Documentación del módulo Hola Mundo en Odoo

Santi Martínez

November 13, 2025

ÍNDICE

1 Introducción

2 Preparación del entorno

3 Activar del modo desarrollador en Odoo

Antes de proceder a su activación, cabe saber que el modo desarrollador permite ver y modificar la estructura interna de Odoo; permisos indispensables para crear o depurar módulos. El modo desarrollador en Odoo se puede activar de tres maneras diferentes:

1. Desde la interfaz propia de Odoo.
   - Entrar en Odoo con el usuario administrador.
   - **Ajustes** → **Activar modo desarrollador** (El cual solo será visible si hay al menos un módulo instalado)
2. Desde la URL:
   - Añadir al final de la URL: `?debug=1`
     Por ejemplo: `http://localhost:8069/web?debug=1`
3. Con una extensión en el navegador:
   - Firefox: `https://addons.mozilla.org/es/firefox/addon/odoo-debug/`
   - Chrome `https://chrome.google.com/webstore/detail/odoo-debug/hmdmhilocobgohohpdpoln`
     `hl=es_PR`

En esta documentación se tratará la opción propia de la interfaz de Odoo, la número uno.

3.1 Entrar en Odoo con el usuario administrador

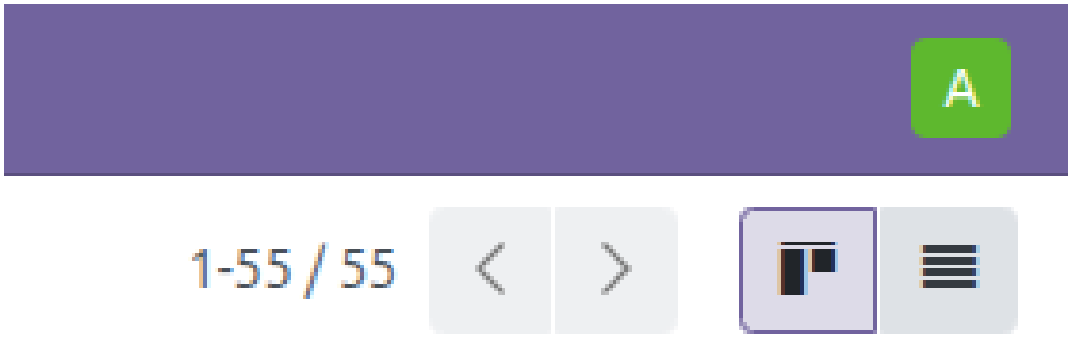


Figure 1: Odoo con usuario administrador

3.2 Activar modo desarrollador

Clicar en **Activar modo de desarrollador**

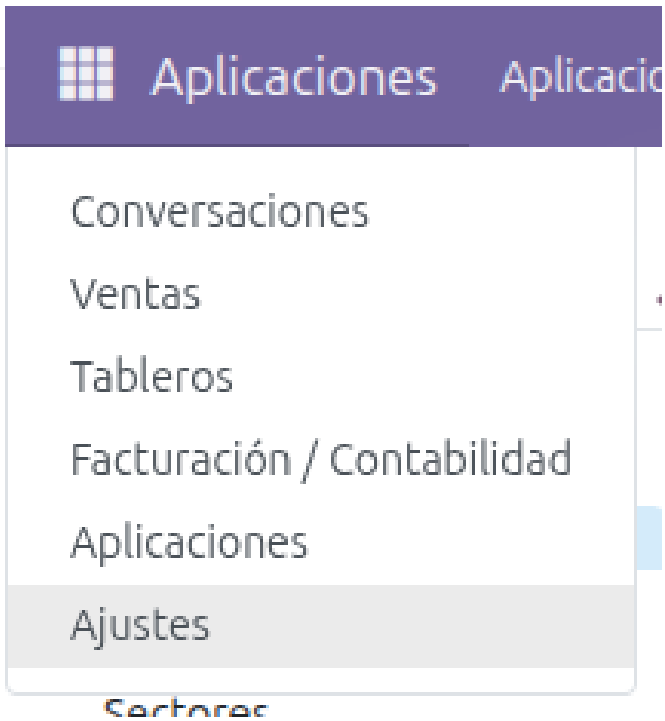


Figure 2: Ajustes

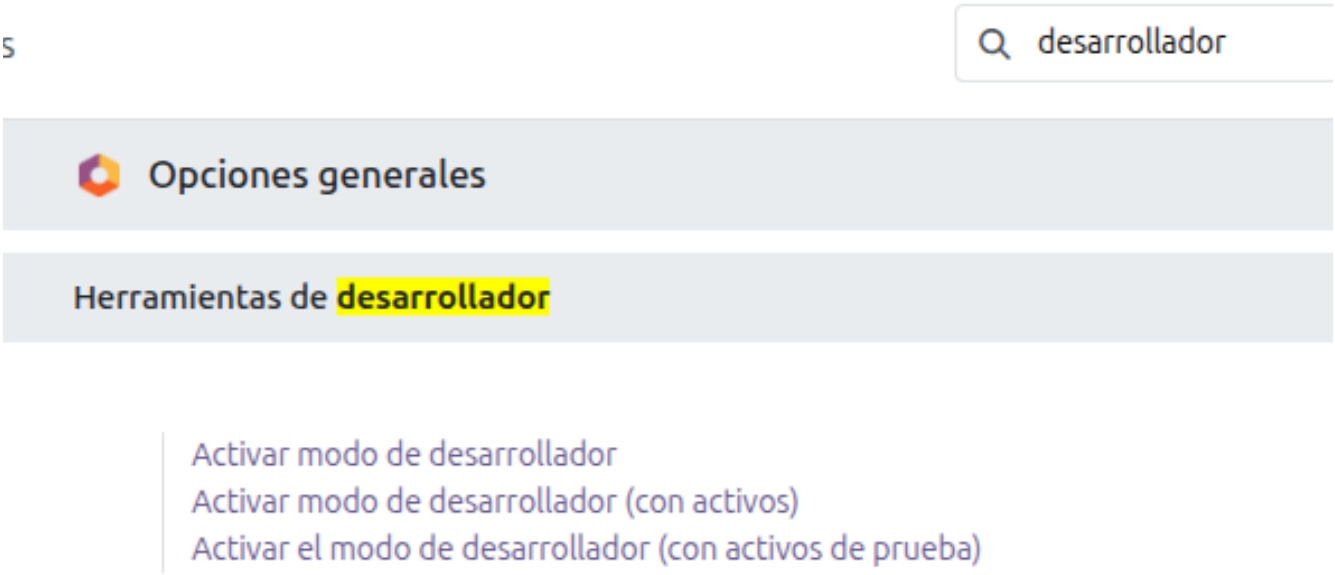


Figure 3: Activar modo desarrollador

4 Primer módulo: "Hola mundo"

El objetivo de este primer módulo es el comprobar que Odoo detacta correctamente los módulos que se crean de forma manual. La ubicación de los módulos personalizados se guarda dentro del directorio configurado para módulos, en este caso dentro de /módulos:

En el interior de esta carpeta, se crean dos ficheros:

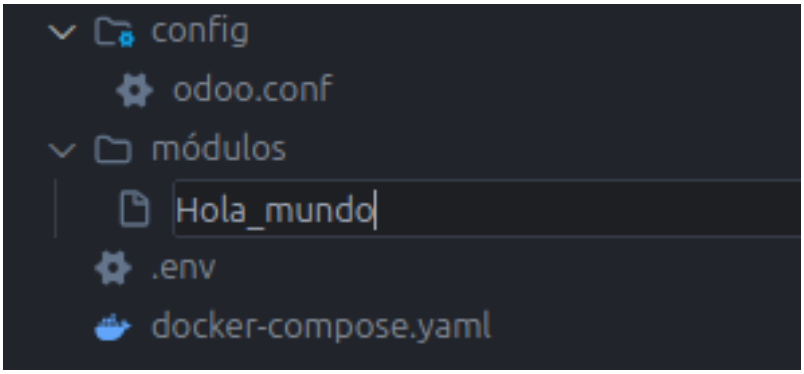
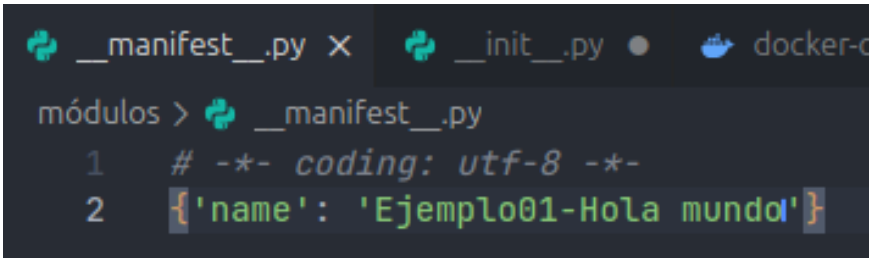


