

Módulo Odoo - Sergi Saravia

1. Introducción

Este documento describe el desarrollo de un módulo personalizado para Odoo, realizado por Sergi Saravia en el marco de la unidad M10-UF2. El objetivo principal es gestionar una entidad personalizada mediante modelos, vistas, seguridad, menús y validaciones dentro de Odoo 18. También se incluyen capturas de pantalla explicativas que ilustran el resultado funcional del módulo.

2. Estructura del módulo

El módulo sigue una estructura estándar de Odoo, organizada del siguiente modo:

```
sergi_odoo/  
├── __init__.py  
├── __manifest__.py  
├── models/  
│   └── models.py  
├── views/  
│   └── views.xml  
├── security/  
│   └── ir.model.access.csv  
└── static/  
    ├── description/  
    │   ├── captura1.png  
    │   ├── captura2.png  
    │   └── ...
```

3. Fichero `__manifest__.py`

Este fichero define los metadatos del módulo. Incluye:

- Nombre del módulo: `sergi_odoo`

- Versión: 1.0
 - Autor: Sergi Saravia
 - Dependencias: `base`
 - Datos cargados: `security/ir.model.access.csv`, `views/views.xml`
 - Aplicación: True
-

4. Inicialización (`__init__.py`)

El archivo `__init__.py` importa el paquete de modelos para que Odoo pueda cargarlo correctamente:

```
from . import models
```

5. Modelos Python

El fichero `models/models.py` define un modelo principal, por ejemplo:

```
from odoo import models, fields, api

class Reserva(models.Model):
    _name = 'sergi.reserva'
    _description = 'Reserva'

    name = fields.Char(string='Nombre')
    fecha = fields.Date(string='Fecha')
    descripcion = fields.Text(string='Descripción')

    @api.onchange('name')
    def _onchange_name(self):
        if self.name:
            self.name = self.name.upper()
```

Este modelo permite registrar reservas con nombre, fecha y descripción, y aplica una validación que convierte el nombre a mayúsculas automáticamente.

6. Vistas XML

En el fichero `views/views.xml` se han definido:

- Vista árbol (tree): muestra un listado con los registros.
- Vista formulario (form): permite crear o editar un registro.
- Acciones: para enlazar las vistas al modelo.
- Menú personalizado: accesible desde el menú superior.

Ejemplo:

```
<menuitem id="menu_reserva_root" name="Reservas"/>
<menuitem id="menu_reserva" name="Gestión de Reservas"
parent="menu_reserva_root"/>
<record id="action_reserva" model="ir.actions.act_window">
    <field name="name">Reservas</field>
    <field name="res_model">sergi.reserva</field>
    <field name="view_mode">tree,form</field>
</record>
<menuitem id="menu_reserva_items" name="Reservas"
parent="menu_reserva" action="action_reserva"/>
```

7. Seguridad (`ir.model.access.csv`)

Se ha definido un archivo CSV con permisos de acceso para el modelo `sergi.reserva`, que permite a los usuarios leer, crear, modificar y eliminar registros.

Ejemplo de línea:

```
access_sergi_reserva_user,access.sergi.reserva.user,model_sergi_reserva,base.group_user,1,1,1,1
```

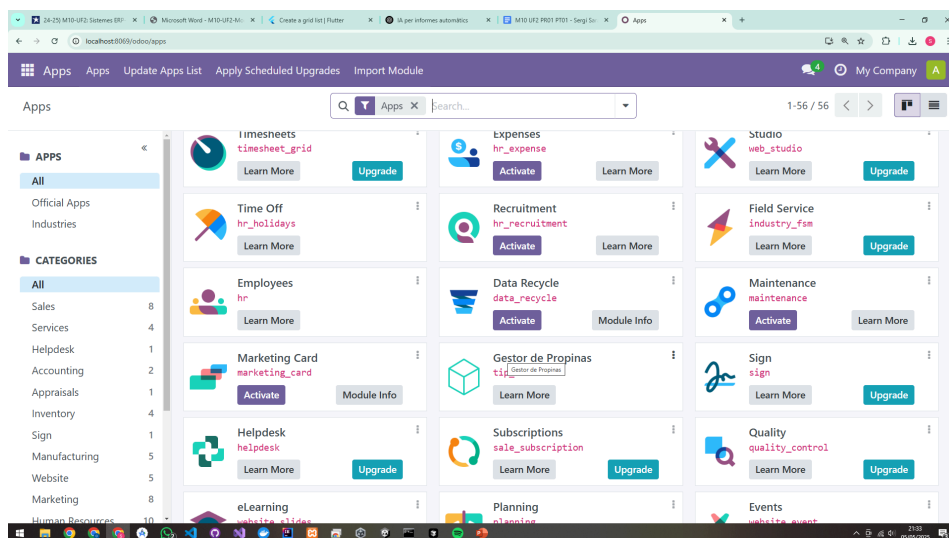
8. Validaciones y lógica

En el modelo Python se incluye una función que convierte automáticamente el campo **name** a mayúsculas con **@api.onchange**.

