

PASOS PARA LA RESOLUCIÓN

1. Se levanta el Docker.

```
PS C:\Users\Usuario\Desktop\Alicia\2. DAM\4. SGE\SGE_practicas\PR06\pr0606> docker compose up -d
[+] Running 2/2
 ✓ Container db          Started
   0.6s
 ✓ Container odoo_16     Started
   1.0s
PS C:\Users\Usuario\Desktop\Alicia\2. DAM\4. SGE\SGE_practicas\PR06\pr0606> docker exec -it 709ffc017 bash
odoo@709ffc01774a:/$ odoo scaffold taller /mnt/extra-addons/
odoo@709ffc01774a:/$
```

2. Establecimiento de credenciales.

Nota: La base de datos se llama db_taller

Warning, your Odoo database manager is not protected. To secure it, we have generated the following master password for it:

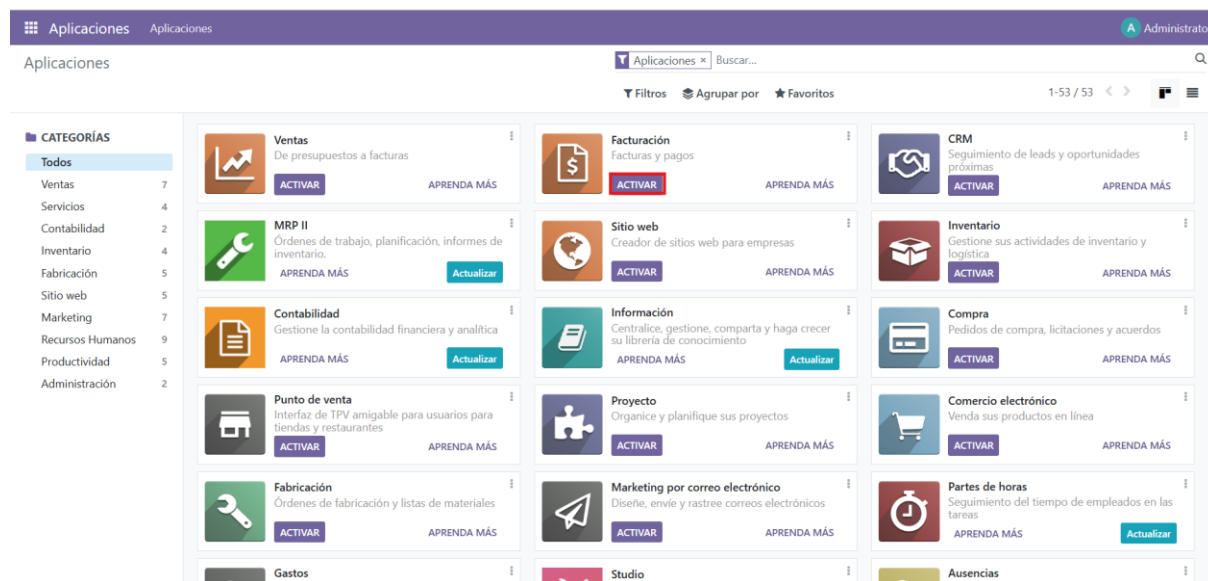
ea9u-2wn2-7fjg

You can change it below but be sure to remember it, it will be asked for future operations on databases.

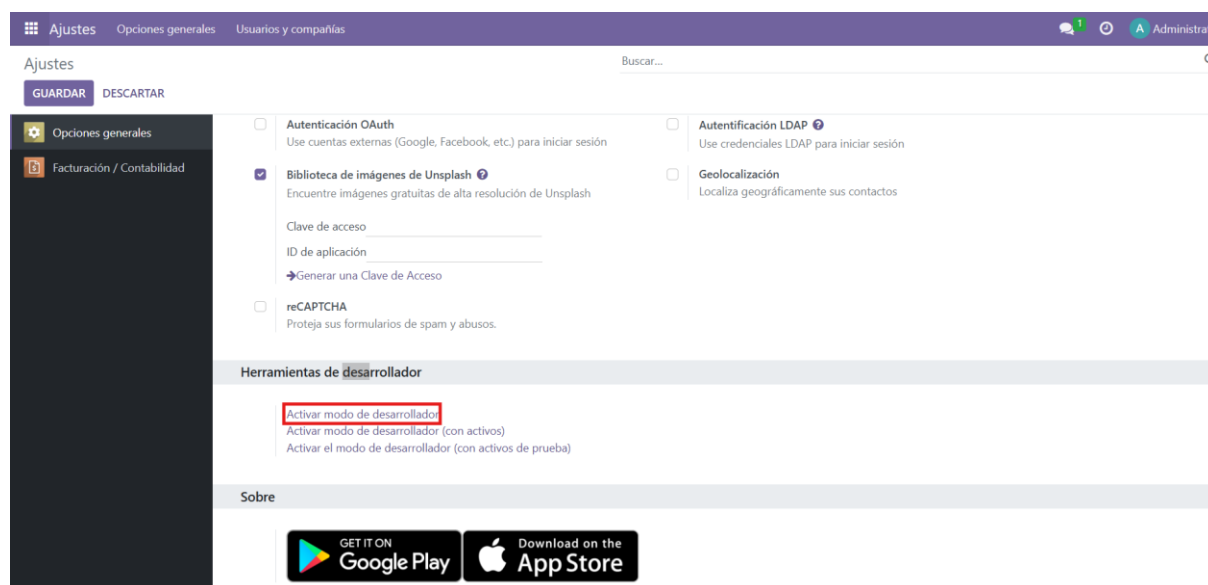
Master Password	<input type="password" value="...."/>
Database Name	<input type="text" value="db_subscription"/>
Email	<input type="text" value="alicia.ibanez.escudero@gmail.com"/>
Password	<input type="password" value="...."/>
Phone number	<input type="text" value="666666666"/>
Language	<input type="text" value="Spanish / Español"/>
Country	<input type="text" value="Spain"/>
Demo data	<input type="checkbox"/>

