

# CÓDIGOS EDITADOS

## \_\_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" />

```

```

        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 -->

<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">
                </tree>
            </field>
        </record>
    </data>
</odoo>

```

```

        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>
<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 --&gt;

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

```

```
</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
```