

Sistemas de Gestión Empresarial

Práctica 3

Primeros módulos en Odoo

Autor:

Santi Martínez

November 16, 2025

ÍNDICE

1	Introducción	2
2	Activar del modo desarrollador en Odoo	3
2.1	Entrar en Odoo con el usuario administrador	4
2.2	Activar modo desarrollador	5
3	Primer módulo: "Hola mundo"	6
4	Creación de módulos en Odoo	8
4.1	Creación de módulos con Odoo Scaffold	8
5	Módulo funcional: "Lista de tareas"	11
5.1	Creación del módulo	11
5.2	Modelo de Datos	12
5.3	Explicación de Campos	12
6	Modificación del módulo "Lista de tareas"	13
6.1	Actualización del módulo	13
6.1.1	Métodos de Acción	13
6.2	Explicación concreta de la actualización	14
6.3	Aplicar cambios	14
6.4	Diferencias entre el original y la actualización	14
6.5	Uso práctico del módulo	15
6.5.1	Creación de nueva tarea	15
6.5.2	Vista de la lista de tareas pendientes	16
7	Problemas surgidos	17
7.1	yourcompany.com	17
7.2	Métodos en el models.py y modificar views.xml	17
8	Conclusión	18
9	Bibliografía	19

1 Introducción

Esta documentación desarrolla la **Práctica 3: Primeros módulos en Odoo**, cuyo objetivo principal es comprender el proceso de creación, instalación y personalización de módulos dentro del entorno de desarrollo Odoo.

Esta práctica tendrá lugar en un nuevo servidor de Odoo creado con docker compose, el cual de desglosará más adelante para poder ver bien cada una de sus partes y la creación de las mismas.



El propósito fundamental es adquirir las habilidades necesarias para configurar y actualizar módulos en Odoo, partiendo de ejemplos básicos como “Hola Mundo” y “Lista de tareas”. En este último se le va a realizar además una actualización para implementarle varias funcionalidades añadidas.

Esta práctica está vinculada a un repositorio de GitHub (https://github.com/santimartinezgb/odoo_practica3-santi-martinez), a través del cual se podrán observar los avances.



2 Activar del modo desarrollador en Odoo

Modo desarrollador: desbloquea acceso a herramientas y ajustes avanzados 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.
- Abrir **Ajustes**
- Bajar a la sección **Herramientas de desarrollador**
- Click en **Activar modo desarrollador** (El cual solo será visible si hay al menos un módulo instalado)
- Una vez activado, la opción **Desactivar el modo desarrollador** se vuelve disponible.

2. Desde la URL:

- Para activar el modo de desarrollador desde cualquier parte de la base de datos agregar **?debug=1** a la URL después de /web.
- Para desactivarlo, use **?debug=0**.
- Usar **?debug=assets** el modo de desarrollador con activos y **?debug=tests** para activarlo con activos de prueba.

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/hmdmhiilocobgohohpdpolnhl=es_PR

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

El **modo desarrollador** en Odoo permite acceder y modificar herramientas avanzadas, activar ajustes internos y crear o depurar módulos mediante interfaz, URL o extensiones.

2.1 Entrar en Odoo con el usuario administrador

Quiere decir iniciar sesión en Odoo usando una cuenta que tenga **privilegios completos**, por lo que el usuario puede:

- Ver y modificar todas las configuraciones
- Instalar módulos
- Gestionar permisos de otros usuarios

Este paso es necesario porque algunas opciones avanzadas, como el modo desarrollador, solo están disponibles para cuentas con permisos administrativos, ya que permiten cambiar la estructura interna y la configuración del sistema.

En otras palabras, es como abrir una aplicación con una cuenta “superusuario” para poder acceder a todo lo que normalmente los usuarios normales no pueden tocar.

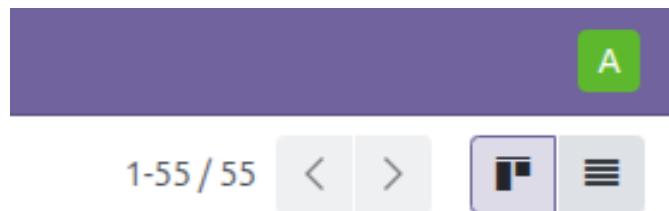


Figure 1: Odoo con usuario administrador

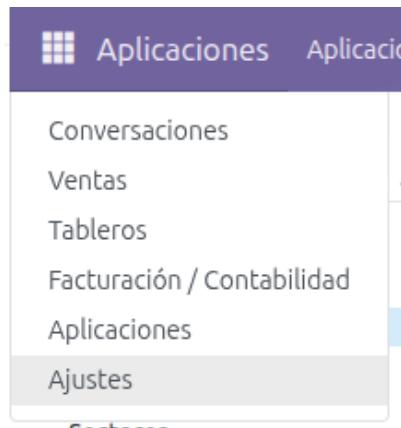
Se puede comprobar que se está dentro de este usuario administrador con la imagen anterior, en la que arriba a la derecha de la pantalla principal de Odoo aparece un cuadro verde con una A de administrador.