[Create database](#) [or restore a database](#)

3. Instalación de una aplicación para poder activar el modo de desarrollador.



3. Activación del modo de desarrollados en *Ajustes*.



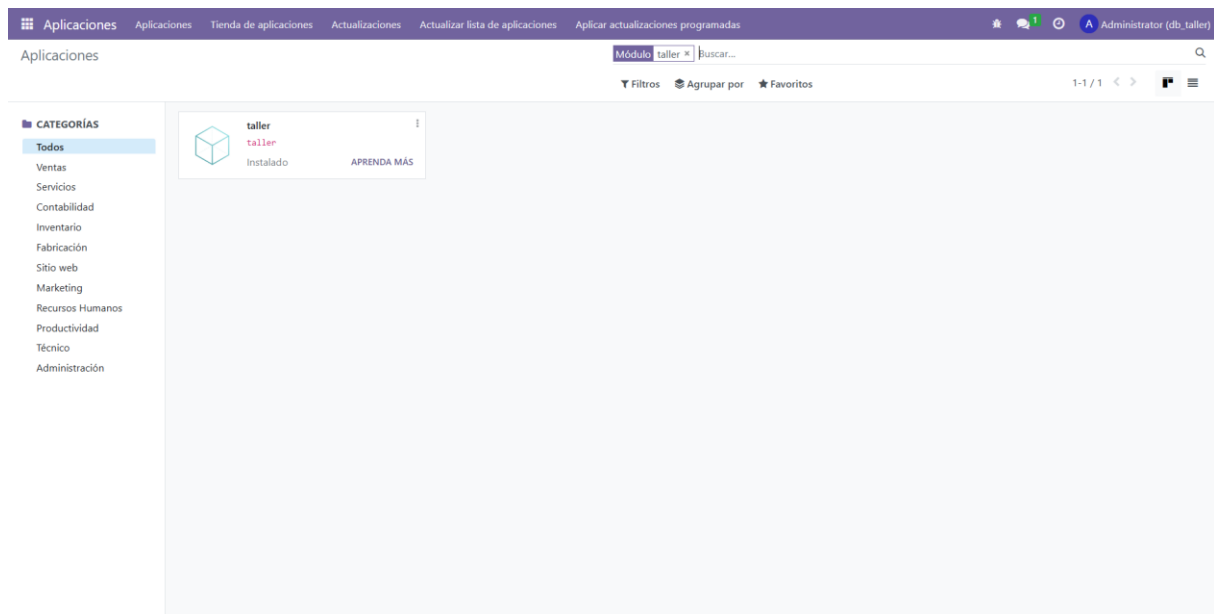
4. Acceso, mediante la línea de comandos, al contenedor Odoo.

```
PS C:\Users\Usuario\Desktop\Alicia\2. DAM\4. SGE\SGE_practicas\PR06\pr0606> docker compose up -d
[+] Running 2/2
 ✓ Container db          Started
    0.6s
 ✓ Container odoo_16     Started
    1.0s
PS C:\Users\Usuario\Desktop\Alicia\2. DAM\4. SGE\SGE_practicas\PR06\pr0606> docker exec -it 709ffc017 bash
odoo@709ffc01774a:/$ odoo scaffold taller /mnt/extra-addons/
odoo@709ffc01774a:/$
```

5. Creación de la estructura básica del módulo.

```
PS C:\Users\Usuario\Desktop\Alicia\2. DAM\4. SGE\SGE_practicas\PR06\pr0606> docker compose up -d
[+] Running 2/2
 ✓ Container db          Started
    0.6s
 ✓ Container odoo_16     Started
    1.0s
PS C:\Users\Usuario\Desktop\Alicia\2. DAM\4. SGE\SGE_practicas\PR06\pr0606> docker exec -it 709ffc017 bash
odoo@709ffc01774a:/$ odoo scaffold taller /mnt/extra-addons/
odoo@709ffc01774a:/$
```

6. Instalación del módulo.



7. Edición de los archivos necesarios para la correcta implementación del módulo.

__manifest__.py

```
# -*- coding: utf-8 -*-
{
    "name": "taller",
    "summary": ""
        Short (1 phrase/line) summary of the module's purpose, used as
        subtitle on modules listing or apps.openerp.com"",
    "description": ""
        Long description of module's purpose
    "",
    "author": "My Company",
    "website": "https://www.yourcompany.com",
    # Categories can be used to filter modules in modules listing
    # Check
https://github.com/odoo/odoo/blob/16.0/odoo/addons/base/data/ir\_module\_category\_data.xml
    # for the full list
    "category": "Uncategorized",
    "version": "0.1",
    # any module necessary for this one to work correctly
    "depends": ["base"],
    # always loaded
    "data": [
        "security/ir.model.access.csv",
        "views/cliente.xml",
        "views/pieza.xml",
        "views/reparacion.xml",
        "views/vehiculo.xml",
        "views/menu.xml",
    ],
    # only loaded in demonstration mode
    "demo": [
        "demo/demo.xml",
    ],
}
```