Figure 4: Ubicación del módulo holamundo

- **_init_.py**. El cual estará vacío
- **_manifest_.py**. Contiene el siguiente código:



El directorio **/módulos** queda así:

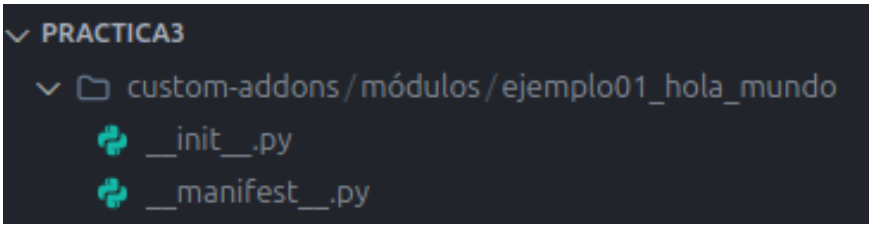
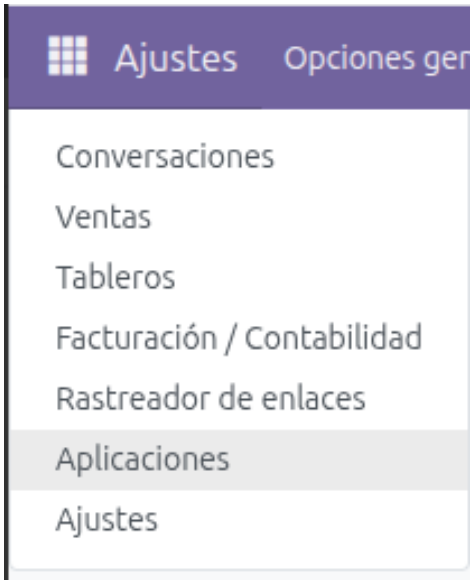
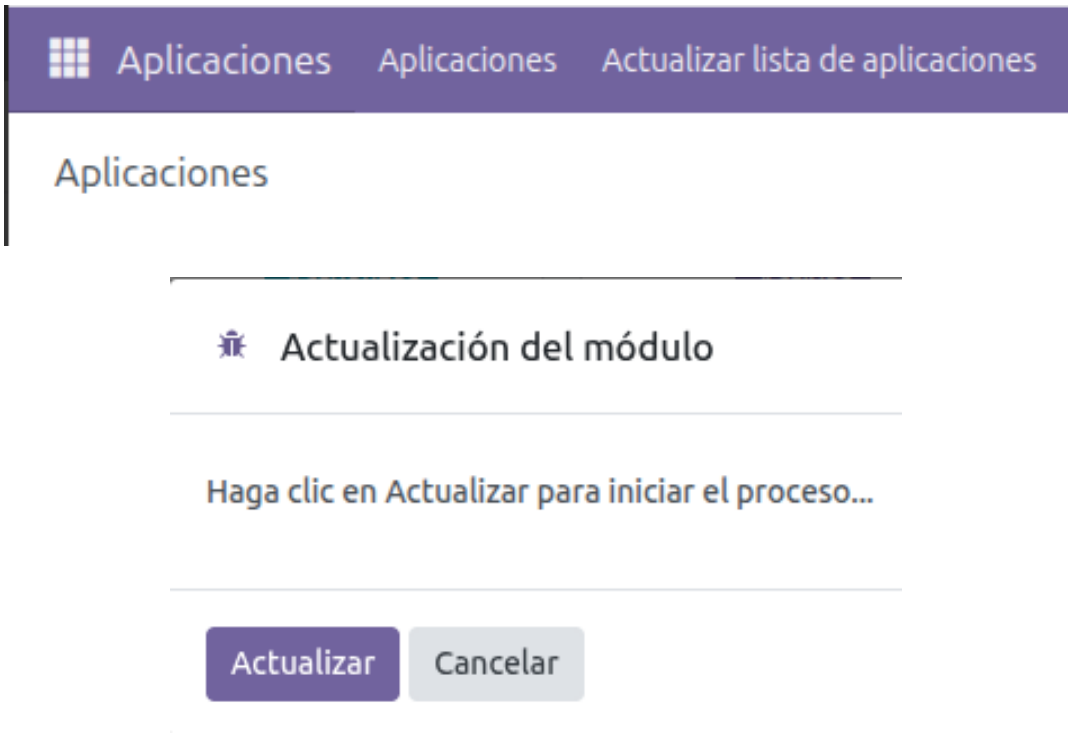


Figure 5: ficheros

Una vez creada la estructura, con el "Modo desarrollador" activado, se actualizará la liasta de aplicaciones.