Entrar con el **usuario administrador** en Odoo permite acceder a todas las configuraciones, instalar módulos y activar opciones avanzadas como modo desarrollador.

2.2 Activar modo desarrollador

Para activar el modo desarrollador en Odoo, se seguirán estos pasos:

1. Hacer clic en el **menú cuadrado** ubicado en la esquina superior izquierda de la interfaz
2. Seleccionar la opción **Ajustes** del menú desplegable



3. Desplazarse hacia abajo en la página hasta localizar la sección **Herramientas de desarrollador**
4. Hacer clic en el botón **Activar modo desarrollador**

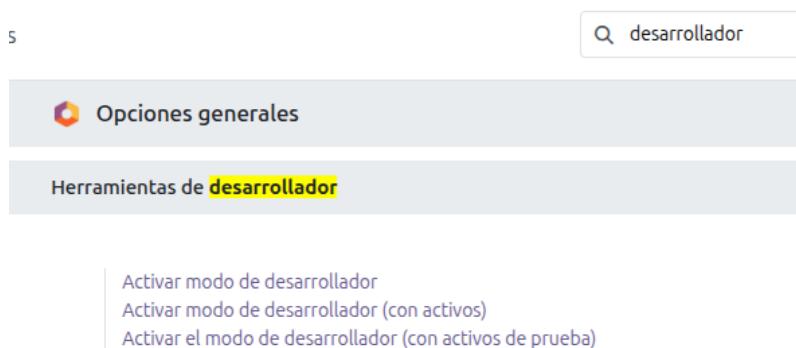


Figure 2: Opción para activar el modo desarrollador

3 Primer módulo: "Hola mundo"

El objetivo de este primer módulo es el comprobar que Odoo detecta 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 **/custom-addons**, el cual se genera automáticamente:

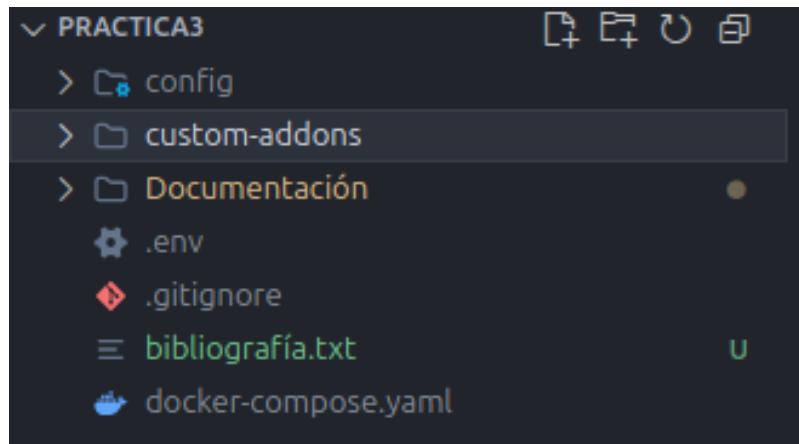
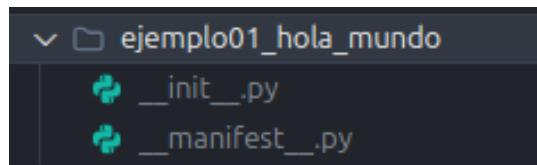


Figure 3: Directorio /custom-addons



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

- **_init_.py**. El cual estará vacío
- **_manifest_.py**. Es el archivo principal de cada módulo; esto quiere decir que proporciona los datos básicos del módulo y le da su espacio dentro de la aplicación (como por ejemplo, de qué categoría va a ser; en el caso de Lista de tareas, creado más adelante de la práctica, tiene categoría Productivity) Contiene el siguiente código:

```
módulos > _manifest_.py
1  # -*- coding: utf-8 -*-
2  {'name': 'Ejemplo01-Hola mundo'}
```

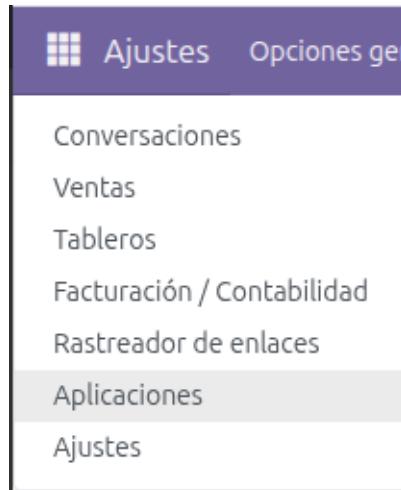
The image shows a code editor window with a dark theme. The file '_manifest_.py' is open. The code inside the file is:

```
# -*- coding: utf-8 -*-
{'name': 'Ejemplo01-Hola mundo'}
```

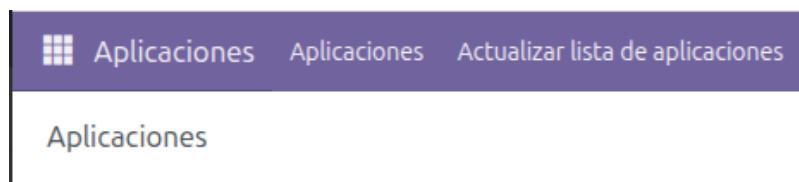
Una vez creada la estructura del módulo, con el **Modo desarrollador activado**, toca ir hacia arriba a la izquierda de la pantalla y clicar en el menú.