ir.model.access.csv

	id	name	model_id:id	group_id:id	perm_read	perm_write	perm_create	perm_unlink
1	access_taller_cliente	taller.cliente	model_taller_cliente	base.group_user	1	1	1	1
2	access_taller_pieza	taller.pieza	model_taller_pieza	base.group_user	1	1	1	1
3	access_taller_reparacion	taller.reparacion	model_taller_reparacion	base.group_user	1	1	1	1
4	access_taller_vehiculo	taller.vehiculo	model_taller_vehiculo	base.group_user	1	1	1	1
5								

taller_cliente.py

```
from odoo import models, fields

class Cliente(models.Model):
    _name = "taller.cliente"
    _description = "Cliente"
    _sql_constraints = [
        ('unique_name', 'unique(name)', 'El nombre debe ser único'),
    ]

    name = fields.Char(string="Nombre", required=True)

    telefono = fields.Integer(string="Teléfono")

    email = fields.Char(string="e-mail")

    direccion = fields.Char(string="Dirección")

    vehiculo_ids = fields.One2many(
        comodel_name="taller.vehiculo", inverse_name="cliente_id",
        string="Vehículos"
    )
```

taller_pieza.py

```
from odoo import models, fields, api

class Pieza(models.Model):
    _name = "taller.pieza"
    _description = "Pieza"
    _sql_constraints = [
        ("unique_nombre", "unique(nombre)", "El nombre debe ser único"),
        (
            "precio_unitario",
            "CHECK(precio_unitario >= 0)",
            "El precio debe ser mayor que 0",
        ),
    ]

    nombre = fields.Char(string="Nombre de la pieza")

    codigo = fields.Char(
        string="Código de la pieza",
    )
```

```

precio_unitario = fields.Float(string="Precio unitario")

reparacion_ids = fields.Many2many(
    string="reparacion",
    comodel_name="taller.reparacion",
    relation="taller_reparacion_this_taller_pieza_rel",
    column1="taller_pieza_id",
    column2="taller_reparacion_id",
)

```

taller_reparacion.py

```

from odoo import models, fields, api

class Reparacion(models.Model):
    _name = "taller.reparacion"
    _description = "Reparación"
    _sql_constraints = [
        ("coste_total", "CHECK(coste_total >= 0)", "El coste total debe ser positivo"),
    ]

    fecha_inicio = fields.Date(
        string="Fecha de inicio",
    )

    fecha_fin = fields.Date(
        string="Fecha de finalización",
    )

    descripcion = fields.Char(string="Descripción")

    estado = fields.Selection(
        string="Estado",
        selection=[
            ("borrador", "Borrador"),
            ("en_curso", "En curso"),
            ("finalizada", "Finalizada"),
        ],
    )

    coste_total = fields.Float(string="Coste total",
compute="_compute_coste_total")

    vehiculo_id = fields.Many2one(
        string="Vehículo",
        comodel_name="taller.vehiculo",

```

```

        ondelete="restrict",
    )

    pieza_ids = fields.Many2many(
        string="Piezas",
        comodel_name="taller.pieza",
        relation="taller_pieza_this_taller_reparacion_rel",
        column1="taller_reparacion_id",
        column2="taller_pieza_id",
    )

    tecnico_id = fields.Many2one(
        string="tecnico",
        comodel_name="res.users",
        ondelete="restrict",
    )

    @api.depends("pieza_ids")
    def _compute_coste_total(self):
        for record in self:
            record.coste_total = sum(
                pieza.precio_unitario for pieza in record.pieza_ids
            )

    def abrir_reparacion(self):
        for record in self:
            record.estado = "en_curso"

    def cerrar_reparacion(self):
        for record in self:
            record.estado = "finalizada"

    def reabrir_reparacion(self):
        for record in self:
            record.estado = "en_curso"

```

taller_vehiculo.py

```

from odoo import models, fields

class Vehiculo(models.Model):
    _name = "taller.vehiculo"
    _description = "Vehículo"
    _sql_constraints = [
        ('unique_matricula', 'unique(matricula)', 'La matrícula debe ser
        única'),
    ]

```

```

]

matricula = fields.Char(string="Matrícula", required=True)

marca = fields.Char(string="Marca")

modelo = fields.Char(string="Marca")

anio = fields.Integer(string="Año")

cliente_id = fields.Many2one(
    string="Cliente",
    comodel_name="taller.cliente",
    ondelete="restrict",
)

reparacion_ids = fields.One2many(
    string="Reparación",
    comodel_name="taller.reparacion",
    inverse_name="vehiculo_id",
)

```

menu.xml

```

<odoo>
  <data>

    <!-- Top menu item -->

    <menuitem name="taller" id="taller.menu_root" />

    <!-- menu categories -->

    <menuitem name="Cliente" id="taller.menu_cliente"
parent="taller.menu_root"
    action="taller.action_taller_cliente" />

    <menuitem name="Pieza" id="taller.menu_pieza"
parent="taller.menu_root"
    action="taller.action_taller_pieza" />

    <menuitem name="Reparacion" id="taller.menu_reparacion"
parent="taller.menu_root"
    action="taller.action_taller_reparacion" />

```

```

        <menuitem name="Vehiculo" id="taller.menu_vehiculo"
parent="taller.menu_root"
        action="taller.action_taller_vehiculo" />

    </data>
</odoo>

