Hoy aprenderemos a hacer esto en Python:

* Crear un modelo en la BB.DD
* Crear vistas para los datos que almacenará el modelo
* Crear una opción de menú para acceder a esas vistas.
* Ver como modificar un modelo ya existente.

Captura de pantalla de computadora

Descripción generada automáticamente

**Los archivos \_\_init\_\_.py y \_\_manifest\_\_.py:**

# Manifest:

Archivo de configuración basado en un diccionario clave-valor de Python: para describir la configuración de la aplicación.

Texto

Descripción generada automáticamente

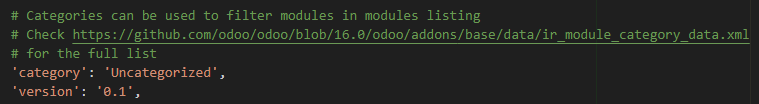
Texto

Descripción generada automáticamente

* Name: nombre del módulo.
* Summary: descripción corta.
* Description: descripción larga.

Texto

Descripción generada automáticamente



Podemos incluir nuestro módulo en una categoría.

**Lo interesante**:

Dependencias:



Lo normal es que incluyamos el módulo base de Odoo (como mínimo., que tiene todo lo relacionado con las configuraciones básicas, usuarios, etc. Y que nos sirve para montar cualquier aplicación. Hay otros módulos, como por ejemplo para el email (“mail”) .

En el apartado data, especificamos los archivos que usara nuestra aplicación:

Texto

Descripción generada automáticamente

Views es fundamental para nuestras vistas.

Por defecto, también tenemos datos de demo, que podemos incluir, para que si marcamos la casilla de activar datos de demostración.

Texto

Descripción generada automáticamente

# Init

Texto

Descripción generada automáticamente

Sirve para importar los controladores y los modelos. Sirve para importar las librerías que necesitemos.

# Instalar nuestra propia aplicación/modulo.

Necesitaremos activar un nuevo menú en odoo para “Actualizar lista de aplicaciones” para que nos aparezca nuestro módulo para instalarlo en el menú de Aplicaciones.

1. Activar el modo desarrollador.
2. Actualizar lista de aplicaciones:



1. Activarla:

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

En más información, podremos ver todos los datos que pusimos en el fichero manifest:

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Si modificamos los datos en el manifest, tendremos que **reiniciar el servicio** de Odoo para que esos cambios se vean reflejados en Odoo:

Texto

Descripción generada automáticamente

# Crear los modelos donde guardaremos los datos

Models.py

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Esto nos permite definir nuestras clases de Python:

Texto

Descripción generada automáticamente

Aquí por ejemplo he descomentado algunas de las líneas que vienen por defecto en el fichero models.py.

* Class invernadero(models.Model): lo que está entre paréntesis es la clase de la que hereda.
* Los atributos name y description siempre los pondremos.

Cada clase que escribamos en models.py se traducirá en una tabla de Postgres.

Es decir, cada clase se mapeará en una tabla de la BB.DD.

Nomenclatura recomendada:

* \_name: nombreMódulo.nombreModelo
* \_description: nombreModulo.descripcionModelo

Después de esos campos, vienen **nuestros** **propios campos**.

**Tipos de datos**:

Text: campo de caracteres más extenso que el campo tipo Char

Texto

Descripción generada automáticamente

Ahora tendríamos que reiniciar el servidor para que surtan efecto.

sudo service odoo restart.

# Crear vistas para ver el contenido del modelo o introducir datos

Se hace en views.xml

Descomentar estas líneas:

Pantalla de computadora

Descripción generada automáticamente

Habría que cambiar el nombre del modelo por el de “invernadero.planta”, ya que lo hemos cambiado anteriormente. Y también cambiar el nombre de los fields, tal y como los pusimos en el modelo.

También habría que definir una acción para abrir una ventana con esa vista (descomentando estas líneas):

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Y descomentar estas líneas para que aparezca en el menú principal de Odoo:

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente