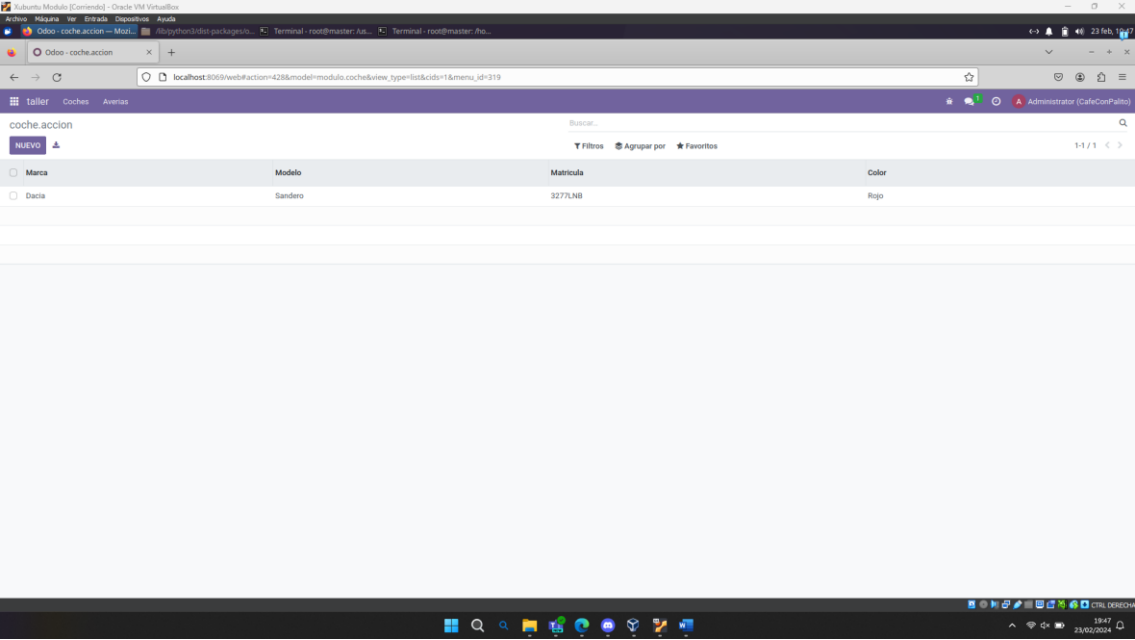
Permisos al módulo coche como ejemplo.



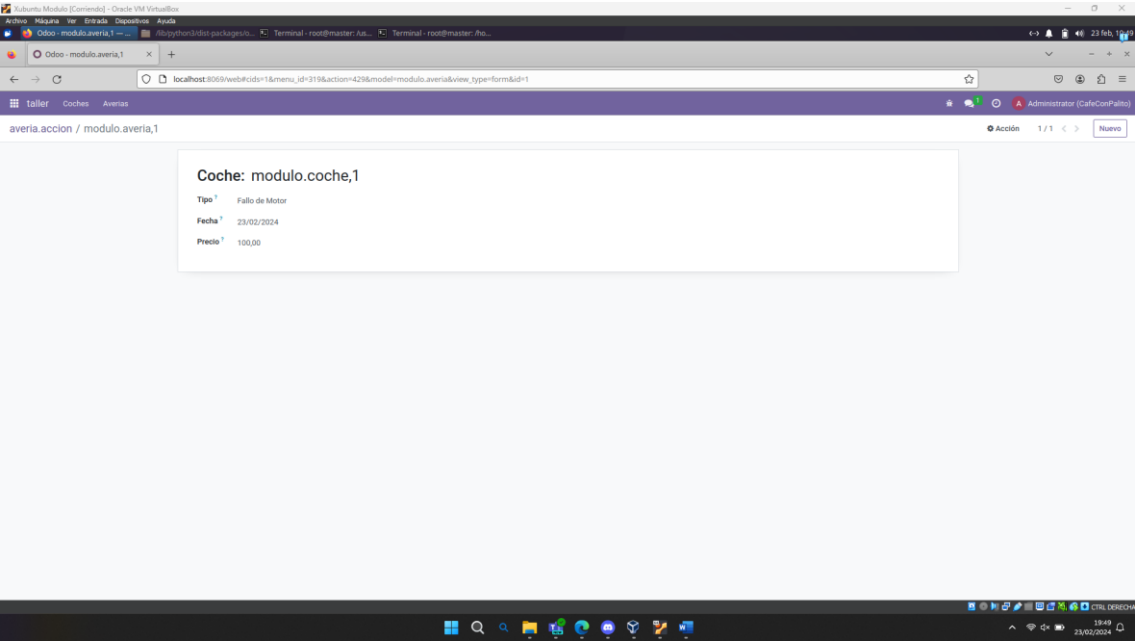
Ahora solo queda hay que probar que el módulo vaya bien.

<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>2ª Evaluación</b>
<b>P5</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Albano Díez de Paulino</b>	<b>Fecha</b>	<b>21/02/2024</b>

Lista de Coches con un coche insertado por mí.

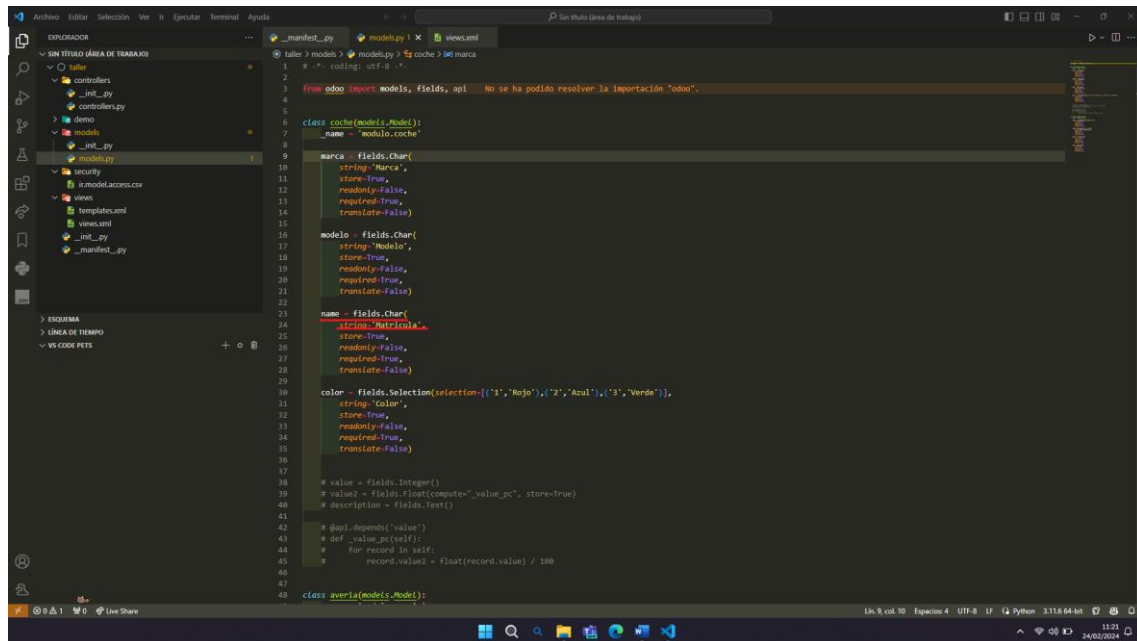


Creando una avería para el coche 1.



<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>2ª Evaluación</b>
<b>P5</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Albano Díez de Paulino</b>	<b>Fecha</b>	<b>21/02/2024</b>

Pero como se puede observar el modelo guarda los coches con un nombre raro, para ello hay que modificar la variable que quieras que guarde en el nombre y ponerla como name, en mi caso quiero que sea la matricula.

