

Práctica Primeros Módulos Odoo

Preparación

- Necesitamos un directorio donde guardar los addons (configurado en la opción `addons_path` del `odoo.conf`). Deberías de tenerlo de cuando se instalaron los módulos de OCA.

```
GNU nano 7.2
[options]
admin_passwd = passwd
db_host = psql
db_user = odoo17
db_password = odoo17
db_port = 5432
addons_path = /opt/odoo/addons
```

- Compartir el directorio por samba para poder acceder a él desde tu equipo.

```
sudo apt update sudo apt install samba
```

```
sudo nano /etc/samba/smb.conf
```

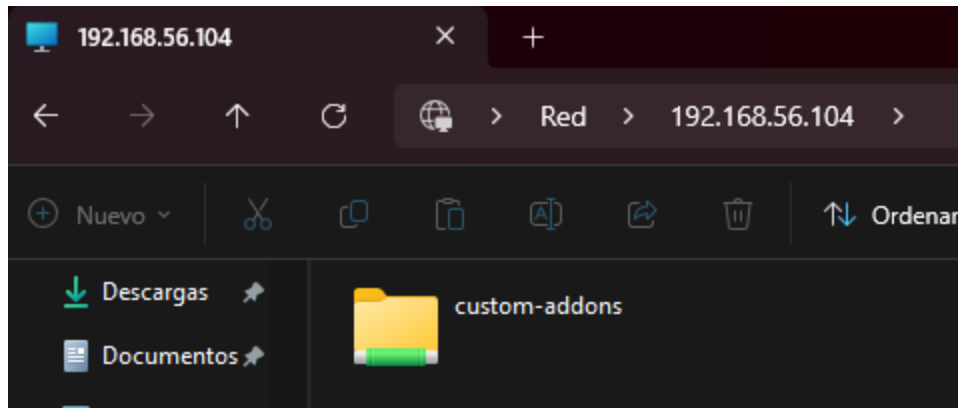
```
[custom-addons]
  path = /home/user/docker/odoo/custom-addons
  browseable = yes
  writable = yes
  guest ok = no
  valid users = user
```

```
sudo chown -R user:user /home/user/docker/odoo/custom-addons
```

```
sudo smbpasswd -a user
```

```
sudo systemctl restart smbd
```

Verificar el acceso al recurso compartido:



- En nuestro equipo tener Visual Studio, Atom o cualquier editor de código que nos guste. Comprobar que puedo acceder al directorio de la máquina virtual (compartido por samba) donde están los addons.

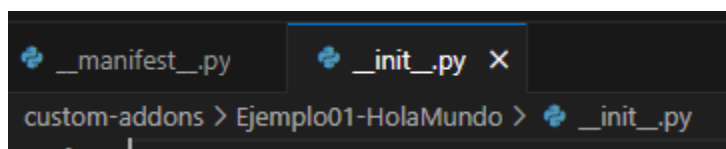
Nuestro primer modulo: “Hola mundo”

1. Añade el módulo MiMundo_tunombre (di te llamas Serafín, pues `mimundo_serafin`) siguiendo las siguientes instrucciones.

Para este primer “Hola mundo”, crearemos una carpeta “Ejemplo01-HolaMundo” dentro del directorio que hayamos configurado para colocar nuestros módulos.

Esta carpeta contendrá dos ficheros:

a) Fichero “`__init__.py`”:
Este fichero estará vacío.