```

cliente.xml

```

<odoo>
  <data>
    <!-- explicit list view definition -->

    <record model="ir.ui.view" id="taller_cliente">
      <field name="name">Cliente</field>
      <field name="model">taller.cliente</field>
      <field name="arch" type="xml">
        <tree>
          <field name="name" />
          <field name="telefono" />
          <field name="email" />
          <field name="direccion" />
          <field name="vehiculo_ids" />
        </tree>
      </field>
    </record>

    <!-- actions opening views on models -->

    <record model="ir.actions.act_window" id="taller.action_taller_cliente">
      <field name="name">Cliente</field>
      <field name="res_model">taller.cliente</field>
      <field name="view_mode">tree,form</field>
    </record>

  </data>
</odoo>

```

pieza.xml

```

<odoo>
  <data>
    <!-- explicit list view definition -->

    <record model="ir.ui.view" id="taller_pieza">
      <field name="name">Pieza</field>
      <field name="model">taller.pieza</field>

```

```

    <field name="arch" type="xml">
        <tree>
            <field name="nombre" />
            <field name="codigo" />
            <field name="precio_unitario" decoration-danger="precio_unitario
&gt; 100"
                decoration-success="precio_unitario &lt; 10"
                decoration-muted="precio_unitario &lt; 10 &lt; 100" />
            <field name="reparacion_ids" />
        </tree>
    </field>
</record>

<!-- actions opening views on models -->

<record model="ir.actions.act_window" id="taller.action_taller_pieza">
    <field name="name">Pieza</field>
    <field name="res_model">taller.pieza</field>
    <field name="view_mode">tree,form</field>
</record>

</data>
</odoo>

```

reparación.xml

```

<odoo>
    <data>
        <!-- explicit list view definition -->

        <record model="ir.ui.view" id="taller_reparacion">
            <field name="name">Reparacion</field>
            <field name="model">taller.reparacion</field>
            <field name="arch" type="xml">
                <tree>
                    <field name="fecha_inicio" />
                    <field name="fecha_fin" />
                    <field name="descripcion" />
                    <field name="estado" decoration-success="estado == 'finalizada'"
                        decoration-primary="estado == 'en_curso'" decoration-muted="estado
== 'borrador'" />
                    <button name="abrir_reparacion"
                        type="object"
                        string="Abrir"
                        class="btn-success"
                        icon="fa-edit"
                        attrs="{ 'invisible': [('estado', '!=', 'borrador')]}" />
                </tree>
            </field>
        </record>
    </data>
</odoo>

```

```

        <button name="cerrar_reparacion"
            type="object"
            string="Cerrar"
            class="btn-danger"
            icon="fa-edit"
            attrs="{ 'invisible': [('estado', '!=', 'en_curso')]}" />

        <button name="reabrir_reparacion"
            type="object"
            string="Reabrir"
            class="btn-secondary"
            icon="fa-edit"
            attrs="{ 'invisible': [('estado', '!=', 'finalizada')]}" />

        <field name="coste_total" decoration-danger="coste_total < 500"
            decoration-warning="coste_total >= 100 and coste_total <=
500"
            decoration-success="coste_total < 100" />
        <field name="vehiculo_id" />
        <field name="pieza_ids" />
        <field name="tecnico_id" />
    </tree>
</field>
</record>

<!-- actions opening views on models -->

<record model="ir.actions.act_window"
id="taller.action_taller_reparacion">
    <field name="name">Reparacion</field>
    <field name="res_model">taller.reparacion</field>
    <field name="view_mode">tree,form</field>
</record>

</data>
</odoo>
<odoo>
<data>
    <!-- explicit list view definition -->

    <record model="ir.ui.view" id="taller_reparacion">
        <field name="name">Reparacion</field>
        <field name="model">taller.reparacion</field>
        <field name="arch" type="xml">
            <tree>
                <field name="fecha_inicio" />

```

```

        <field name="fecha_fin" />
        <field name="descripcion" />
        <field name="estado" decoration-success="estado == 'finalizada'"
            decoration-primary="estado == 'en_curso'" decoration-muted="estado
== 'borrador'" />
        <button name="abrir_reparacion"
            type="object"
            string="Abrir"
            class="btn-success"
            icon="fa-edit"
            attrs="{ 'invisible': [('estado', '!=', 'borrador')]}" />

        <button name="cerrar_reparacion"
            type="object"
            string="Cerrar"
            class="btn-danger"
            icon="fa-edit"
            attrs="{ 'invisible': [('estado', '!=', 'en_curso')]}" />

        <button name="reabrir_reparacion"
            type="object"
            string="Reabrir"
            class="btn-secondary"
            icon="fa-edit"
            attrs="{ 'invisible': [('estado', '!=', 'finalizada')]}" />

        <field name="coste_total" decoration-danger="coste_total < 500"
            decoration-warning="coste_total >= 100 and coste_total <=
500"
            decoration-success="coste_total < 100" />
        <field name="vehiculo_id" />
        <field name="pieza_ids" />
        <field name="tecnico_id" />
    </tree>
</field>
</record>

<!-- actions opening views on models -->

<record model="ir.actions.act_window"
id="taller.action_taller_reparacion">
    <field name="name">Reparacion</field>
    <field name="res_model">taller.reparacion</field>
    <field name="view_mode">tree,form</field>
</record>

</data>

