

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/extr-addons/
odoo@709ffc01774a:/$ █
```

2. Establecimiento de credenciales.

Nota: La base de datos se llama db_taller

The screenshot shows the Odoo database setup interface. It includes a warning message about the master password being generated, followed by a field for the master password itself. Below this are fields for the database name, email, password, phone number, language (set to Spanish/Español), country (set to Spain), and a checkbox for demo data. At the bottom are buttons for 'Create database' and 'or restore a database'.

Warning, your Odoo database manager is not protected. To secure it, we have generated the following master password for:	
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"/>
Database Name	db_subscription
Email	alicia.ibanez.escudero@gmail.com
Password <input type="password"/>
Phone number	666666666
Language	Spanish / Español
Country	Spain
Demo data	<input type="checkbox"/>
Create database or restore a database	

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

The screenshot shows the Odoo App Store interface. On the left, there's a sidebar titled 'CATEGORÍAS' with a 'Todos' section listing various modules like Ventas, Servicios, Contabilidad, etc., each with a count (e.g., Ventas: 7). The main area displays a grid of applications:

- Ventas: De presupuestos a facturas. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Facturación: Facturas y pagos. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- CRM: Seguimiento de leads y oportunidades próximas. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- MRP II: Órdenes de trabajo, planificación, informes de inventario. Includes 'ACTUALIZAR' button.
- Sitio web: Creador de sitios web para empresas. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Inventario: Gestione sus actividades de inventario y logística. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Contabilidad: Gestione la contabilidad financiera y analítica. Includes 'ACTUALIZAR' button.
- Información: Centralice, gestione, comparta y haga crecer su librería de conocimiento. Includes 'ACTUALIZAR' button.
- Compra: Pedidos de compra, licitaciones y acuerdos. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Punto de venta: Interfaz de TPV amigable para usuarios para tiendas y restaurantes. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Proyecto: Organice y planifique sus proyectos. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Comercio electrónico: Venda sus productos en línea. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Fabricación: Órdenes de fabricación y listas de materiales. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Marketing por correo electrónico: Diseñe, envíe y rastree correos electrónicos. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Partes de horas: Seguimiento del tiempo de empleados en las tareas. Includes 'APRENDA MÁS' and 'ACTUALIZAR' buttons.
- Gastos
- Studio
- Ausencias

3. Activación del modo de desarrolladores en Ajustes.

The screenshot shows the Odoo 'Ajustes' (Settings) screen. On the left, there's a sidebar with 'Opciones generales' selected. The main area has tabs for 'GUARDAR' and 'DESCARTAR'. The 'Opciones generales' tab is active, showing the following configuration options:

- Autenticación OAuth**: Use cuentas externas (Google, Facebook, etc.) para iniciar sesión. Includes an input field for 'Clave de acceso' and another for 'ID de aplicación'.
- Biblioteca de imágenes de Unsplash**: Encuentre imágenes gratuitas de alta resolución de Unsplash. Includes an input field for 'Clave de acceso' and a note about generating a new access key.
- Geolocalización**: Localiza geográficamente sus contactos.
- reCAPTCHA**: Proteja sus formularios de spam y abusos.

Herramientas de desarrollador (Developer Tools):
- **Activar modo de desarrollador**: Includes three sub-options: 'Activar modo de desarrollador (con activos)', 'Activar modo de desarrollador (con activos de prueba)', and 'Activar el modo de desarrollador (con activos de prueba)'.

Sobre (About): Includes links to 'GET IT ON Google Play' and 'Download on the App Store'.

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.

The screenshot shows the Odoo application manager interface. On the left, there's a sidebar titled 'CATEGORÍAS' with various modules listed under 'Todos': Ventas, Servicios, Contabilidad, Inventario, Fabricación, Sitio web, Marketing, Recursos Humanos, Productividad, Técnico, and Administración. The 'taller' module is selected and highlighted in blue. In the main content area, there's a card for the 'taller' module with the following details: icon, name 'taller', status 'Instalado' (Installed), and a link 'APRENDA MÁS' (Learn More). At the top, there are navigation tabs like 'Aplicaciones', 'Actualizaciones', and 'Actualizar lista de aplicaciones'. On the right, there are search and filter options, and at the bottom right, there are pagination controls.

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_categor
y_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

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="técnico",
    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 < 10"
            decoration-muted="precio_unitario < 10 &lt; 100" />
        <field name="reparacion_ids" />
    </tree>
</field>
</record>

<!-- actions opening views on models --&gt;

&lt;record model="ir.actions.act_window" id="taller.action_taller_pieza"&gt;
    &lt;field name="name"&gt;Pieza&lt;/field&gt;
    &lt;field name="res_model"&gt;taller.pieza&lt;/field&gt;
    &lt;field name="view_mode"&gt;tree,form&lt;/field&gt;
&lt;/record&gt;

&lt;/data&gt;
&lt;/odoo&gt;
</pre>