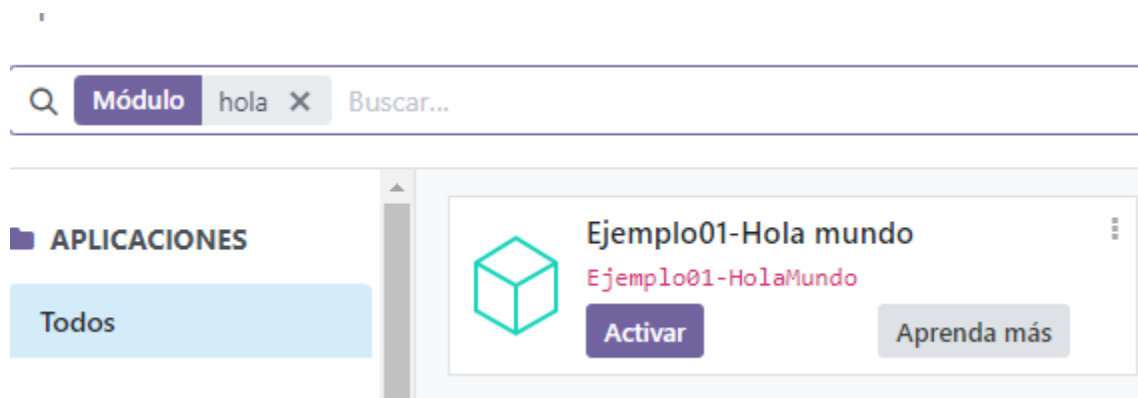
b) Fichero “`__manifest__.py`”:
Este fichero contendrá el siguiente código:

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

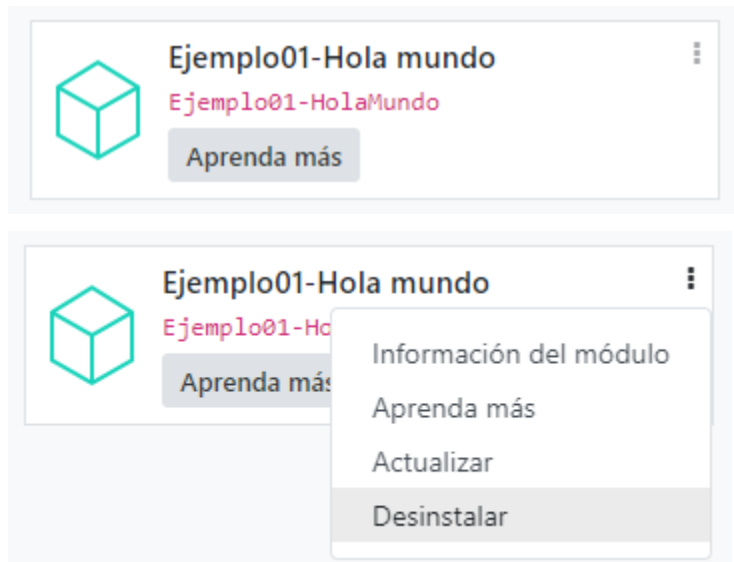
```
__manifest__.py X  __init__.py
custom-addons > Ejemplo01-HolaMundo > __manifest__.py
1  # -*- coding: utf-8 -*-
2  {
3      'name': 'Ejemplo01-Hola mundo',
4      'version': '1.0',
5      'summary': 'Primer módulo Hola Mundo',
6      'category': 'Tools',
7      'author': 'kevin',
8      'website': 'http://www.ejemplo.com',
9      'description': """
10     Este es nuestro primer módulo en Odoo.
11     """,
12      'depends': [],
13      'data': [],
14      'installable': True,
15      'application': True,
16  }
```

Una vez creada la estructura, con el “Modo Desarrollador” activado, podréis ir al listado de módulos y “Actualizar la lista de aplicaciones” tal como se observa en la imagen

Tras ello, eliminando los filtros de búsqueda por defecto y buscando “hola mundo”, podremos encontrar nuestro módulo. Si todo ha funcionado correctamente, veremos algo similar a:



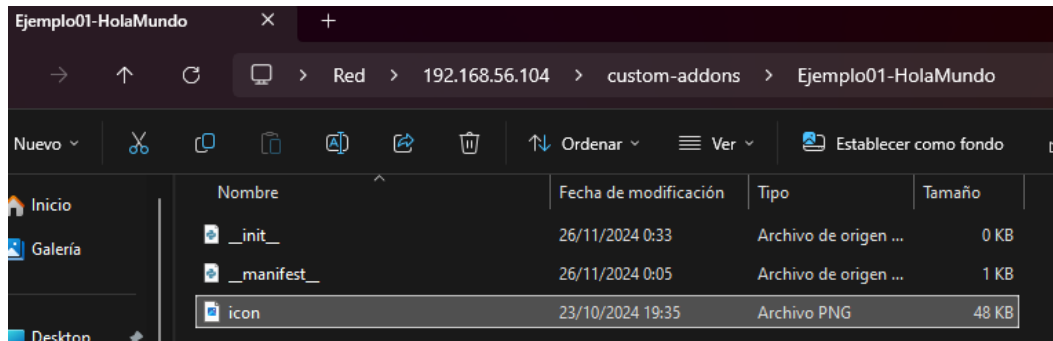
2. Ahora podremos instalar nuestro módulo para probarlo (aunque este ejemplo no hace nada). Desinstálalo.



