

Asignatura

Sistema de Gestión Empresarial

UNIDAD 7

Desarrollo de componentes



UNIVERSAE
Instituto Superior de FP

Estructura básica de un módulo de Odoo

Descripción

Un módulo de Odoo consta mínimo de 2 ficheros, que son los principales:

- `__init__.py`
- `__manifest__.py`
- `models/`
- `views/`

```
"C:\Program Files\Odoo 16.0.20221114\python\python"  
"C:\Program Files\Odoo 16.0.20221114\server\odoo-bin"  
scaffold mimodulo addons/
```



```
bitnami@debian: ~/htdocs$
```

7.2. Generación de un módulo

Mediante la creación manual de archivos y carpetas se puede crear un nuevo módulo completamente funcional. Sin embargo, Odoo proporciona un mecanismo para generar automáticamente esta estructura básica, incluyendo en algunos de los archivos líneas ejemplo del contenido que debe tener, lo cual es de gran ayuda para el programador.

7.2.1. Creación de un módulo en el comando scaffold

La creación de la estructura de un módulo base se puede hacer automáticamente usando el comando scaffold que, como apoyo, genera la cantidad mínima de archivos y carpetas necesarias para que el módulo sea reconocido por Odoo.

```
$ sudo odoo scaffold "nombre del módulo" "carpeta_para_addons"/
```

Así, por ejemplo, para crear el módulo llamado `jig-modulo` en la misma carpeta donde están el resto de los módulos, con odoo se hace de la siguiente manera:

```
$ sudo odoo scaffold mi_modulo addons/
```

que genera el directorio `mi_modulo` con el contenido básico específico.

```
profesor@profesor-VirtualBox:~/odoo$ ./odoo-bin scaffold mi_modulo1 addons
profesor@profesor-VirtualBox:~/odoo$ ls
addons      debian      MANIFEST.in  README.md    setup
CONTRIBUTING.md  doc        odoo         requirements.txt  setup.cfg
COPYRIGHT      LICENSE    odoo-bin     SECURITY.md    setup.py
```

Imagen 1. Resultado del comando Scaffold

Como hemos estudiado, cada uno de los ficheros generados incluye un ejemplo del contenido que debe tener el fichero.

En `_init_.py` se incluyen las carpetas en las que debe buscar Python los archivos `.py` que se van a ejecutar.



```
bitnami@debian:~/htdocs/directoriodulos/modulocrm$ ls
controllers demo __init__.py __manifest__.py models security views
bitnami@debian:~/htdocs/directoriodulos/modulocrm$ cat __init__.py
# -*- coding: utf-8 -*-

from . import controllers
from . import modelsbitnami@debian:~/htdocs/directoriodulos/modulocrm$ _
```

GNU nano 5.4

models.py

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

# from odoo import models, fields, api

# class modulocrm(models.Model):
#     _name = 'modulocrm.modulocrm'
#     _description = 'modulocrm.modulocrm'

#     name = fields.Char()
#     value = fields.Integer()
#     value2 = fields.Float(compute="_value_pc", store=True)
#     description = fields.Text()
#
#     @api.depends('value')
#     def _value_pc(self):
#         for record in self:
#             record.value2 = float(record.value) / 100
```

Creación paso a paso de un módulo

Descripción

La creación de un módulo se debe tener en cuenta que los modelos son la base del módulo y se asocian con tables de la bbdd.

Por otro lado, tenemos las vistas asociadas a los modelos, en `views/views.xml`






Aplicaciones

Buscar...

▼ Filtros Agrupar por ★ Favoritos

CATEGORÍAS


Todos	
Ventas	48
Servicios	10
Contabilidad	143
Inventario	17
Fabricación	12



Proyecto
project

Instalado


APRENDER MÁS



Partes de horas
timesheet_grid

APRENDER MÁS


Actualizar



Servicio externo
industry_fsm

APRENDER MÁS


Actualizar



Registros de tareas
hr_timesheet

ACTIVAR


APRENDER MÁS



Employee Hourly Wage
hr_hourly_cost

ACTIVAR

MÁS INFORMACIÓN



Gastos del proyecto
project_hr_expense

ACTIVAR

MÁS INFORMACIÓN

The background is a solid blue color with a subtle, larger-scale grid pattern. Overlaid on this are numerous small, light blue arrows of various sizes and orientations, some pointing towards the center and others away from it, creating a sense of movement and direction.

UNIVERSAE

— CHANGE YOUR WAY —