```

```
</odoo>
```

vehiculo.xml

```
<odoo>
  <data>
    <!-- explicit list view definition -->

    <record model="ir.ui.view" id="taller_vehiculo">
      <field name="name">Vehiculo</field>
      <field name="model">taller.vehiculo</field>
      <field name="arch" type="xml">
        <tree>
          <field name="matricula" />
          <field name="marca" />
          <field name="modelo" />
          <field name="anio" />
          <field name="cliente_id" />
          <field name="reparacion_ids" />
        </tree>
      </field>
    </record>

    <!-- actions opening views on models -->

    <record model="ir.actions.act_window" id="taller.action_taller_vehiculo">
      <field name="name">Vehículo</field>
      <field name="res_model">taller.vehiculo</field>
      <field name="view_mode">tree,form</field>
    </record>

  </data>
</odoo>
```

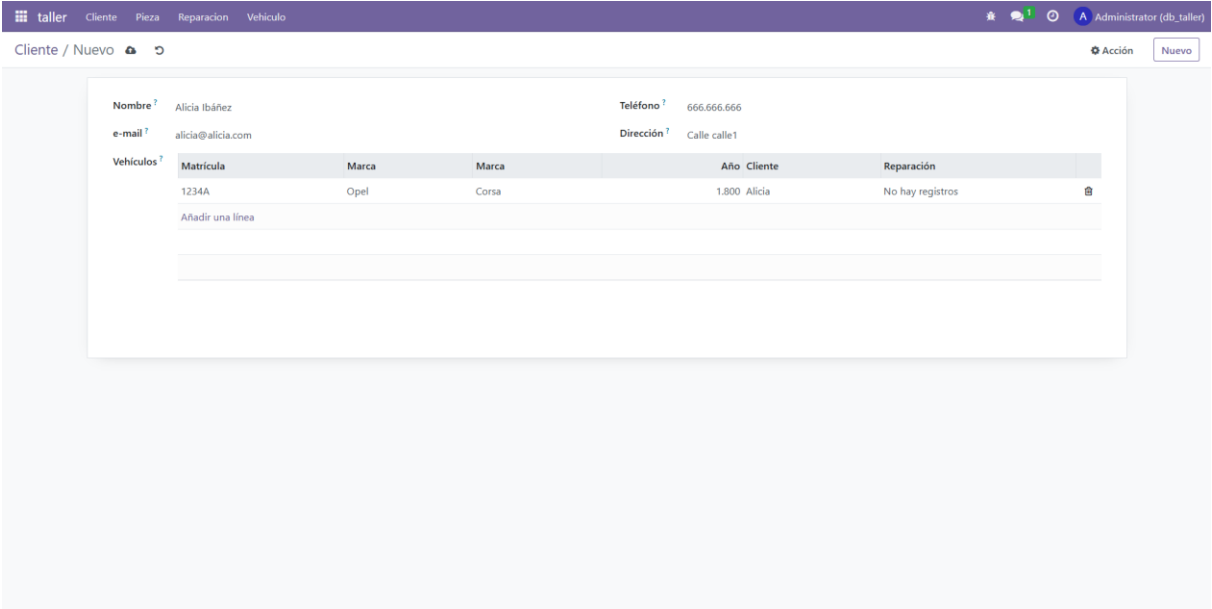
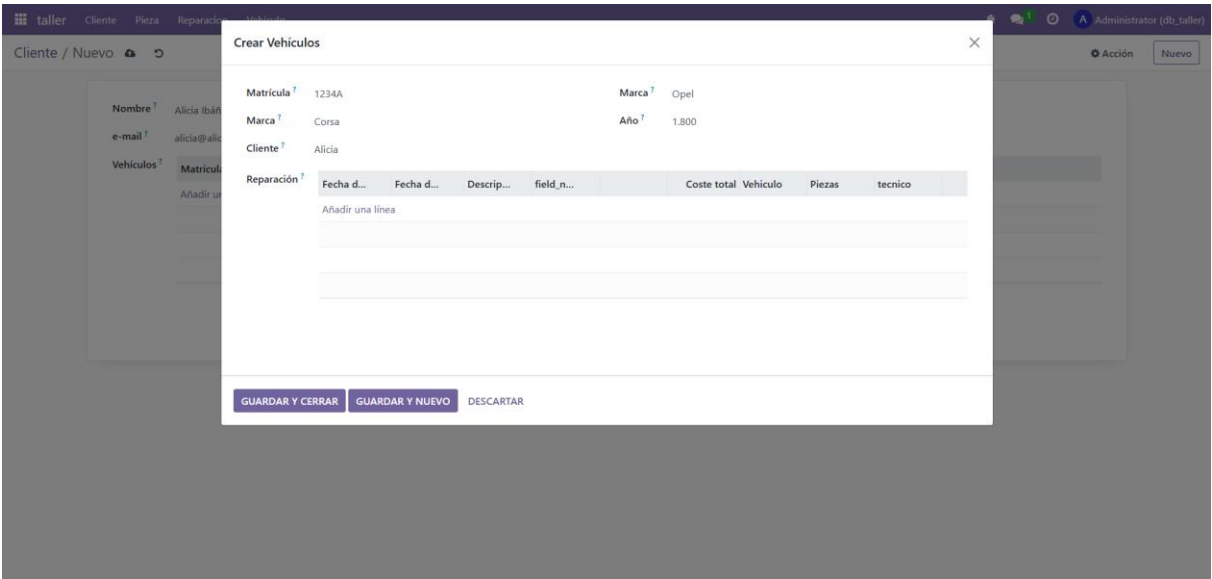
__init__.py

```
# -*- coding: utf-8 -*-
from . import taller_cliente
from . import taller_pieza
from . import taller_reparacion
from . import taller_vehiculo
```

8. Agregación de registros para comprobar el correcto funcionamiento

taller Cliente Pieza Reparacion Vehiculo					Administrator (db_taller)				
Cliente					Buscar...				
NUEVO					Filtros Agrupar por Favoritos				
1-1 / 1									
Nombre					Teléfono e-mail		Dirección		Vehiculos
Alicia					0				No hay registros

taller Cliente Pieza Reparacion Vehiculo					Administrator (db_taller)						
Cliente / Nuevo					Acción Nuevo						
Nombre ?					Teléfono ?						
Alicia Ibáñez					666.666.666						
e-mail ?					Dirección ?						
alicia@alicia.com					Calle calle1						
Vehiculos ?											
Matricula					Marca		Marca		Año Cliente Reparación		
Añadir una línea											



Nombre [?]