3. En la página 74 del libro *Odoo Essentials* tienes información de algunos de los campos que puedes rellenar en el fichero `__manifest__.py`. Añade la siguiente información:

- a) Nombre del módulo
- b) Resumen
- c) Autor
- d) Licencia (elige una con sentido)
- e) Página web
- f) Configura el flag aplicación para que salga el módulo cuando tienes el filtro Apps
- g) Añade el módulo a la categoría Pruebas/Practica
- h) Añade un icono

```
# -*- coding: utf-8 -*-
{
    'name': 'Ejemplo01-Hola Mundo',
    'summary': 'Primer módulo Hola Mundo para aprender Odoo.',
    'version': '1.0',
    'author': 'kevin',
    'license': 'LGPL-3',
    'website': 'http://www.ejemplo.com',
    'category': 'Pruebas/Practica',
    'application': True,
    'icon': '/Ejemplo01-HolaMundo/icon.png',
    'depends': [],
    'data': [],
    'installable': True,
}
```



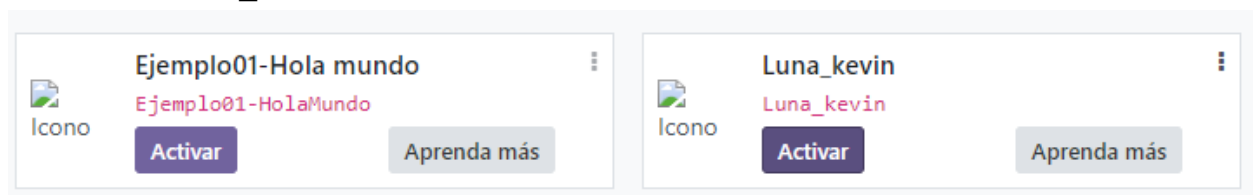
4. Crea un nuevo módulo que se llame luna_tunombre,

a) Configúralo de manera similar al módulo Mundo.

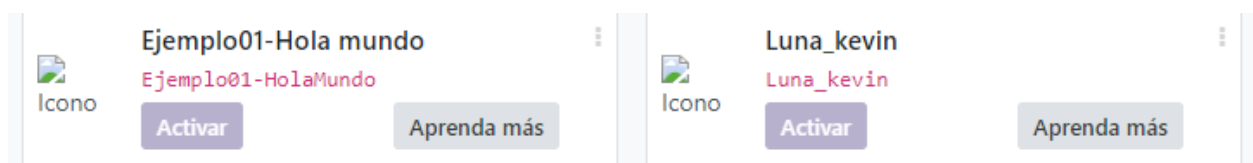
b) En las dependencias configura para que dependa del módulo Mundo. Con los dos módulos desinstalados instala el módulo Luna. Al depender de Mundo se debe de instalar Mundo automáticamente.

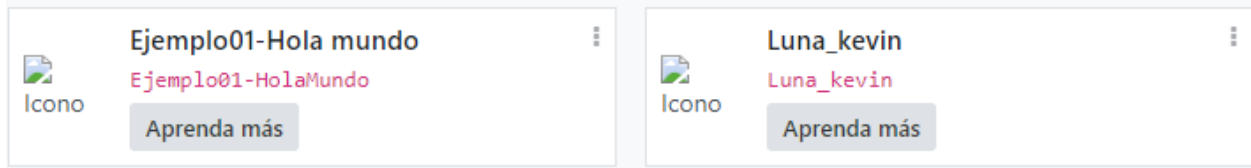
```
# -*- coding: utf-8 -*-
{
    'name': 'Luna_kevin',
    'summary': 'Módulo Luna que depende de Mundo.',
    'version': '1.0',
    'author': 'kevin',
    'license': 'LGPL-3',
    'website': 'http://www.ejemplo.com',
    'category': 'Pruebas/Practica',
    'application': True,
    'icon': '/Luna_kevin/icon.png',
    'depends': ['Ejemplo01-HolaMundo'],
    'data': [],
    'installable': True,
}
```