Una vez se esté visualizando el menú, seguir los siguientes pasos:

1. Entrar en el menú de Odoo e ir a la opción de **Aplicaciones**:



2. Clicar en **Actualizar lista de aplicaciones**:



3. Confirmar que **sí** se quiere actualizar la lista de aplicaciones:

4. Si todo se ha hecho correctamente, al escribir en el buscador 'Hola mundo', se mostrará en pantalla el módulo de ejemplo **"Hola mundo"**:

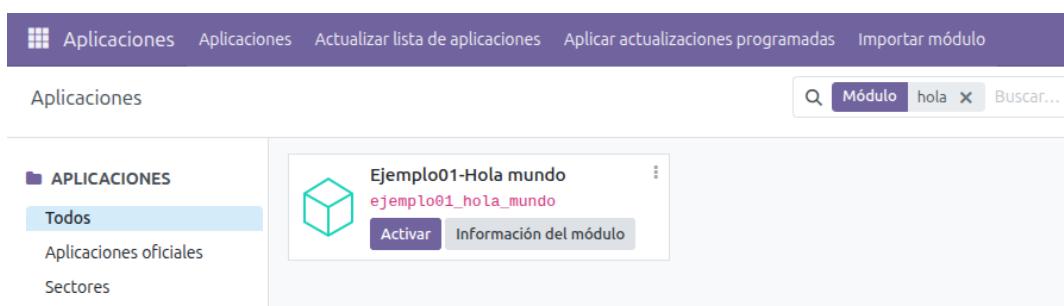


Figure 4: Aparición del módulo Hola mundo

4 Creación de módulos en Odoo

Crear un módulo en Odoo significa desarrollar una extensión o componente adicional que añade, modifica o personaliza las funciones existentes del sistema. Odoo está construido de forma modular, lo que permite que cada parte del sistema (como ventas, inventario, contabilidad o recursos humanos) sean un módulo independiente cada uno.

El proceso de creación de un módulo comienza definiendo su estructura básica, y esto comienza con un archivo principal llamado `_manifest_.py`, que describe sus características, dependencias y los elementos que lo componen.

Este enfoque modular hace que Odoo es altamente personalizable, con lo que se pueden desarrollar desde pequeñas mejoras hasta aplicaciones empresariales completas.

4.1 Creación de módulos con Odoo Scaffold

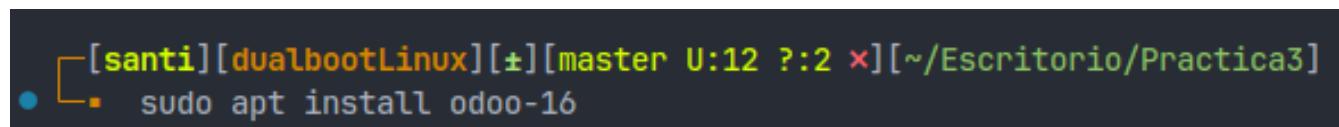
Utilizar el comando `odooscaffold` se utiliza para generar la estructura base de un nuevo módulo dentro del entorno de desarrollo de Odoo.

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:

```
odooscaffold "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
```



```
[santi][dualbootLinux][±][master U:12 ?:2 ✘][~/Escritorio/Practica3]
● └─ sudo apt install odoo-16
```

Figure 5: 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 6: código de creación de módulos con scaffold