Aceite

Código de la pieza [?]

123456B

Precio unitario [?]

15,00

reparacion [?]

Fecha d...	Fecha d...	Descrip...	field_n...	Coste total	Vehiculo	Piezas	tecnico
Añadir una línea							

GUARDAR

DESCARTAR

GUARDAR Y CERRAR

GUARDAR Y NUEVO

DESCARTAR

Nombre ?

Alicia Ibáñez

e-mail ?

alicia@alic...

Vehículos ?

Matrícula

1234A

Añadir un...

Acción

2 / 2

< >

Nuevo

Administrador (db_taller)

Abrir: Vehiculos

X

Matrícula ?

1234A

Marca ?

Opel

Marca ?

Corsa

Año ?

1.800

Cliente ?

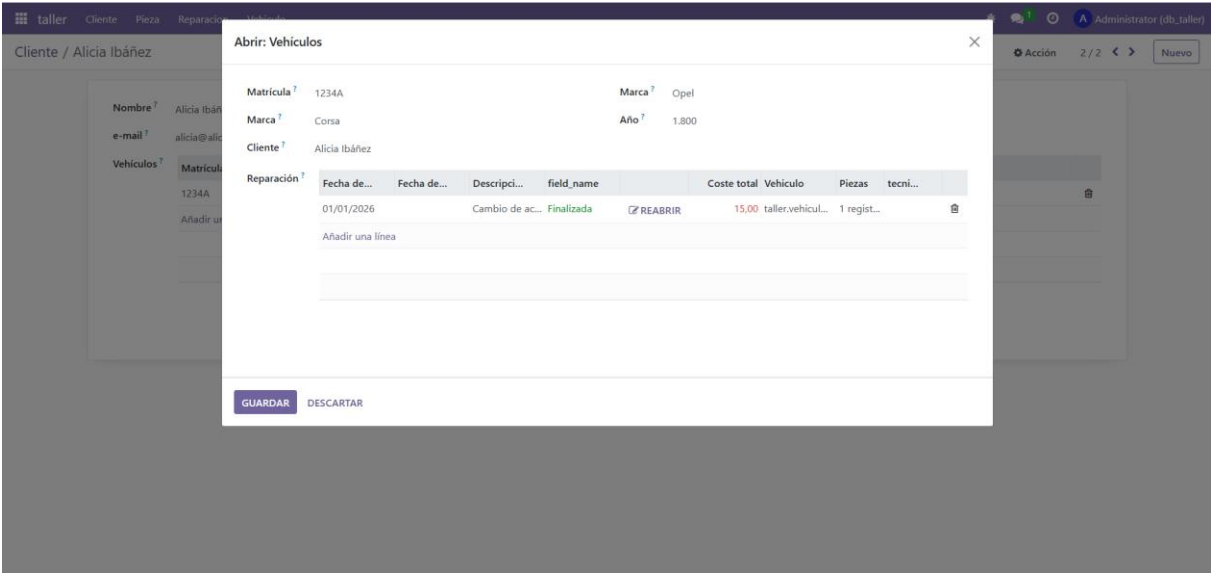
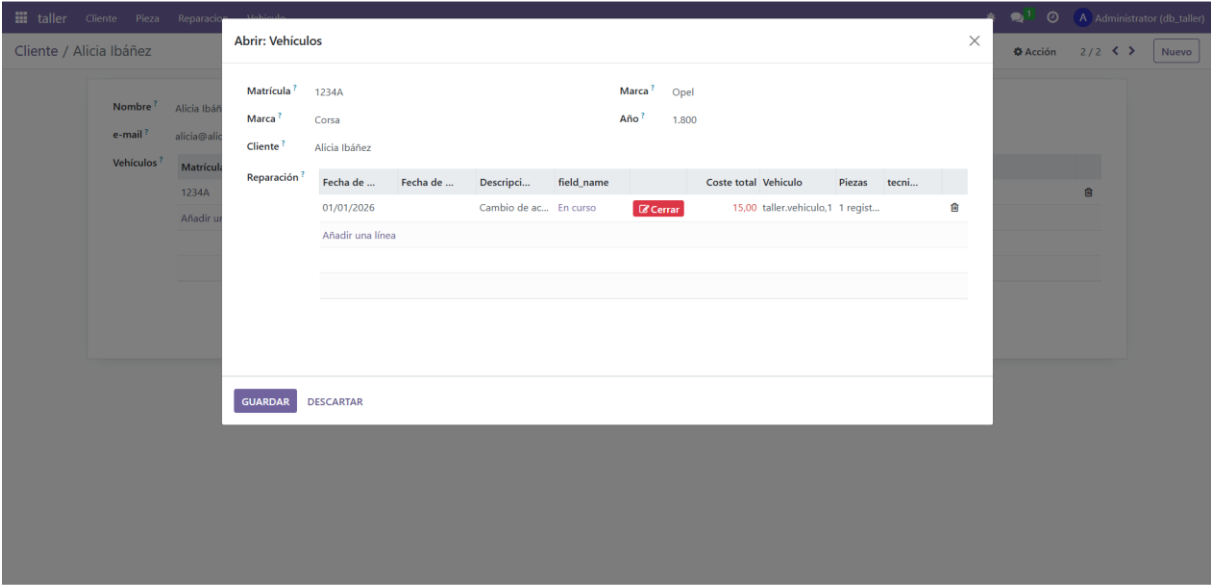
Alicia Ibáñez