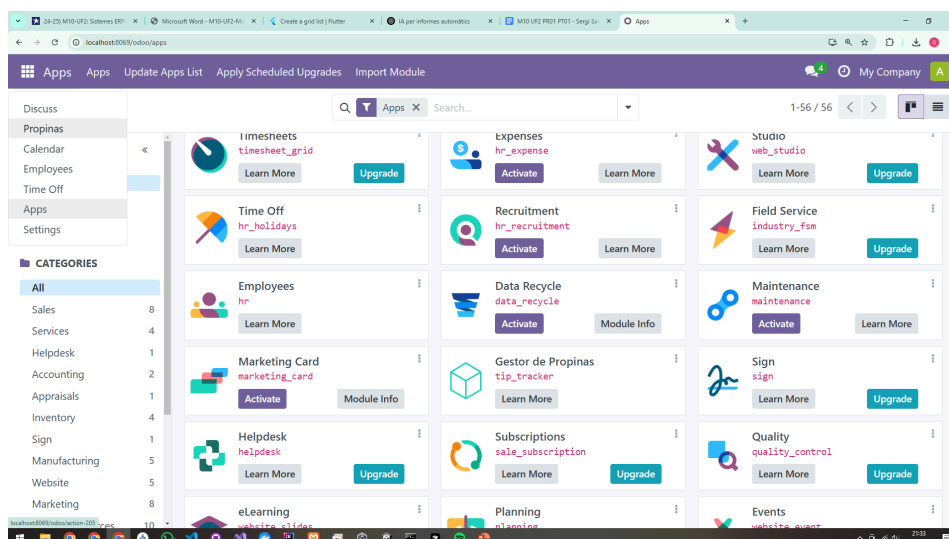
Esto asegura consistencia en la base de datos y mejora la calidad del dato ingresado.

9. Capturas de pantalla

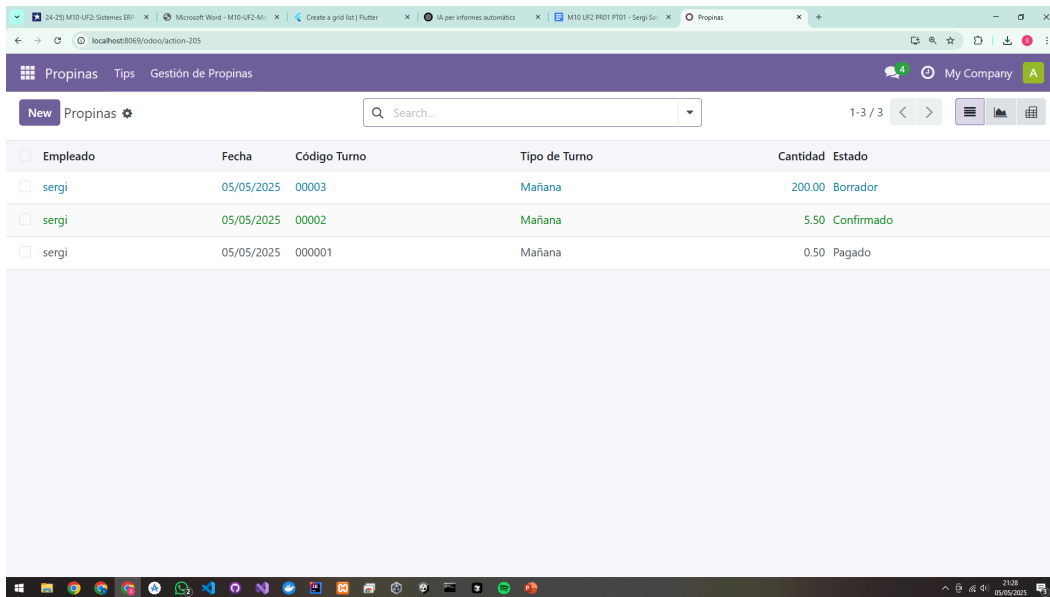
Entramos a Odoo con admin - admin y vemos el módulo



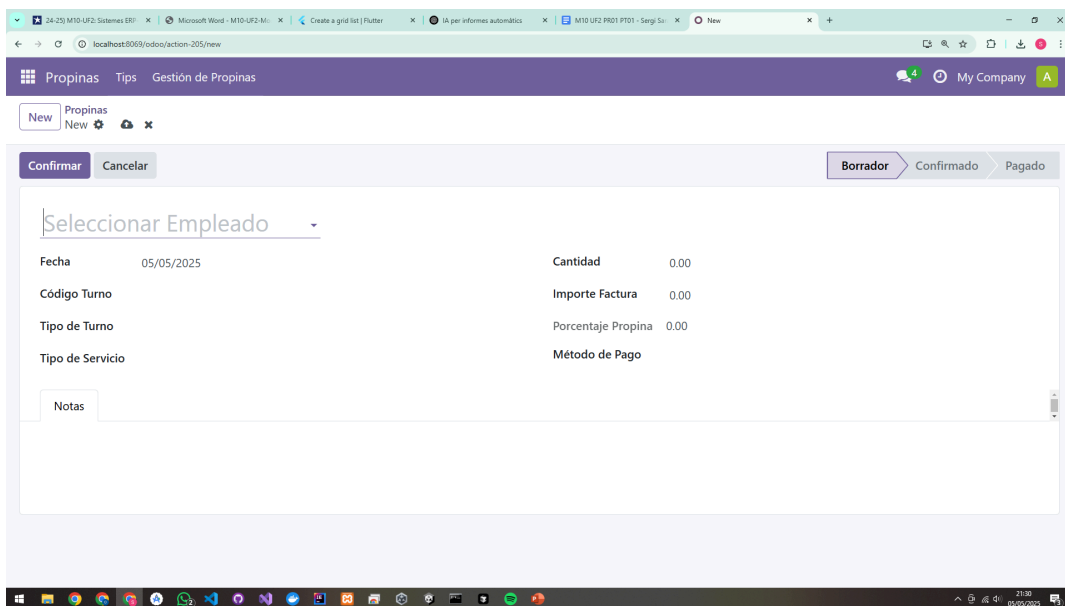
Vamos a la barra lateral desde dónde accedemos al módulo



Al entrar se muestra la pantalla de inicio con las propinas registradas.



Pulsando a nuevo, accedemos al menú, confirmamos y/o marcamos como



Navegación del menú personalizado dentro de Odoo.

[illegible]