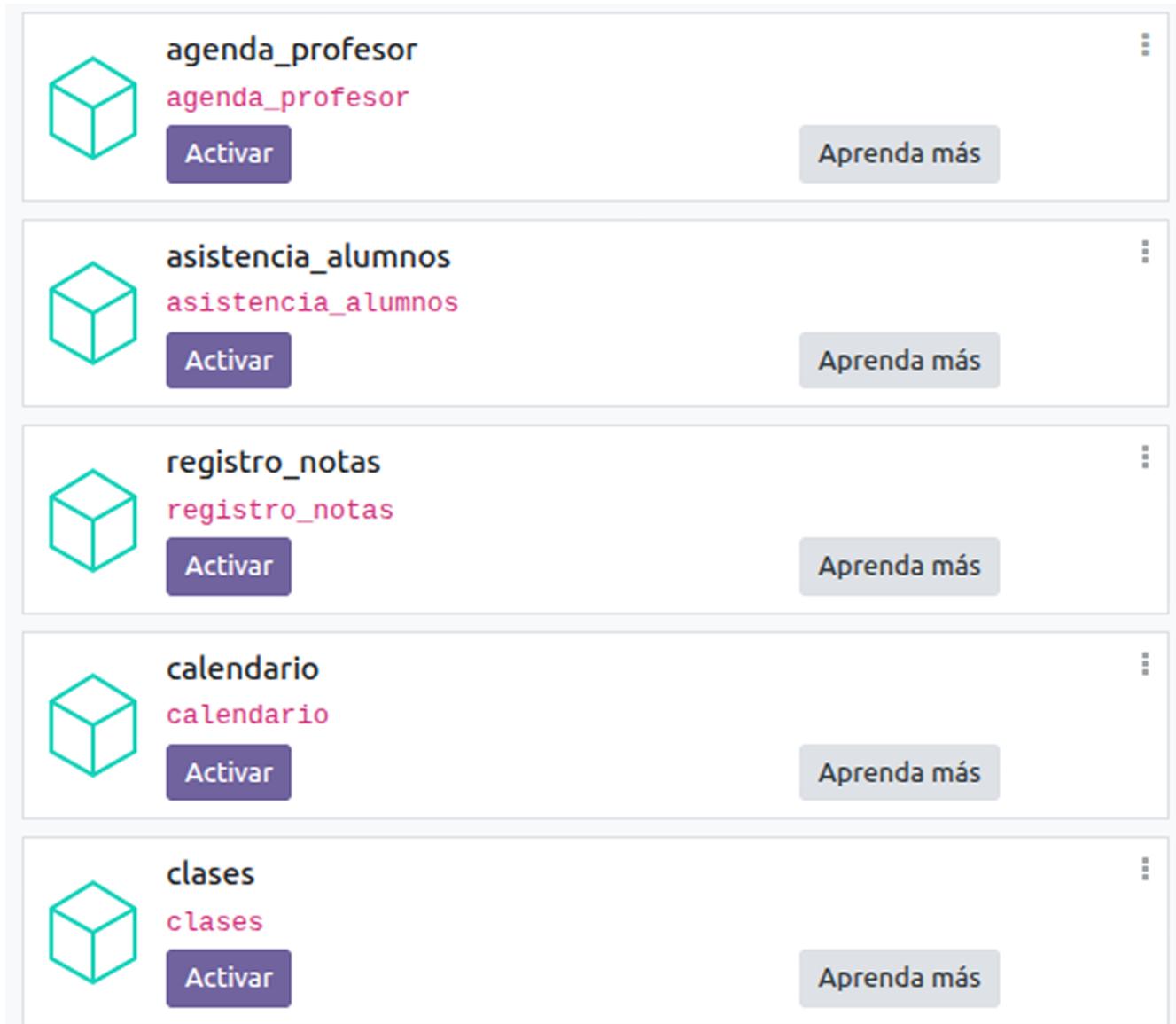


Figure 7: Módulos en Odoo

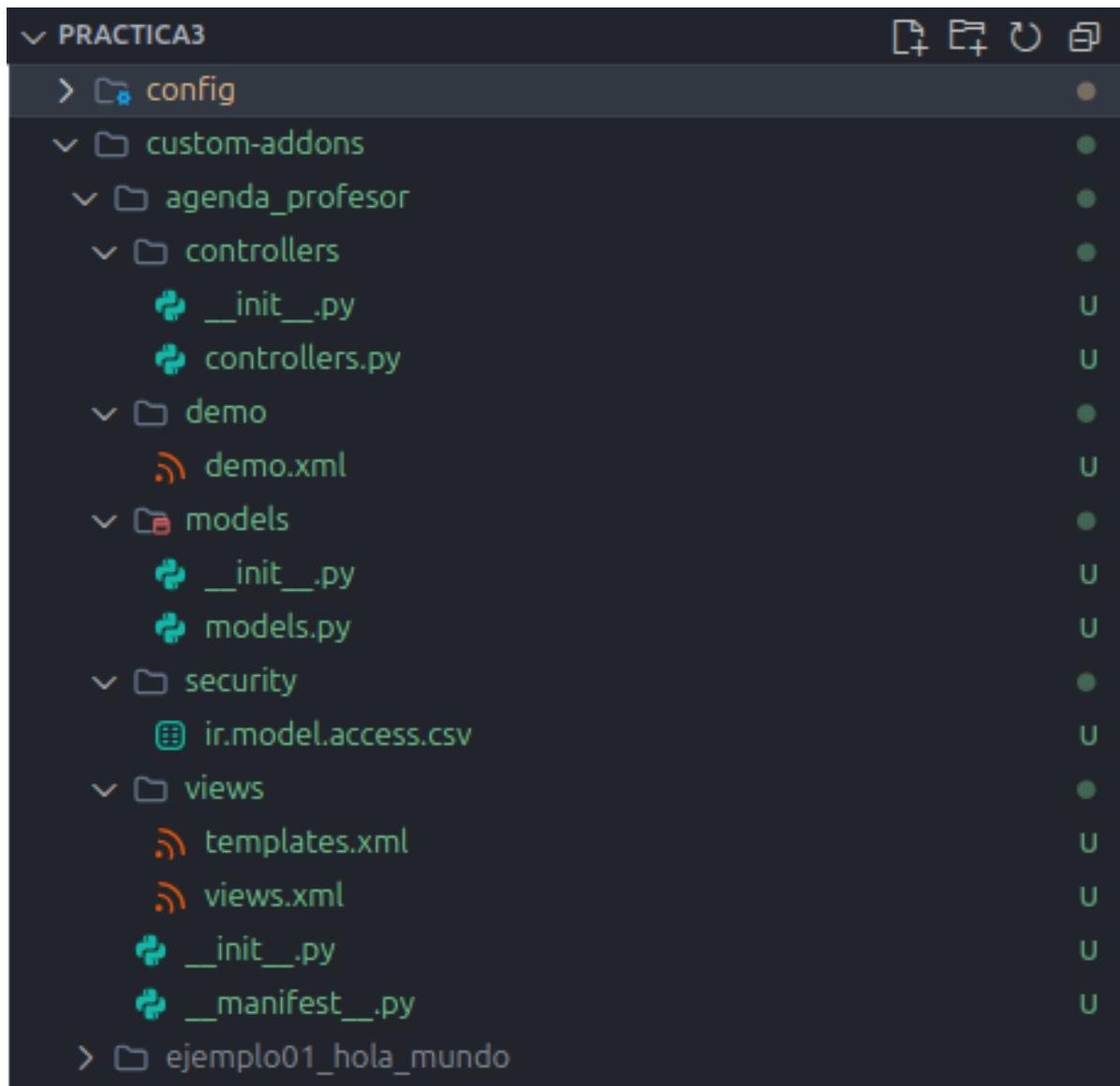


Figure 8: Contenido de cada módulo

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.

5 Módulo funcional: "Lista de tareas"

Este módulo permite **crear y gestionar tareas** en Odoo de forma sencilla, ofreciendo una herramienta práctica para organizar y dar seguimiento a las tareas dentro de una empresa o equipo de trabajo.

El sistema permite mantener un registro estructurado de todas las tareas creadas, facilitando la planificación y la priorización del trabajo diario, además de mejorar la comunicación entre los miembros del equipo.

Cada tarea cuenta como identificador personal con:

- **Título:** Nombre de la tarea
- **Descripción:** Descripción de la tarea

Además, cada tarea posee un estado de **completada, pendiente o en proceso**.

Este módulo permite crear, gestionar y controlar tareas en Odoo con campos de título, descripción, valor numérico, cálculo automático y estado, mejorando la organización, seguimiento y productividad del trabajo.

5.1 Creación del módulo

Para este caso, la creación del módulo se realiza con el propio scaffold, visto anteriormente, para facilitar y automatizar la creación de subdirectorios.

```
[santi][dualbootLinux][±][master U:9 ✘][~/Escritorio/Practica3]
● └─ ~/odoo/odoo-bin scaffold lista_de_tareas ~/Escritorio/Practica3/custom-addons
```