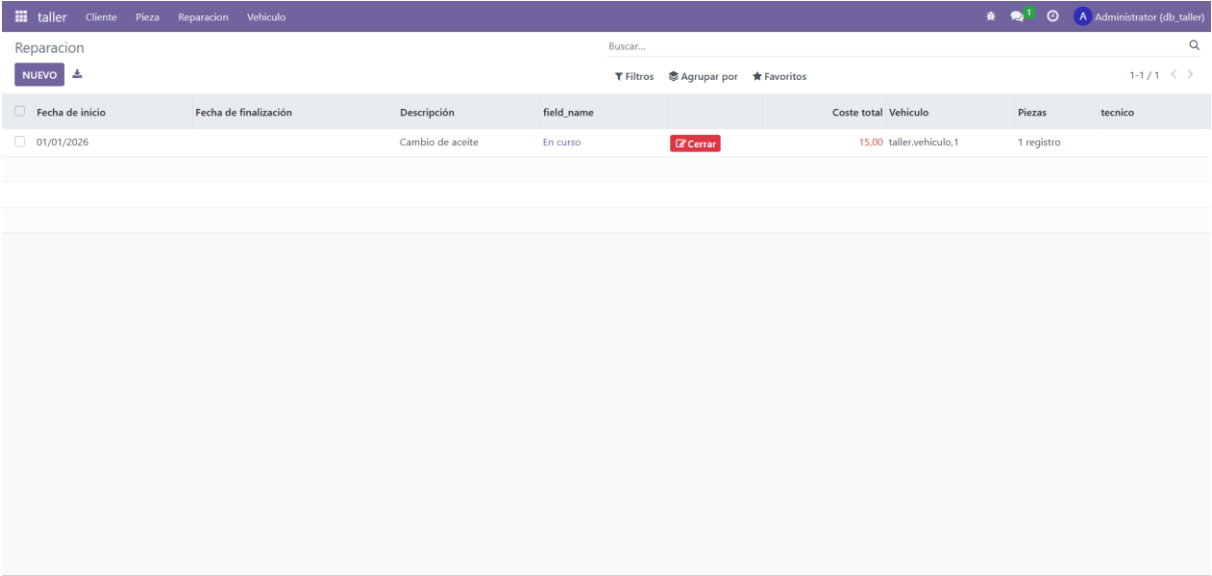
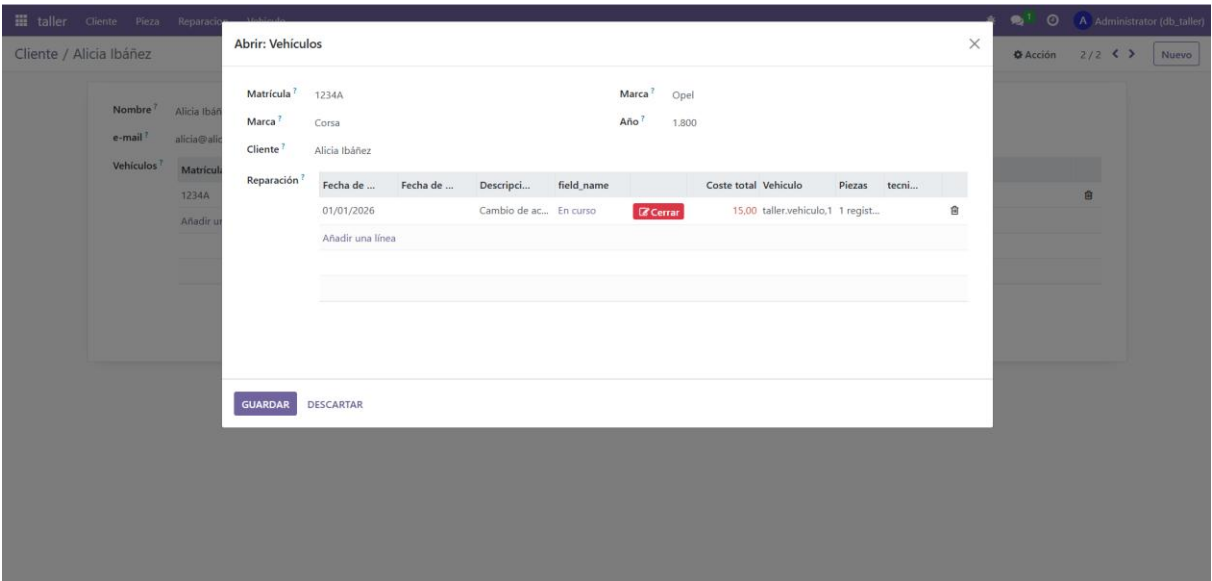
Reparación ?