1. Entrar en el menú de Odoo e ir a la opción de **Aplicaciones**:
2. Ir a **Actualizar lista de aplicaciones**:
3. Confirmar que **sí** se quiere actualizar:
4. Muestra el módulo **"Hola mundo"**:





5 Creación de módulos en Odoo

Crear un módulo en Odoo y establecer una aplicación propia integrada dentro de un ecosistema más amplio. Considerar cada módulo como una mini aplicación que forma parte de una aplicación madre mayor: Odoo. Diseñar el módulo de manera que funcione con independencia del resto, definiendo su propia lógica, modelos de datos, vistas, controladores y funcionalidades específicas. Aprovechar la estructura modular de Odoo para garantizar que cada componente opere de forma autónoma, manteniendo la coherencia del sistema general.

Desarrollar cada módulo como una aplicación dentro de otra aplicación, asegurando autonomía y flexibilidad en su diseño. Utilizar esta organización modular para ampliar Odoo de manera escalable y personalizada, ajustando las funciones a las necesidades particulares de cada entorno

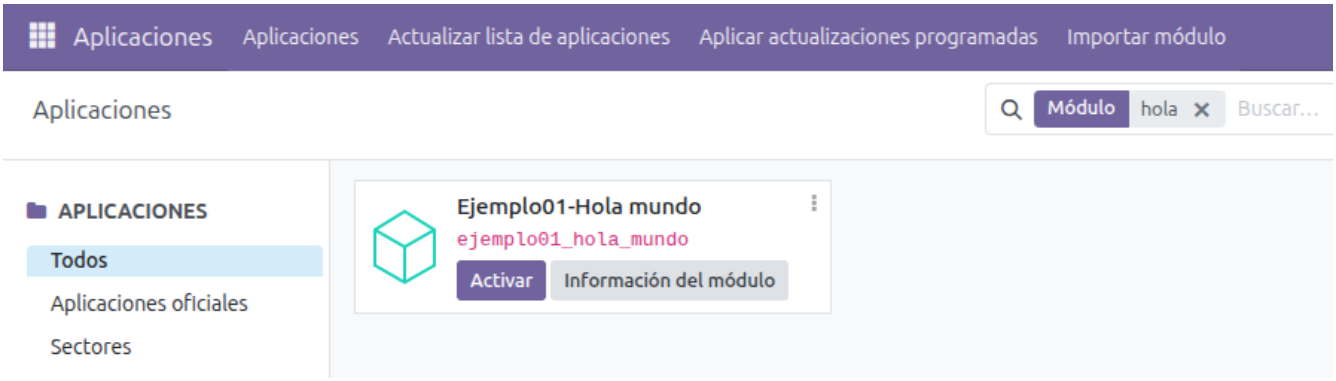


Figure 6: Aparición del módulo Hola mundo

sin alterar la base del sistema. Mantener la integración fluida entre módulos para que cada uno actúe como una pieza esencial dentro de un conjunto perfectamente engranado, reforzando así la adaptabilidad y el rendimiento global de la plataforma.

5.1 Creación de módulos con Odoo Scaffold

Utilizar el comando `odoo scaffold` para generar la estructura base de un nuevo módulo dentro del entorno de desarrollo de Odoo. Este comando permite crear automáticamente todos los archivos y carpetas necesarias para iniciar un módulo funcional, siguiendo la arquitectura estándar de Odoo. Para ello, introducir el comando:

```
odoo scaffold "nombre del modulo" "ruta"
```

Antes de ese paso, hay que instalar Odoo 16 por terminal, en este caso con el comando:

sudo apt install odoo-16

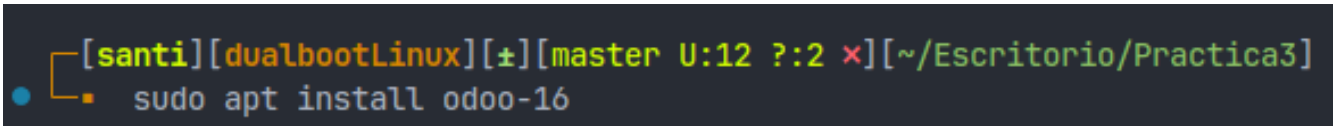


Figure 7: Instalación de Odoo 16

En este caso se crearán 5 módulos nuevos, orientados a la educación, en concreto a módulos enfocados en la organización personal del profesorado:

- agenda_profesor
- clases
- asistencia_alumnos
- calendario
- registro_notas

Para ello se utilizan los siguientes comandos:

```
┌─[santi][dualbootLinux][±][master U:9 ?:3 ×][~/Escritorio/Practica3]
└─▪  ~/odoo/odoo-bin scaffold agenda_profesor ~/Escritorio/Practica3/custom-addons
~/odoo/odoo-bin scaffold clases ~/Escritorio/Practica3/custom-addons
~/odoo/odoo-bin scaffold asistencia_alumnos ~/Escritorio/Practica3/custom-addons
~/odoo/odoo-bin scaffold calendario ~/Escritorio/Practica3/custom-addons
~/odoo/odoo-bin scaffold registro_notas ~/Escritorio/Practica3/custom-addons
```

Figure 8: código de creación de módulos con scaffold

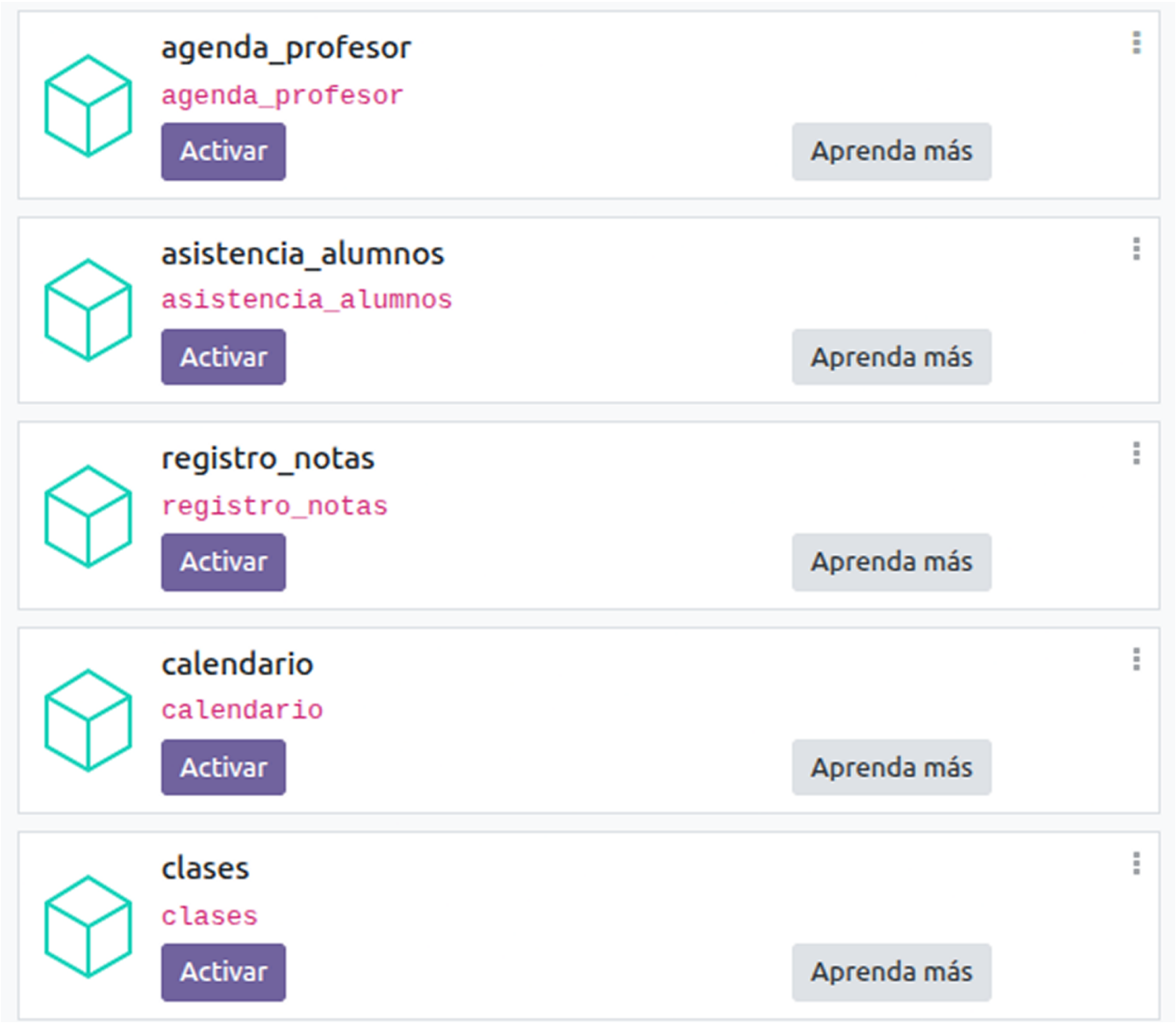


Figure 9: Contenido de cada módulo