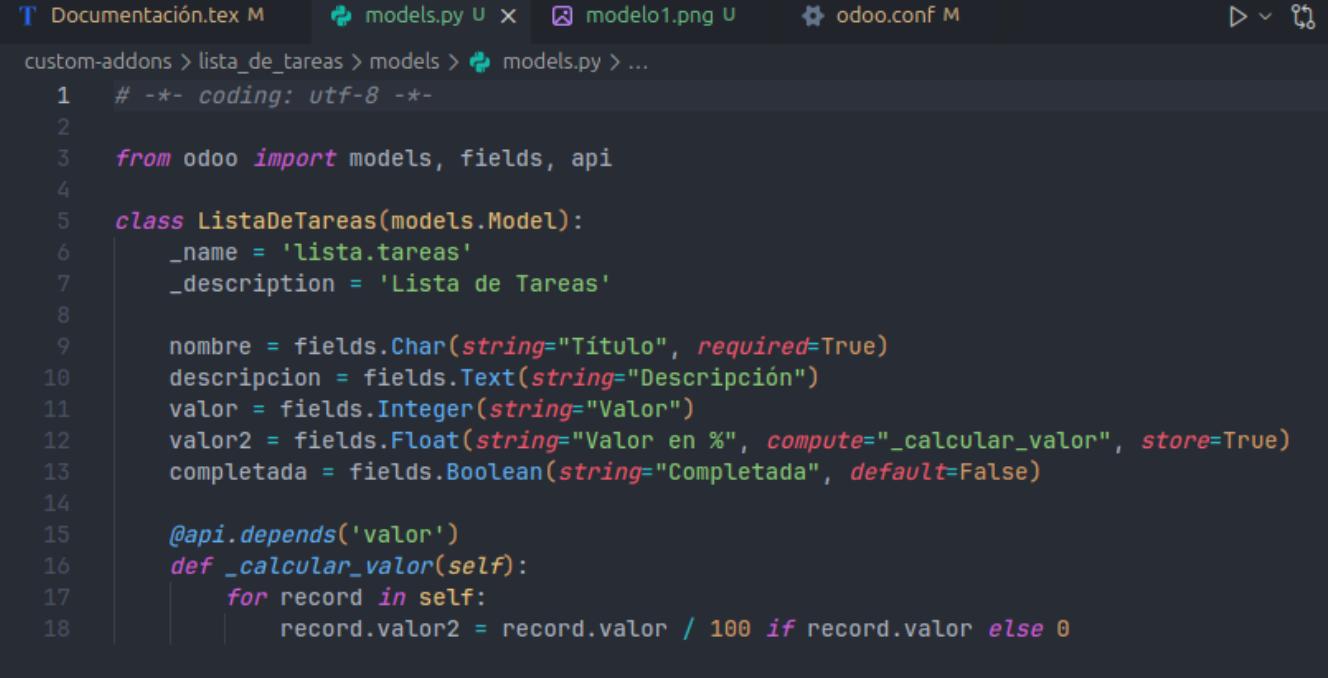
Figure 9: Creación de Lista de tareas

5.2 Modelo de Datos

El archivo `_models_.py` funciona como una tabla para almacenar las tareas creadas con su información básica.

La función `_calcular_valor` se encargaba de transformar el valor numérico en porcentaje; encargaba en pasado porque con la actualización de este módulo que se explica más adelante, esa opción quedó

`@api.depends('valor')` es un método que le dice a Odoo que solo recalule el porcentaje si el valor original cambia, evitando así errores y haciendo que el sistema sea más fiable.



The screenshot shows a code editor with the following details:

- File tabs: Documentación.tex M, models.py U, modelo1.png U, odooconf M.
- File path: custom-addons > lista_de_tareas > models > models.py > ...
- Code content:

```
1  # -*- coding: utf-8 -*-
2
3  from odoo import models, fields, api
4
5  class ListaDeTareas(models.Model):
6      _name = 'lista.tareas'
7      _description = 'Lista de Tareas'
8
9      nombre = fields.Char(string="Título", required=True)
10     descripción = fields.Text(string="Descripción")
11     valor = fields.Integer(string="Valor")
12     valor2 = fields.Float(string="Valor en %", compute="_calcular_valor", store=True)
13     completada = fields.Boolean(string="Completada", default=False)
14
15     @api.depends('valor')
16     def _calcular_valor(self):
17         for record in self:
18             record.valor2 = record.valor / 100 if record.valor else 0
```

Figure 10: Módulo en Lista de tareas

Este modelo permite al módulo Lista de tareas: crear, almacenar y hacer seguimiento de tareas con cálculos automáticos dentro de Odoo.

5.3 Explicación de Campos

- **nombre:** Título de la tarea, obligatorio
- **descripcion:** Descripción o detalles de la tarea
- **valor:** Es el valor numérico que se le da a la tarea
- **valor2:** Calcula (con el método `_calcular_valor`, mencionado anteriormente) el porcentaje del valor
- **completada:** Indica si la tarea está completada, por defecto es False

6 Modificación del módulo ”Lista de tareas”

6.1 Actualización del módulo

El módulo da al usuario funcionalidades para crear, actualizar y gestionar tareas. Este proporciona los siguientes campos para cada tarea y hacer dinámica y agradable la gestión de estas tareas:

- **Nombre y Descripción:** Es el título de la tarea. El campo **descripcion** otorga una explicación un poco más amplia de lo que es la tarea.
- **Valores Numéricos:** Son números asociados a cada tarea. Este campo usa el parámetro `store=True`, lo que hace que el valor persista en la base de datos, es decir, un guardado permanente.
- **Estado:** Implementa una selección con cuatro opciones posibles: pendiente, en progreso, completada y cancelada. El valor por predeterminado es 'pendiente'. Este campo determina el estado de la tarea.
- **Indicador de completado:** Es un campo booleano que quiere decir al usuario si la tarea está (true) o no (false) completada.
- **Color:** Según el color en el que se muestre, pretende representar un estado diferente la tarea.

6.1.1 Métodos de Acción

Son métodos que actualizan el estado de la tarea:

- **action_iniciar:** Actualiza el estado de la tarea a 'en_progreso'. Este método se pone cuando el usuario quiere decir que está realizando la tarea, pero sin haberla terminado.
- **action_completar:** Cambia el estado de la tarea a 'completada' y el campo **completada** lo pasa a true.
- **action_cancelar:** Modifica el estado a 'cancelada'. Este método permite al usuario eliminar/cancelar tareas.
- **action_reabrir:** Restablece una tarea al estado 'pendiente' y pone el campo **completada** en false.