```

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)			
								buscar...	1	0	A
Cliente								Filtros	Agrupar por	Favoritos	
NUEVO											
<input type="checkbox"/>	Nombre			Teléfono	e-mail	Dirección	Vehiculos				
	<input type="checkbox"/>	Alicia		0			No hay registros				

taller						Cliente	Pieza	Reparacion	Vehiculo	Administrator (db_taller)			
										buscar...	1	0	A
Cliente / Nuevo													
Nombre ⁷ Alicia Ibáñez						Teléfono ⁷ 666.666.666							
e-mail ⁷ alicia@alicia.com						Dirección ⁷ Calle calle1							
Vehiculos ⁷						Matricula	Marca	Marca	Año	Cliente	Reparación		

Crear Vehículos

Nombre [?]	Alicia Ibáñez	Matrícula [?]	1234A	Marca [?]	Opel
e-mail [?]	alicia@alicia.com	Fecha d...	Corsa	Año [?]	1.800
Cliente [?]	Alicia	Descrip...		Coste total	Vehículo
Vehículos [?]	Matrícula	field_n...	Piezas	tecnico	
Reparación [?]					
Añadir una línea					

GUARDAR Y CERRAR **GUARDAR Y NUEVO** **DESCARTAR**

Crear Vehículos

Nombre [?]	Alicia Ibáñez	Matrícula [?]	1234A	Marca [?]	Opel	Marca [?]	Corsa	Teléfono [?]	666.666.666	Dirección [?]	Calle calle1
e-mail [?]	alicia@alicia.com	Fecha d...		Descrip...		Año [?]	Cliente	Año	Alicia	Reparación	No hay registros
Vehículos [?]	Matrícula	Marca	Marca	field_n...		Coste total	Vehículo	Piezas	tecnico		
Reparación [?]											
Añadir una línea											

Nombre: Alicia Ibáñez
e-mail: alicia@alicia.com
Vehículos: Matrícula: 1234A
Añadir una línea

Abrir: Piezas

Nombre de la pieza?	ACEITE	Código de la pieza?	123456B	
Precio unitario?	15,00			
reparación?	Fecha d... Añadir una línea	Fecha d... Añadir una línea	Descripción field_n...	Coste total Vehículo Piezas técnico

