Documentación Técnica: Creación de módulos de Odoo

Autor: Pedro José Meixús Belsol

9 de diciembre de 2025

Índice general

1. Implantación módulo basico **2**
1.1. Creación y funcionamiento 2

2. Implantación del Primer módulo **3**
2.1. Creación y funcionamiento 3
2.2. Problemas y Errores 4
2.2.1. Tabulaciones 4
2.2.2. Trees 5

3. Modificación del Primer módulo **6**
3.1. Mejoras propuestas 6
3.2. Creación y funcionamiento 6

Capítulo 1

Implantación módulo basico

1.1. Creación y funcionamiento

Empezaremos con la creación de un módulo 'Hola mundo' que nos servirá para comprobar que tenemos mapeadas correctamente las carpetas.

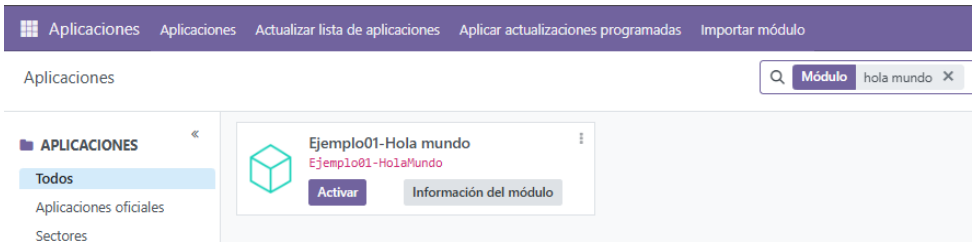
Primero crearemos el archivo **'__init__.py'**, el cuál dejaremos vacío en este caso.

Por otro parte también crearemos el archivo **'__manifest__.py'**, en el que meteremos el siguiente código:

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

Código 1.1: Contenido de __manifest__.py

Con esto, tendremos un módulo vacío, y que ahora nos debería aparecer en los módulos de Odoo. Vamos a comprobarlo:



Si nos sale disponible en la búsqueda y con el nombre que le hemos puesto es que lo hemos hecho bien y tenemos todo mapeado correctamente. Podríamos activarlo, pero no hará nada ya que está vacío.

Como en mi caso no he tenido ningún error con esto, pasaremos directamente al siguiente módulo.

Capítulo 2

Implantación del Primer módulo

2.1. Creación y funcionamiento

Para la creación de este módulo he utilizado **Odoo Scaffold**, usando los siguiente comandos:

```
docker-compose exec web /bin/bash
odoo scaffold lista_tareas /mnt/extra-addons/
chmod 777 -R /mnt/extra-addons/lista_tareas
```

Código 2.1: Creación de módulo con Odoo Scaffold

Con el primer comando accedemos a la consola del contenedor de Odoo y después con Odoo Scaffold creamos el módulo lista_tareas con todo lo necesario.

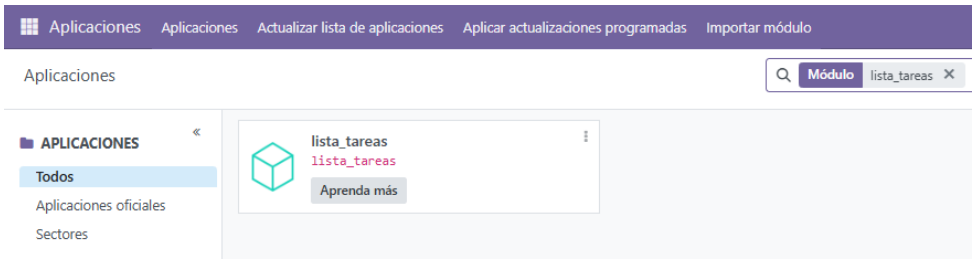
Tras haberlo creado utilizamos **chmod** para darle todos los permisos que necesite.

Ahora, utilizando el código que se nos aporta para crear una lista de tareas básica vamos a meterlo en nuestro archivos, especificamente en estos tres:

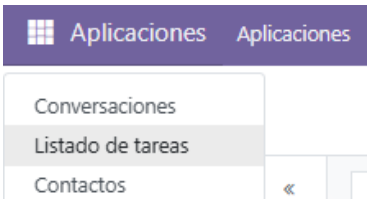
Tabla 2.1: Archivos cambiado para la lista de tareas

Archivo	**Propósito**
`__manifest__.py`	Contiene la información básica del módulo (nombre,descripción,etc)
`models.py`	Es un ejemplo o molde de los datos y que apartados tendrán
`views.xml`	Define como se verán las vistas de nuestro módulo

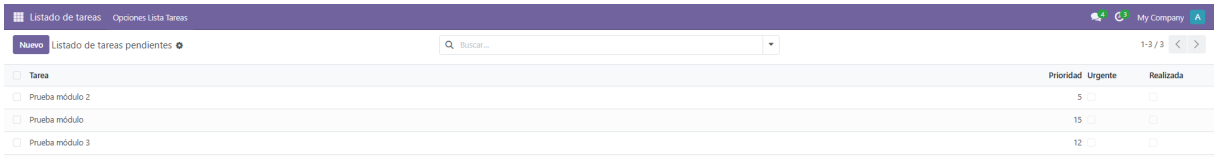
Una vez hayamos añadido/modificado el código necesario en los archivos mencionados previamente el módulo estará listo. Si lo buscamos debería aparecernos además de dejarnos activarlo y acceder a él.



Ahora que está activado accederemos desde la parte superior izquierda.