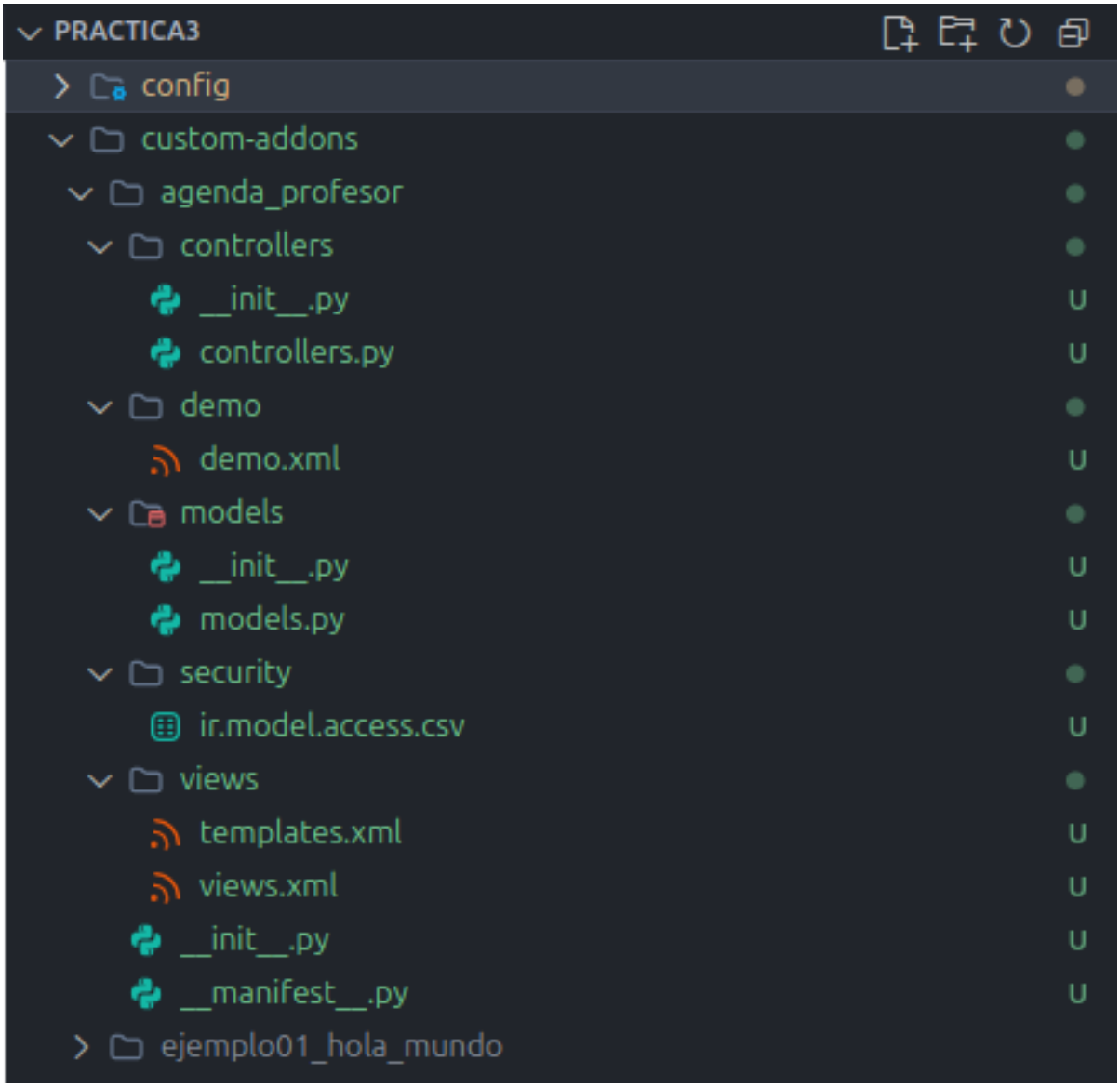


Figure 10: Contenido de cada módulo

Breve explicación de cada fichero generado en el interior de cada uno de los módulos:

- **models/models.py**: define un ejemplo del modelo de datos y sus campos.
- **views/views.xml**: describe las vistas de nuestro módulo (formulario, árbol, menús, etc.).
- **demo/demo.xml**: incluye datos "demo" para el ejemplo propuesto de modelo.
- **controllers/controllers.py**: contiene un ejemplo de controlador de rutas, implementando algunas rutas.
- **views/templates.xml**: contiene dos ejemplos de vistas "qweb" usado por el controlador de rutas.
- **__manifest__.py**: es el manifiesto del módulo. Incluye información como el título, descripción, así como ficheros a cargar.