GUARDAR DESCARTAR

GUARDAR Y CERRAR **GUARDAR Y NUEVO** DESCARTAR

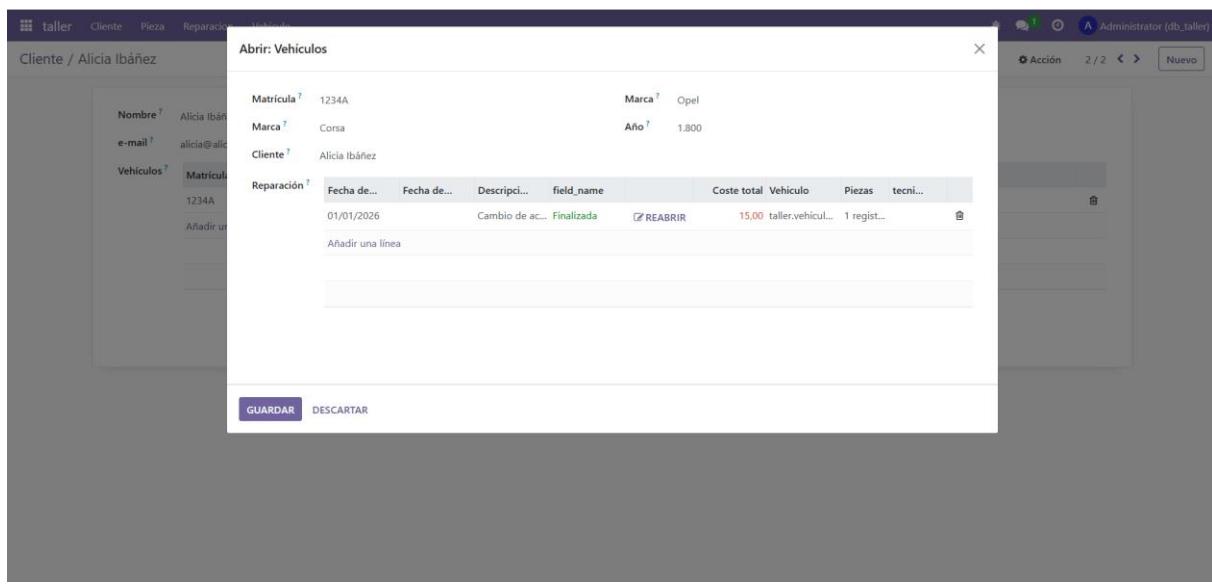
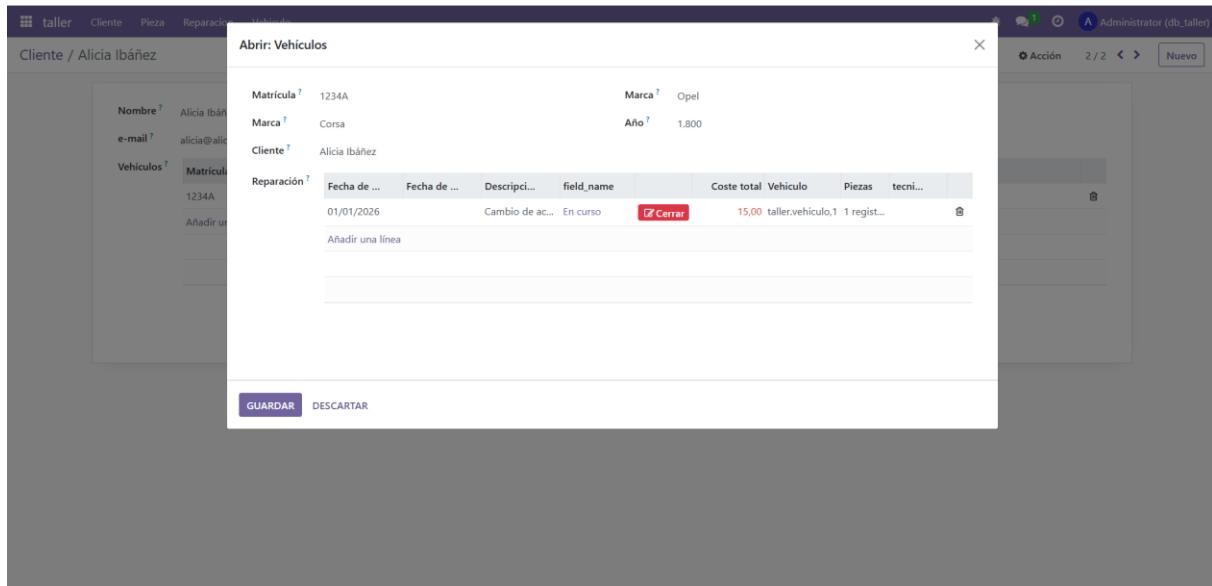
Abrir: Vehículos

Nombre?	Alicia Ibáñez	Matrícula?	1234A	Marca?	Opel
Marca?	Corsa	Año?	1.800		
e-mail?	alicia@al...				
Cliente?	Alicia Ibáñez				
Vehículos?	Matricul...				
	1234A				
	Añadir u...				

Reparación?