Y dentro, se debería mostrar lo esperado, una lista donde podremos añadir tareas y marcarlas como completadas o urgentes, además de ponerle una prioridad.



2.2. Problemas y Errores

A lo largo de la creación de este módulo si me surjieron un par de problemas:

2.2.1. Tabulaciones

Como no tengo demasiada experiencia con Python, olvidé algunas de las tabulaciones para el correcto funcionamiento del código y Odoo me estaba dando algunos errores al intentar activar el módulo. Simplemente repasé el código y fui tabulando todo lo necesario. De hecho, lo que más tiempo me llevó fue darme cuenta de que el 'api.depends' estaba mal tabulado y por eso el booleano de **Urgencia** no se activaba nunca.

2.2.2. Trees

Utilizando el código proporcionado había un problema, y es que este utilizaba **tree** para la lista de tareas, pero Odoo no lo aceptaba y no me dejaba acceder al módulo. La solución fue cambiarlo por un **list** que es muy similar y cumple la misma función. Con eso Odoo ya me permitía acceder.

Al final, terminé simplificandolo y me decanté por dejarlo como un simple booleano que se activa a mano, igual que el de 'Tarea realizada'.

Capítulo 3

Modificación del Primer módulo

3.1. Mejoras propuestas

El objetivo será añadir la fecha de creación y fecha de finalización a cada tarea. Esto es especialmente útil para tener un registro más avanzado y detallado de nuestras tareas.

3.2. Creación y funcionamiento

Empezaremos por la fecha de creación, para esto añadiremos esto en **'models.py'**:

```
fechacreacion = fields.Datetime(string='Fecha de creacion',default=
    fields.Datetime.now)
```

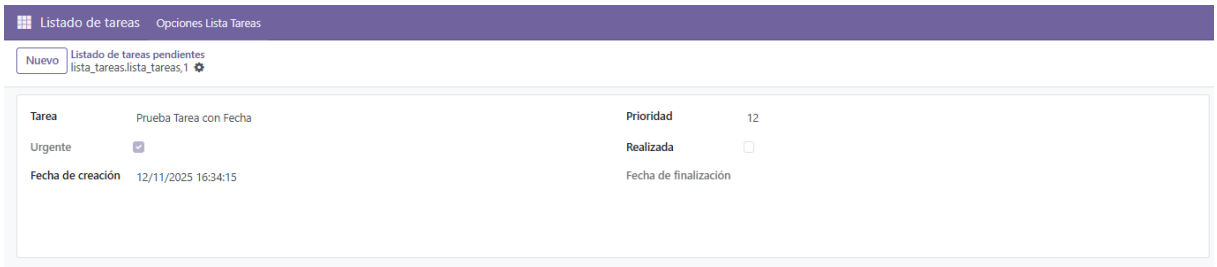
Código 3.1: Variable que obtiene la fecha actual

Esto se encargará de marcar la fehca y hora en el momento en el que se crea una nueva tarea. Después para que esto se muestre añadiremos un field en **´views.xml´**

```
<field name='fecha_creacion'></field>
```

Código 3.2: Campo que se muestra en odoo con la fecha de creación

Cuando creemos una tarea se tomará la fecha actual y se pondrá como fecha de creación como se muestra a continuación:



Ahora para la fecha de finalización haremos algo muy similar. Añadiremos en esto en **'models.py'**:

```
fecha_terminada = fields.Datetime(string='Fecha de finalizacion',compute
    ='_compute_fecha_terminada',store=True)
```

Código 3.3: Variable que obtiene la fecha actual y llama una funcion

Esta se encargará de llamar una función que tomará la fecha actual en el momento en el que se marque la tarea como realizada. La función es la siguiente:

```
    @api.depends('realizada')
    def _compute_fecha_terminada(self):
        for record in self:
            if record.realizada and not record.fecha_terminada:
                record.fecha_terminada = fields.Datetime.now()
            else:
                record.fecha_terminada = False
```

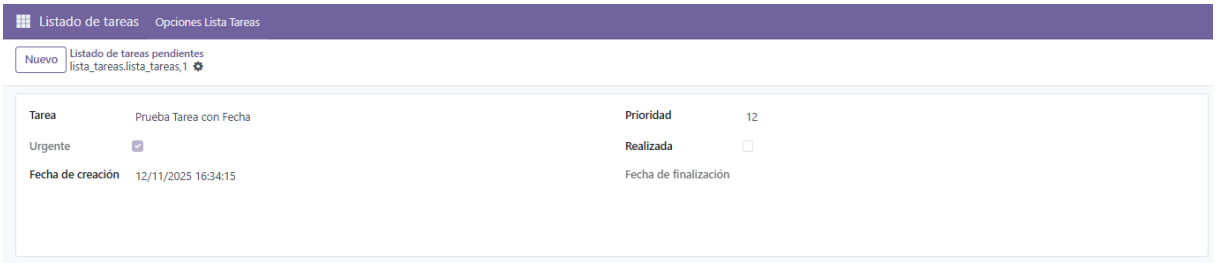
Código 3.4: Funcion para tomar la fecha actual cuando se marca como completada

Ahora solo queda mostrala en su respectivo campo, esto lo haremos añadiendo el field a **'views.xml'**:

```
<field name='fecha_terminada'></field>
```

Código 3.5: Campo que se muestra en odoo con la fecha de finalización

Entonces al marcar una tarea como terminada se tomará la fecha actual y se pondrá como fecha de finalización como se muestra a continuación:



A la hora de modificar este módulo no he tenido ningún problema en especial, así que con esto quedaría terminado.