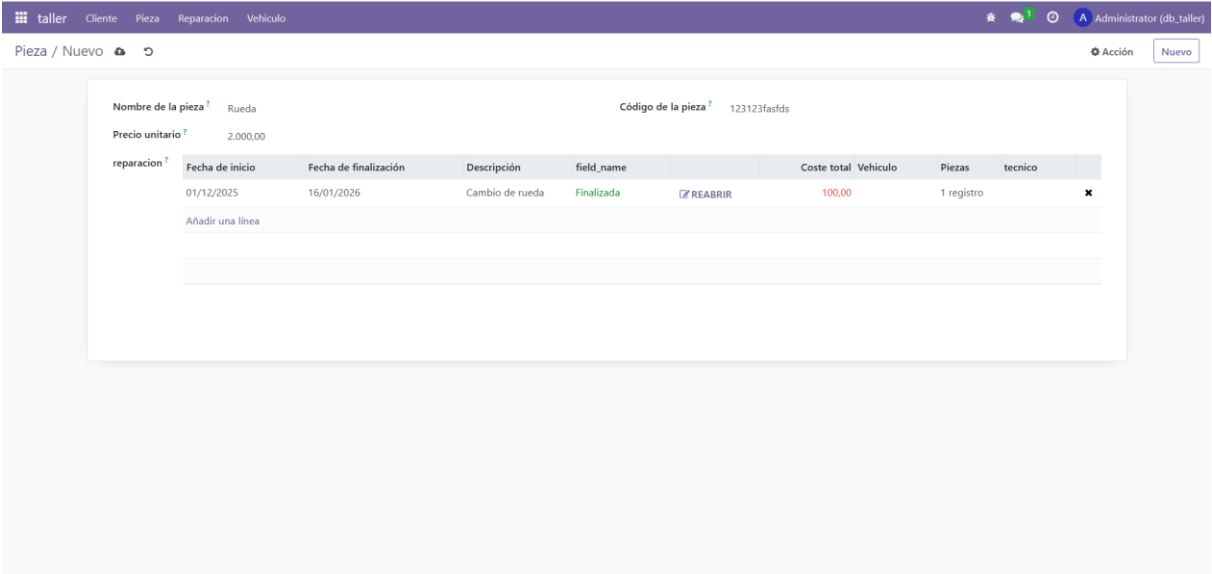
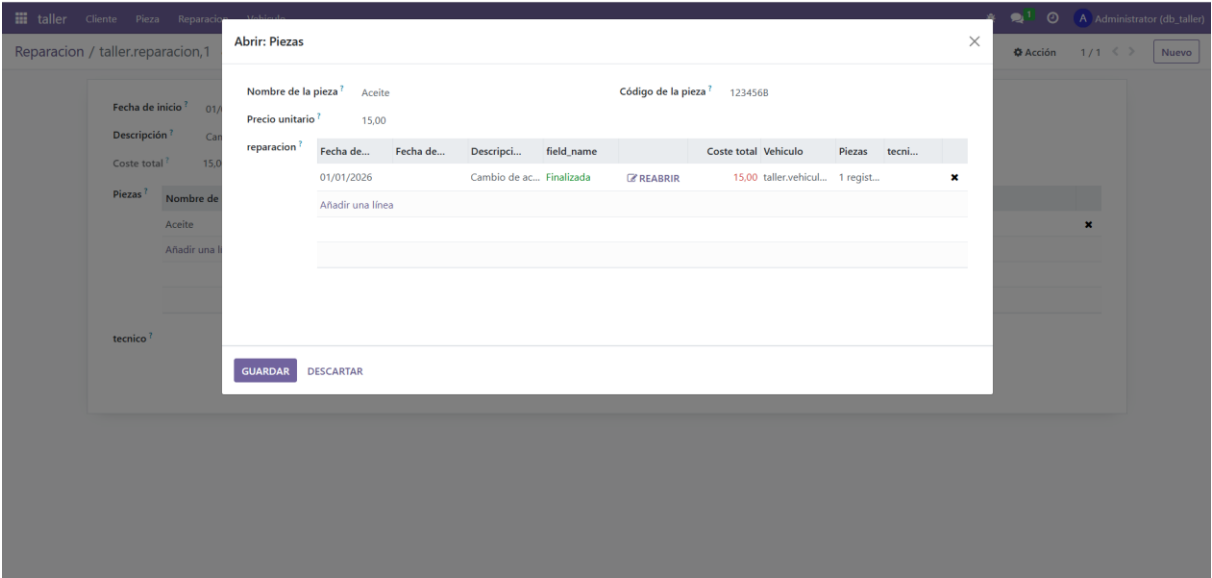
Fecha de i...	Fecha de ...	Descripción	field_name	Coste total	Vehiculo	Piezas	tecni...
01/01/2026		Cambio de ac...	Borrador	15,00	taller.vehiculo.1	1 regist...	
Añadir una línea							

GUARDAR

DESCARTAR







taller

Cliente

Pieza

Reparacion

Vehiculo

<

Reparacion / Nuevo

Fecha de inicio

Descripción

Coste total

Piezas

Nombre de

Rueda

Aceite

Añadir una

tecnico

Crear tecnico

Estás invitando a un nuevo usuario

Nombre

Asier

Dirección de correo electrónico

p. ej. email@sucompañía.com

Permisos de acceso

Técnico / Acceso a direcciones priv.

Técnico / Acceso a la función de ex.

Contabilidad / Administrador de F.

Permisos extra / Características té.

Permisos extra / Creación de conta.

Técnico / Editor de plantillas de co.

Contabilidad / Facturación

Técnico / Impuesto B2B

Tipos de usuario / Usuario interno

Podrá definir reglas de acceso adicionales editando el usuario recién creado bajo el menú Ajustes/Usuarios.

Teléfono

Móvil

GUARDAR Y CERRAR

DESCARTAR