6 Módulo funcional: ”Lista de tareas”

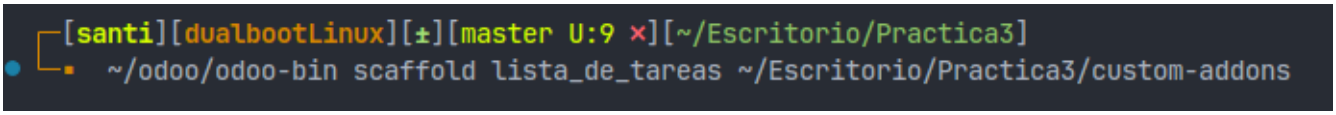


Figure 11: Creación de Lista de tareas

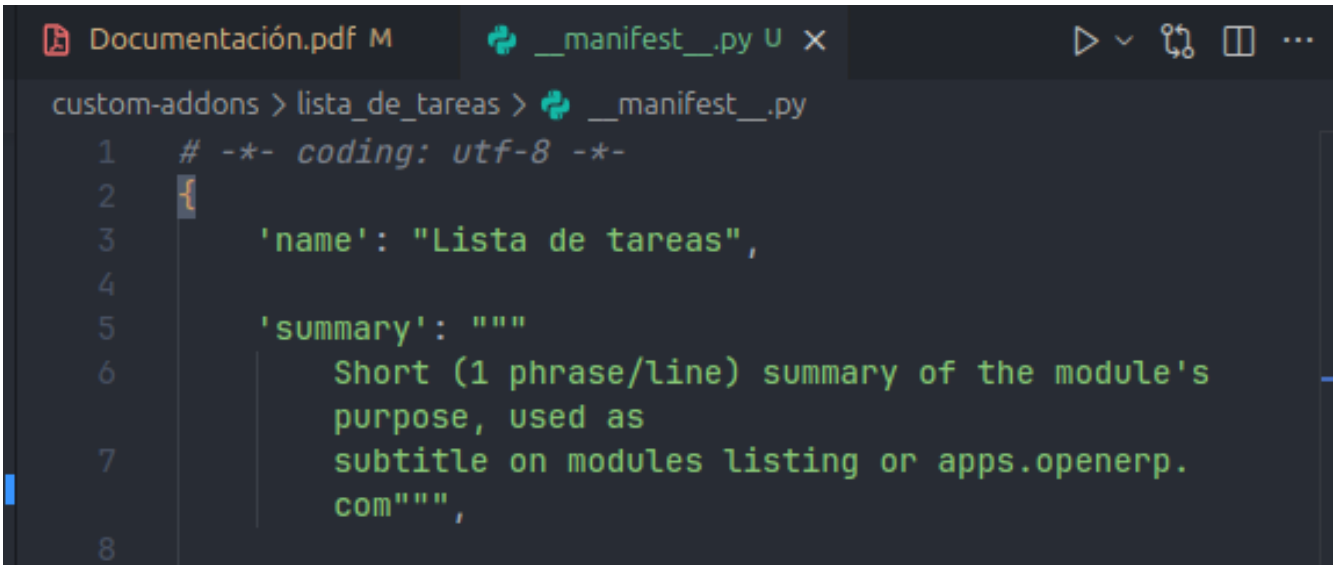


Figure 12: Cambiar el nombre a **Lista de tareas** sin _

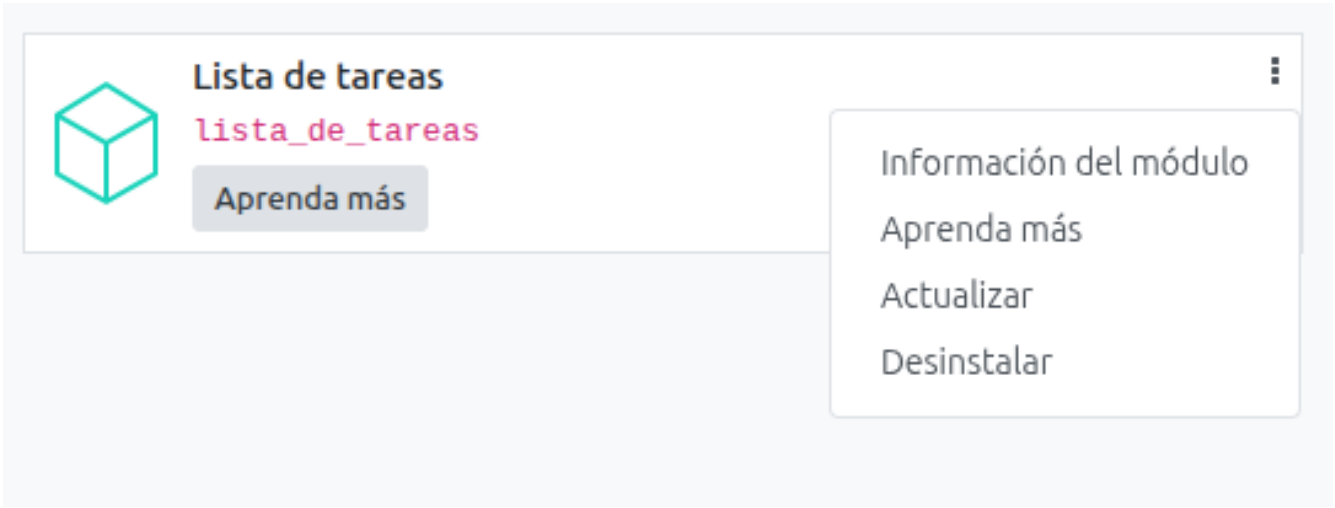


Figure 13: Módulo en Odoo

7 Modificación del módulo "Lista de tareas"

Este módulo permite crear y gestionar tareas en Odoo de forma sencilla. Cada tarea tiene un título, descripción, un valor numérico, un cálculo automático y un estado de completada.

7.1 Modelo de Datos

Documentación.tex M | models.py U | modelo1.png U | odoo.conf M

custom-addons > lista_de_tareas > models > models.py > ...

```python
# -*- coding: utf-8 -*-

from odoo import models, fields, api

class ListaDeTareas(models.Model):
    _name = 'lista.tareas'
    _description = 'Lista de Tareas'

    nombre = fields.Char(string="Título", required=True)
    descripcion = fields.Text(string="Descripción")
    valor = fields.Integer(string="Valor")
