

## Síntesis conceptual

<b>Grado:</b> Desarrollo de aplicaciones multiplataforma
<b>Asignatura:</b> Sistemas de gestión empresarial
<b>Unidad:</b> 7. Desarrollo de componentes

## Resumen

No solo se pueden modificar los objetos de *Odoo* mediante interfaz gráfica, sino que con código *Python* y XML principalmente se puede crear la estructura de cualquier módulo de *Odoo*.

Existen IDE o procesadores especializados que nos permiten trabajar directamente sobre el servidor de *Odoo* sin necesidad de estar descargando y subiendo los ficheros continuamente.

Para crear una estructura básica de un módulo de *Odoo* usamos el comando **scaffold**. Las carpetas que se crean cuando se usa este comando incluyen algunas líneas que se encuentran comentadas y sirven para ayudar al programador a poder escribir su propio código.

Existen dos ficheros fundamentales para un módulo que se alojan en el directorio raíz, estos son *\_init\_.py* y *\_manifest\_.py*. El segundo es donde se almacena toda la información de descripción del módulo además de sus dependencias, y la declaración de los ficheros *.xml* que lo integran. El fichero *\_init\_.py* por otro lado es el primer fichero en ser ejecutado y el que contiene información y la localización de los otros ficheros *.py* que se van a ejecutar,

Para escribir las vistas usamos lenguaje *.xml* y *Qweb*. Estas vistas siempre se van a referir a un módulo en concreto. Las vistas se definen en el fichero o los ficheros *views.xml*, almacenado en el directorio *views*, también se definen aquí los elementos de menú y las acciones de ventana asociadas a estas vistas.

Para desarrollar los informes, de nuevo se usa el lenguaje XML y *Qweb*. Estos ficheros se suelen alojar en el directorio *reports*, Los informes deben de estar declarados en el fichero *\_init\_.py*.

Si queremos que a las vistas únicamente puedan acceder los grupos que nos interesan, usaremos, dentro del directorio *security* el fichero *security.xml*. En dicho fichero se debe de añadir una línea por cada uno de los grupos. Los permisos de cada uno de los grupos se definen en un fichero csv. Por último, hay que indicar que el fichero *security.csv* tiene la siguiente cabecera:

- *id*
- *name*
- *mode\_id:id*
- *group\_id:id*
- *perm\_read*
- *perm\_write*
- *perm\_create*
- *perm\_unlink*

También los siguientes campos se ponen en 0 o 1 según se tenga o no el permiso:

- *perm\_read*
- *perm\_write*
- *perm\_create*
- *perm\_unlink*

## Conceptos fundamentales

- **Atom:** es una variante de *Chromium* que se diseña con la intención de ser un editor de texto de código abierto especialmente para codificación.
- **PyCharm:** Es un IDE especializado en *Python* que tiene una versión *Community*. Es de código abierto y además permite también trabajar con distintos lenguajes al mismo tiempo que depurar aplicaciones en *hosts* remotos.
- **Visual Studio Code:** editor de código que está optimizado para poder crear y depurar algunas aplicaciones. Fue creado por Microsoft.
- **SublimeText:** editor de texto para creación de código característico de los sistemas UNIX.
- **scaffold:** comando para creación de módulos en *Odoo*.