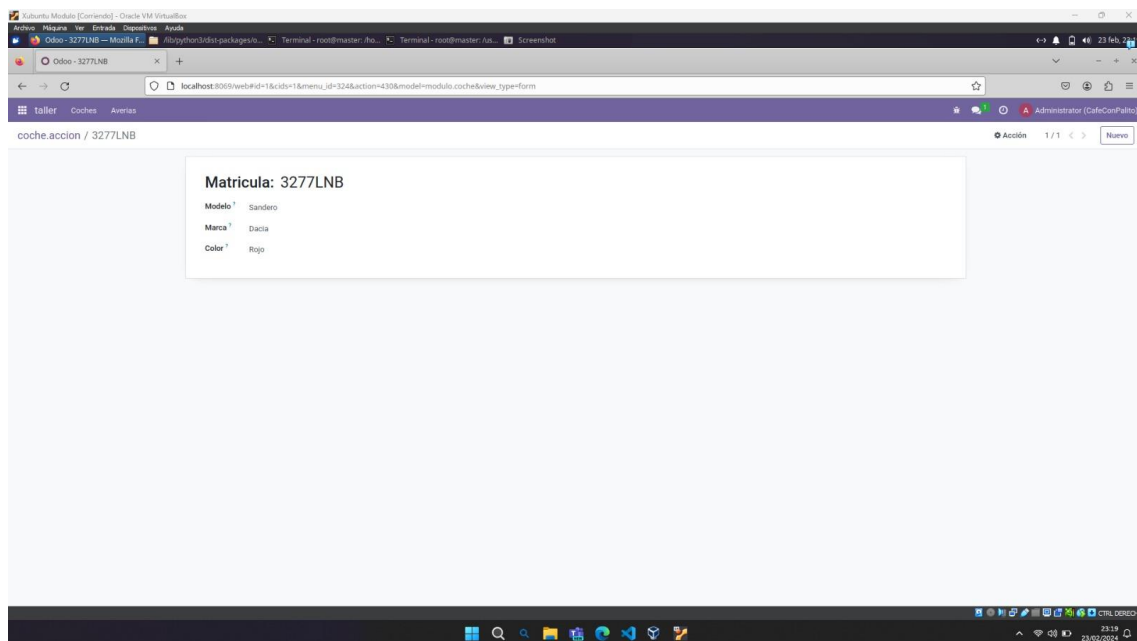


```

1 # coding: utf-8 -*-
2
3 from django import models, fields, api
4
5 class coche(models.Model):
6     name = models.CharField(
7         max_length=100,
8         unique=True,
9         help_text='El nombre del coche',
10        )
11
12     marca = fields.CharField(
13         max_length=100,
14         unique=True,
15         help_text='La marca del coche',
16        )
17
18     modelo = fields.CharField(
19         max_length=100,
20         unique=True,
21         help_text='El modelo del coche',
22        )
23
24     name = fields.CharField(
25         max_length=100,
26         unique=True,
27         help_text='El nombre del coche',
28        )
29
30     color = fields.CharField(
31         max_length=100,
32         unique=True,
33         help_text='El color del coche',
34        )
35
36     # value = fields.IntegerField()
37     # value2 = fields.FloatField(compute="value_pc", store=True)
38     # description = fields.TextField()
39
40     # @api.depends('value')
41     # def _value_pc(self):
42     #     for record in self:
43     #         record.value2 = float(record.value) / 100
44
45 class averia(models.Model):
46

```

Ahora vuelvo a probar el modulo y los coches se guardan con la matricula.



<b>Sistemas de Gestión Empresarial</b>	<b>Ejercicios</b>	<b>2ª Evaluación</b>
<b>P5</b>	<b>Grupo</b>	<b>DAMT2</b>
<b>Nombre y apellidos: Albano Díez de Paulino</b>	<b>Fecha</b>	<b>21/02/2024</b>

Ya solo falta el diagrama UML de las entidades.