    valor2 = fields.Float(string="Valor en %", compute="_calcular_valor", store=True)
    completada = fields.Boolean(string="Completada", default=False)

    @api.depends('valor')
    def _calcular_valor(self):
        for record in self:
            record.valor2 = record.valor / 100 if record.valor else 0
```

Figure 14: Módulo en Lista de tareas

7.2 Explicación de Campos

- **nombre**: Título de la tarea, obligatorio.
- **descripcion**: Descripción o detalles de la tarea.
- **valor**: Número que se asigna a la tarea.
- **valor2**: Calculado automáticamente como value/100.
- **completada**: Indica si la tarea está completada.

Para poder aplicar estos cambios al módulo Lista de tareas, es necesario reiniciar los contenedores y en los 3 puntos del módulo en Odoo y darle click en **Actualizar**. Esto hará que todos esos cambios en el módulo sean vigentes.

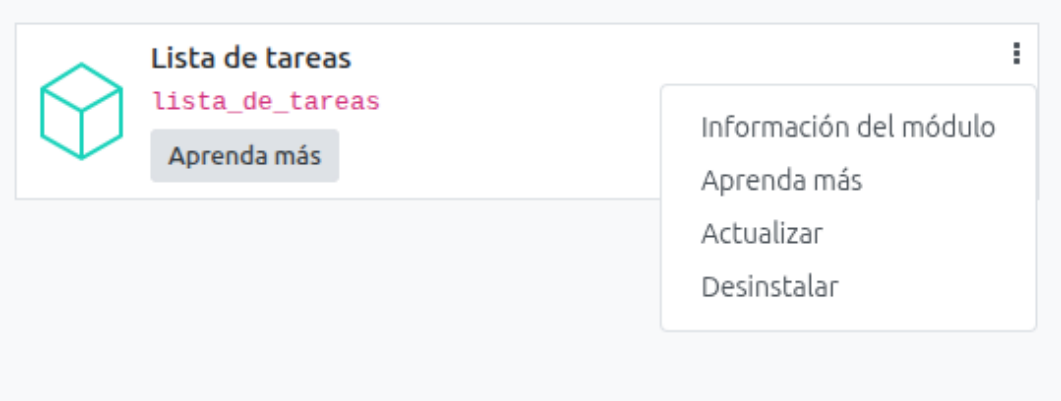


Figure 15: Módulo en Lista de tareas

8 Conclusiones