6.2 Explicación concreta de la actualización

El módulo permite **crear, actualizar y gestionar tareas**, con campos como nombre, descripción, valores numéricos, estado y color. Incluye métodos para valores y colores, métodos de acción para cambiar estados y opción de etiquetado para clasificar tareas, facilitando su seguimiento y búsqueda.

6.3 Aplicar cambios

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.

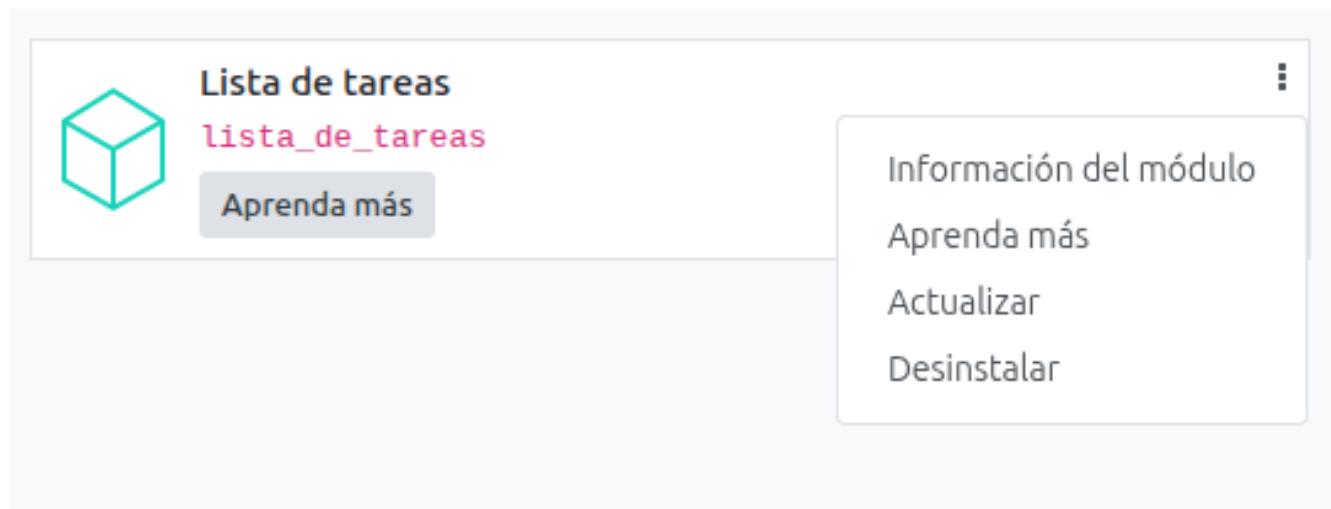


Figure 11: Módulo en Lista de tareas

6.4 Diferencias entre el original y la actualización

El original es un modelo básico, solo para guardar datos de tareas y calcular un valor porcentual, sin ninguna gestión adicional ni interacción en la interfaz.

La actualización es un módulo completo y funcional, pensado para gestionar tareas con estados, colores, etiquetas y acciones.

6.5 Uso práctico del módulo

6.5.1 Creación de nueva tarea

El módulo da al usuario la capacidad de crear **nuevas tareas** de manera sencilla y rápida. Una vez creadas, estas tareas se almacenan y se muestran en una lista, lo que permite visualizarlas de forma clara y tener un control completo sobre ellas.

Esta pantalla distingue entre los siguientes campos:

- Título de la nueva tarea
- Descripción de la tarea
- Estado
 - pendiente
 - completada
 - en progreso
 - cancelada
- Casilla para indicar que está completada

The screenshot shows a user interface for creating a new task. At the top, there is a purple header bar with a grid icon on the left and a square icon on the right. Below the header, there is a toolbar with buttons labeled "Nuevo" (New), a back arrow, a gear icon, a cloud icon, and a refresh/circular arrow icon. The main form area has the following fields:

- Título:** Ejemplo de nueva tarea para la documentación
- Descripción:** (This field is currently empty.)
- Estado:** Pendiente
- Completada:** (This checkbox is empty.)

Figure 12: Creación de tarea

6.5.2 Vista de la lista de tareas pendientes

Esta es una vista general de todas las tareas creadas por el usuario, y su propósito es permitir que cualquier persona pueda revisar de manera rápida qué trabajos tiene por hacer y cuáles ya fueron completados.

En esta pantalla se muestran todas las tareas sin importar su estado. Además, esta vista ayuda a mantener un control más organizado del trabajo diario.

Título	Estado	Completada
Corrección exámenes 1º DAM	Pendiente	<input type="checkbox"/>
Evaluación de 2º DAM	Pendiente	<input type="checkbox"/>
Evaluación 1º DAM	Pendiente	<input type="checkbox"/>
Reunión sobre prácticas 2º DAM	Completada	<input checked="" type="checkbox"/>
Reunión nuevos alumnos	Pendiente	<input type="checkbox"/>

Figure 13: Vista con alguna tarea completada

La **idea principal** de esta vista es ofrecer al usuario una herramienta sencilla, clara y accesible, donde pueda comprobar el avance de sus actividades y detectar qué asuntos requieren atención inmediata.