Fecha de i...	Fecha de ...	Descripción	field_name	Coste total	Vehículo	Piezas	tecn...
01/01/2026		Cambio de ac...	Borrador	<input checked="" type="checkbox"/> Abrir	15,00	taller.vehiculo.1	1 regist...
Añadir una línea							

GUARDAR **DESCARTAR**



Abrir: Vehiculos

Nombre [?]	Alicia Ibáñez	Matrícula [?]	1234A	Marca [?]	Opel					
Marca [?]	Corsa	Año [?]	1.800							
Cliente [?]	Alicia Ibáñez									
Vehiculos [?]	Matrícula 1234A Añadir una línea	Reparación [?]	Fecha de ... 01/01/2026	Fecha de ... Cambio de aceite	Descripción En curso	field_name	Coste total 15,00	Vehiculo taller.vehiculo,1	Piezas 1 registro...	tecni...
<input type="button" value="Cerrar"/> <input type="button" value="GUARDAR"/> <input type="button" value="DESCARTAR"/>										

Reparacion

NUEVO		Buscar...	Filtros	Agrupar por	Favoritos	1-1 / 1		
<input type="checkbox"/>	Fecha de inicio	Fecha de finalización	Descripción	field_name	Coste total	Vehiculo	Piezas	tecnico
<input type="checkbox"/>	01/01/2026		Cambio de aceite	En curso	<input type="button" value="Cerrar"/>	15,00	taller.vehiculo,1	1 registro
<input type="button" value="GUARDAR"/> <input type="button" value="DESCARTAR"/>								

taller Cliente Pieza Reparacion Vehiculo

Reparacion / taller.reparacion,1

Abrir: Piezas

Nombre de la pieza	Aceite	Código de la pieza	1234568					
Precio unitario	15,00							
reparacion	Fecha de inicio	Fecha de finalización	Descripción	field_name	Coste total	Vehiculo	Piezas	tecnico
	01/01/2026		Cambio de aceite	Finalizada	<input checked="" type="checkbox"/> REABRIR	15,00	taller.vehicul...	1 registro
Añadir una línea								

Piezas

- Aceite
- Añadir una línea

tecnico

GUARDAR **DESCARTAR**

taller Cliente Pieza Reparacion Vehiculo

Pieza / Nuevo

Abrir: Piezas

Nombre de la pieza	Rueda	Código de la pieza	123123fasfds					
Precio unitario	2.000,00							
reparacion	Fecha de inicio	Fecha de finalización	Descripción	field_name	Coste total	Vehiculo	Piezas	tecnico
	01/12/2025	16/01/2026	Cambio de rueda	Finalizada	<input checked="" type="checkbox"/> REABRIR	100,00		1 registro
Añadir una línea								

Pieza		
<input type="checkbox"/> Nombre de la pieza	Código de la pieza	Precio unitario reparacion
<input type="checkbox"/> Aceite	123456B	1.00 1 registro
<input type="checkbox"/> Rueda	ssfsfd1	100.00 No hay registros
<input type="checkbox"/> Ruedas	123123fasfds	2.000,00 1 registro

Estás invitando a un nuevo usuario

Nombre

Dirección de correo electrónico

Permisos de acceso

- Técnico / Acceso a direcciones privadas
- Técnico / Acceso a la función de exportación
- Contabilidad / Administrador de facturación
- Permisos extra / Características técnicas
- Permisos extra / Creación de controles
- Técnico / Editor de plantillas de cotizaciones
- 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**

taller Cliente Pieza Reparacion Vehiculo
 1
A Administrator (db_taller)

Reparacion

NUEVO

<input type="checkbox"/> Fecha de inicio	Fecha de finalización	Descripción	field_name		Coste total	Vehiculo	Piezas	tecnico
<input type="checkbox"/> 01/01/2026	16/01/2026	Cambio de aceite	Finalizada	<input checked="" type="checkbox"/> REABRIR	1,00	taller.vehiculo.1	1 registro	
<input type="checkbox"/> 01/12/2025	16/01/2026	Cambio de rueda	Finalizada	<input checked="" type="checkbox"/> REABRIR	100,00		1 registro	
<input type="checkbox"/> 01/01/2026	31/01/2026	sadasdsad	Borrador	<input checked="" type="checkbox"/> Abrir	101,00	taller.vehiculo.1	2 registros	Asier