Activamos Luna_kevin



Y se activan los dos módulos





Comando scaffold

Los ficheros necesarios que aparecen en un módulo de Odoo siguen siempre la misma estructura

Vamos a realizar una breve explicación de cada uno los ficheros generados:

- “__manifest__.py”: es el manifiesto del módulo. Incluye información como el título, descripción, así como ficheros a cargar.
- “views”: describe las vistas de nuestro módulo (formulario, árbol ,menús, etc.).
- “demo”: incluye datos “demo” para el ejemplo propuesto de modelo.
- “security”: Aquí se guardarán los ficheros de seguridad.
- “controllers”: aquí tendremos “métodos” para añadir control sobre los datos más allá de la declaración del modelo.

Con el comando scaffold creamos la estructura básica de ficheros para la implementación de un módulo. Por ejemplo, si quiero crear el módulo lista_tareas y el directorio donde tengo los addons es /mnt/extra-addons el comando sería:

```
odoo scaffold lista_tareas /mnt/extra-addons/
```

Si estás utilizando docker composer deberás ejecutar el comando utilizando docker exec

```
docker exec contenedor comando
```

1. Crea un módulo llamado hola_sol utilizando el comando scaffold. Modifica algunos parámetros del fichero manifest e instálalo.

En mi caso odoo está instalado en docker, ejecutamos bash y creamos la estructura básica, en mi caso en /opt/odoo/addons:

```
odoo scaffold hola_sol /opt/odoo/addons
```

```
user@kevinserver:~/docker/odoo$ docker exec -it --user root odoo_odoo_1 bash
root@f875101482a3:/# odoo scaffold hola_sol /opt/odoo/addons
root@f875101482a3:/#
```

```
user@kevinserver:~/docker/odoo/custom-addons$ tree hola_sol
hola_sol
├── controllers
│   ├── controllers.py
│   └── __init__.py
├── demo
│   └── demo.xml
├── __init__.py
├── manifest__.py
├── models
│   ├── __init__.py
│   └── models.py
├── security
│   └── ir.model.access.csv
└── views
    ├── templates.xml
    └── views.xml

6 directories, 10 files
user@kevinserver:~/docker/odoo/custom-addons$ _
```

Manifest por defecto:

```
# -*- coding: utf-8 -*-
{
    'name': "hola_sol",

    'summary': "Short (1 phrase/line) summary of the module's purpose",

    '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/15.0/odoo/addons/base/data/ir\_module\_category\_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/views.xml',
    'views/templates.xml',
],
# only loaded in demonstration mode
'demo': [
    'demo/demo.xml',
],
}

```

Manifest modificado:

```

# -*- coding: utf-8 -*-
{
    'name': "Hola Sol",
    'summary': "A practice module to demonstrate Odoo functionality.",
    'description': """
This module serves as an example for learning how to create and manage Odoo modules.
""",
    'author': "Tu Nombre",
    'website': "https://www.example.com",
    'category': 'Practice/Examples',
    'version': '1.0',
    'license': 'LGPL-3',

    # Dependencias necesarias
    'depends': ['base'],

    # Archivos cargados al instalar el módulo
    'data': [
        # Añadir aquí archivos XML si fuera necesario
        'views/views.xml',
        'views/templates.xml',
    ],

    # Archivos de datos de demostración (opcional)
    'demo': [
        'demo/demo.xml',
    ],

    # Configuración de la aplicación
    'installable': True,
    'application': True,
    'auto_install': False,
}

```




Hola Sol



hola_sol

Activar

Aprenda más



Hola Sol



hola_sol

Aprenda más