Gracias a esta vista general, el usuario puede decidir con prioridad qué tarea realizar primero, cuáles se pueden posponer y cuáles ya están finalizadas.

7 Problemas surgidos

7.1 yourcompany.com

Al principio el módulo lista de tareas no hacía más que enviarme a yourcompany.com y no sabía el por qué. Revisando el código me di cuenta en el `__manifest__.py` que aparecía:

`'views/templates.xml'`,

Ese archivo viene del módulo generado por el scaffolding de Odoo, e incluye la página de ejemplo de yourcompany.com, por eso me entraba ahí solamente.

Ese `templates.xml` no debería estar en un módulo que solo muestra listas y formularios.

Cuando Odoo detecta un archivo `templates.xml` con rutas web, puede redirigir al usuario a una página en blanco o colgada, como fue el caso de yourcompany.com. Así que comenté esa línea de código y listo.

7.2 Métodos en el `models.py` y modificar `views.xml`

El tema de métodos dentro del módulo Lista de tareas fue algo que, no sé si evitable o inevitablemente, tuve que apoyarme en vídeos de youtube donde hacían algo muy parecido, en concreto en este vídeo <https://www.youtube.com/watch?v=VMfm-CvigXM> + ciertos prompts en AI para conseguir el funcionamiento.

Esa parte me resultó bastante tediosa de realizar; principalmente porque hasta ya avanzado el trabajo (y verlo casi construido) no visualicé correctamente el esquema mental de cómo se construyen estos módulos. Entiendo que todo eso puede ser porque es la primera vez que realizo este tipo de actividad.

8 Conclusión

Esta práctica me ha ayudado a entender bastante el funcionamiento modular que tiene Odoo, pese a que tanto al principio como en proceso de la práctica no entendiese perfectamente cada funcionamiento.

La propia documentación de odoo <https://www.odoo.com/documentation/15.0/es/index.html>, me ha parecido muy completa para todo tipo de pequeña duda que aparece a medida llevas a cabo los procesos de creación y modificación de módulos; también para la documentación de los mismos procesos, la cual viene adecuadamente explicada.

Sobre el lenguaje que manejan los módulos de Odoo, **python**, pese al escaso tiempo que llevo trabajando con él, me parece un lenguaje agradable de leer y aprender; pero ahora mismo para ciertas cosas como `@api.depends` o la creación de ciertos tipos de variables, se me haga pesado por mero desconocimiento.

Lo que me sigue dando problemas (los cuales no incluí en la documentación por que son más de índole personal) es el volver a levantar el servidor con docker-compose. Sin tocar dada, el servidor me da problemas que no estaban la última vez que estuve trabajando con ello, lo cual en este caso no conseguí arreglar y tuve que decantarme por montar otro de nuevo para esta práctica. El trabajar más con docker para mejorar mi capacidad de análisis sobre los problemas que me aparezcan, es algo que tengo que empezar a hacer ya mismo.

Respecto a odoo, sigo viéndolo como algo práctico, pero me recuerda mucho a mi experiencia pasada trabajando en temas administrativos puros, es algo que no me levanto de la cama y pienso: ¡Qué maravilla que hoy toca gestionar datos en Odoo!.

Como conclusión final, decir que con lo que me quedo a nivel gustos es el objetivo de seguir aprendiendo bien docker y la opinión personal de que creo que estoy mejorando a la hora de documentar y presentar trabajos, lo que va de la mano a que también le esté cogiendo algo de gusto.

9 Bibliografía

- **Uso del scaffold:** <https://stackoverflow.com/questions/24385750/odoo-scaffolding>
- **Documentación oficial de Odoo:** <https://www.odoo.com/documentation/>
- **Documentación Odoo (modo desarrollador):** https://www.odoo.com/documentation/15.0/es/applications/general/developer_mode.html
- **Crear una portada bonita en Latex:** <https://manualdelatex.com/tutoriales/crear-una-portada-bonita-en-latex/>
- **Documentación Odoo (creación de módulo nuevo):** https://www.odoo.com/documentation/15.0/es/administration/odoo_sh/getting_started/first_module.html
- **Documentación Odoo (actualización de un módulo):** <https://www.youtube.com/watch?v=T2EQu8qlvXE>
- **Para xml:** <https://www.odoo.com/documentation/15.0/developer/reference/backend/views.html>
- **Herencia de modelos en Odoo:** <https://www.cybrosys.com/blog/model-inheritance-in-odoo>
- **Cuadro de colores en LaTeX:** <https://ondahostil.wordpress.com/2017/05/17/lo-que-he-aprendido-en-este-mes-de-mayo/>
- **ClaudeAI (apoyo en la actualización del módulo Lista de tareas):** <https://claude.ai/>