**Desarrollo de un módulo - Configuración inicial**

**¿Qué es un módulo?**

Cuando instalamos un sistema ERP, encontramos situaciones en las que no es posible cubrir todas las necesidades del cliente con los componentes existentes (<https://www.odoo.com/apps>)  
Odoo es un sistema flexible y nos ofrece mecanismos para ampliar su funcionalidad mediante desarrollos propios que se integren con el resto de componentes.  
En el apartado siguiente veremos la estructura que deben tener esos módulos para que sean reconocidos e integrados por Odoo.

**Nuestro primer módulo**

**Elementos principales de un módulo**

Un módulo es un directorio situado dentro de un directorio de addons de Odoo. Como ya sabemos, podemos especificar el directorio que deseemos en el fichero de configuración del servidor (en nuestro caso, odoo.conf).

Este fichero se encuentra en /etc/odoo.conf

Deberemos añadir/ modificar el directorio en el que queramos trabajar con los nuevos módulos que desarrollemos, por lo que deberemos añadir las ubicaciones separadas por una , en la línea “addons\_path:

Texto

Descripción generada automáticamente

En nuestro caso trabajaremos en esa ruta /opt/odoo/custom-addons (este directorio custom\_addons, lo deberemos crear nosotros, además deberá ser propiedad del usuario odoo, creado durante la instalación del servicio)

Para nuestro primer módulo de ejemplo crearemos la carpeta “ejemplo\_comunicacion” dentro del directorio custom\_addons. Esto lo podremos hace de múltiples formas, o bien usando la extensión de Visual Studio Code SFTP, con la aplicación WinSCP (como vimos en el anterior documento), o creándola directamente desde nuestro servidor.

El directorio de un módulo contiene al menos los siguientes ficheros.

\_\_init\_\_.py  
\_\_manifest\_\_.py

Adicionalmente:

* Ficheros definición de modelo/objeto.py (Business objects)
* Fichero definición de vistas, acciones, workflows en xml (si fuese necesario)
* Ficheros de datos ejemplo/iniciales en xml/csv(Data files)
* Ficheros definición de controlador web (Web Controllers)
* Contenido estático a usar: imágenes, css, javascript (Static web objects)

El fichero \_\_init\_\_.py de momento lo podemos dejar vacio, pero en el fichero \_\_manifest\_\_.py deberemos agregar la descripción básica del modulo, como es su nombre, esto lo haremos en formato diccionario.

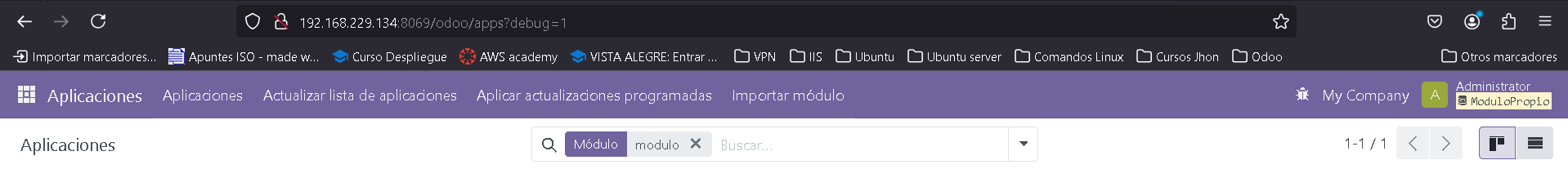
# \_\*\_ coding: utf-8 \_\*\_

{

    'name':'Primer modulo para SGE'

}

Una vez configurado nuestro módulo, debemos ir a Odoo / Configuración / Actualizar lista de módulos (recordad que esta opción solo aparece si tenemos el modo desarrollador habilitado para el usuario).



Una vez actualizada la lista, ya lo tendremos disponible para instalar/activar

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Construcción automática de un modulo

Scaffold

**Elaborando nuestro módulo**

[Ocultar](https://www.iax.es/rea/informatica01/elaborando_nuestro_mdulo.html)

Desde la versión 8, Odoo incorpora un script para generar la estructura y ficheros necesarios para crear un módulo. Estas técnicas, introducidas originalmente en lenguajes tipo Ruby, se conocen como scaffolding

$ ./odoo-bin scaffold <module name> <where to put it>

Vamos a generar un módulo soporte que nos permita introducir incidencias en el sistema.

$ ./odoo-bin scaffold modulo\_personal custom\_addons/

En esta estructura, los ficheros \_\_init\_\_.py corresponden a la forma que tiene Python de especificar cualquier paquete. Se ejecutan al cargar la aplicación. Se encargan de cargar los componentes necesarios para el módulo, la lógica del módulo.

Imagen que contiene Texto

Descripción generada automáticamente

El resto de los ficheros se utilizan para:

* controllers/controllers.py: contiene un ejemplo de un controlador de rutas, implementa algunas.
* demo/demo.xml: datos de ejemplo para el módulo/modelo.
* models/model.py: contiene un ejemplo del modelo de datos y sus campos.
* security/ir.model.access.csv: lista de control de acceso
* views/views.xml: describe las vistas del modelo (tree, form, qweb, menú, acciones…)

En el fichero \_\_manifest\_\_.py deberemos configurar algunas cosas :

Los parámetros:

* sumarry
* description
* author
* website
* category

Texto

Descripción generada automáticamente

La clave ‘depends’ indicara los módulos del que depende el modulo actual, por defecto todos los módulos que creamos dependen de los modelos base, pero esto podría cambiar en caso de que creemos un modulo que amplie las funcionalidades de otro modulo en cuyo caso solo dependería de ese modulo principal.

Lo siguiente que debemos hacer es descomentar la línea de security

Texto

Descripción generada automáticamente

A continuación, os dejo con el video correspondiente a toda esta configuración🡪

[*Nuestro primer módulo*](https://www.youtube.com/watch?v=TEr7tpka0gk&list=PLOgymZ7xUgbb4iTyZxCjAPO_lExJCpSXr&index=2) ([CC BY-NC-SA](http://creativecommons.org/licenses/?lang=es))