



SIST. GESTIÓN EMPRESARIAL

PRÁCTICA 05

Shaila Pérez Fernández

Organización, consulta y tratamiento de la información

Desarrollo de Aplicaciones Multiplataforma (DAM)

IES Juan José Calvo Miguel

PRÁCTICA 5. CONTENIDO.

ACTIVIDAD 1. MANEJO DE USUARIOS EN ODOO 10.

NOMBRE PARA EL MÓDULO → practicasfct

Desarrollo de componentes en Odoo 10.

Vamos a cd addons

Para crear un módulo vamos a tener que configurar una serie de cosas →
Nuestro módulo va a ser un directorio extra en nuestro directorio addons

1º PASO: reiniciar el servidor de Odoo → `systemctl restart odoo10.service`

(Siempre hay que reiniciar el servidor cuando hagamos modificaciones en los archivos de Python, para que los recargue).

2º PASO: Configurar errores Odoo → archivo de log donde Odoo guarda los errores. Vamos a →
`cd /etc/systemd/system/`

`ls`

Podremos ver

Nano `odoo10.service` (Esto lo hace Pablo porque ya lo tiene configurado). → Aquí tenemos un logfile que se puede direccionar a otra ruta → aquí podemos ver porque rompe y los problemas.



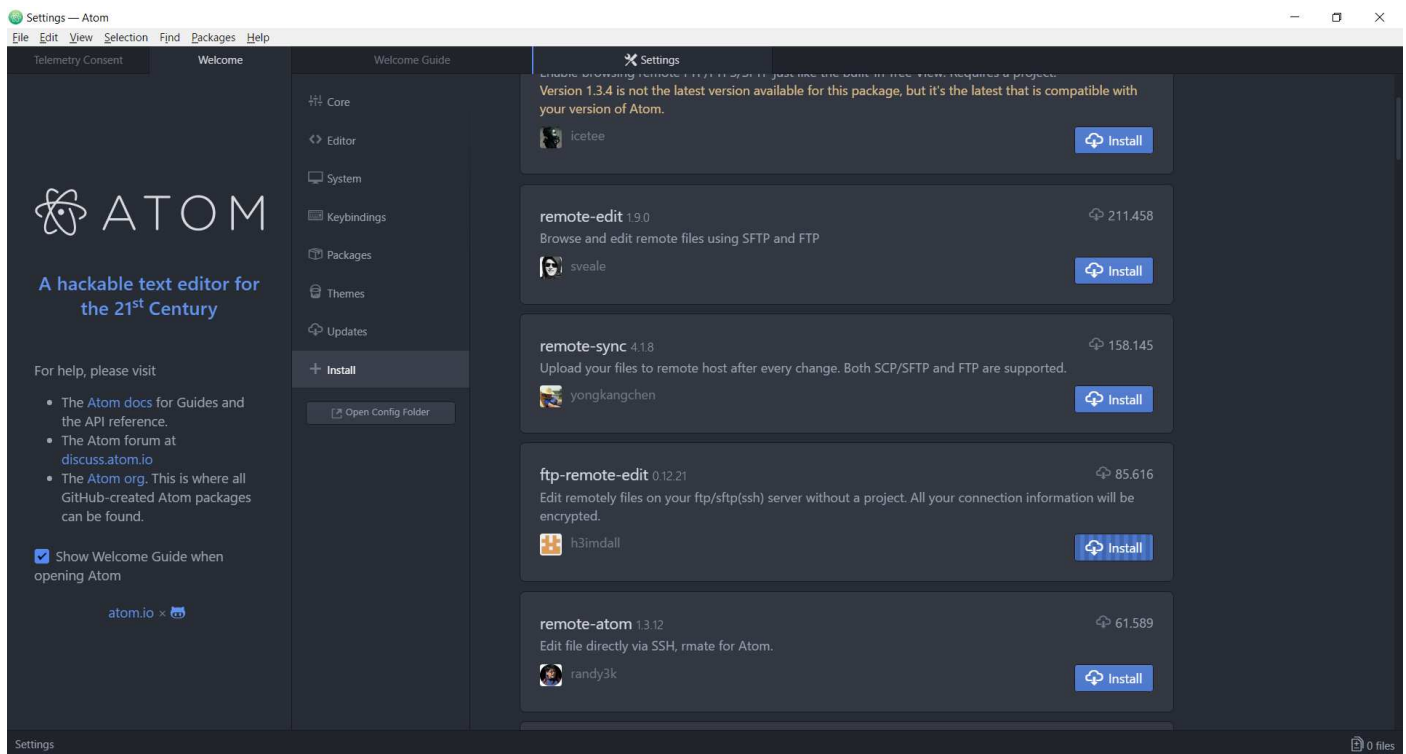
```
GNU nano 2.5.3 File: odoo-server.conf Modified ^
[options]
admin_passwd = admin

xmlrpc = True
xmlrpc_port = 8069

db_host = 192.168.9.11
db_port = 5432
db_user = odoo
db_password = odoo

addons_path = /home/odoo/odoo/addons
log_level=info
logfile=/home/odoo/odoo.log
```

Descargar atom → instalar plugin → Vamos a Packages → Settings View → Open → Install →
Buscar → remote → ftp-remote-edit



3º PASO: Editar servidores → Vamos a Packages → Ftp-remote → Edit servers →

Ftp-Remote-Edit Server Settings

You can edit each connection at the time. All changes will only be saved by pushing the save button.

▼

The name of the server.

The hostname or IP address of the server. Port

Username for authentication.

Password/Passphrase for authentication.

☐ Use sftp (ssh) connection.

Initial Directory.

Ftp-Remote-Edit Server Settings

You can edit each connection at the time. All changes will only be saved by pushing the save button.

odooshaila



New

Delete

Duplicate

The name of the server.

odooshaila

The hostname or IP address of the server.

Port

192.168.9.11

21

Username for authentication.

odoo

Password/Passphrase for authentication.

☒ Use sftp (ssh) connection.

Path to private keyfile.

path to private keyfile (optional)

Initial Directory.

/home/odoo/odoo/addons/

Save

Cancel

```
login as: root
root@192.168.9.11's password:
Access denied
root@192.168.9.11's password:
Access denied
root@192.168.9.11's password:
Welcome to Ubuntu 16.04 LTS (GNU/Linux 4.15.17-3-pve x86_64)

 * Documentation:  https://help.ubuntu.com/
New release '18.04.1 LTS' available.
Run 'do-release-upgrade' to upgrade to it.

You have mail.
Last login: Thu Feb  7 08:54:42 2019 from 192.168.8.176
root@shailapf40SGE1819:~# service odoo restart
Failed to restart odoo.service: Unit odoo.service not found.
root@shailapf40SGE1819:~# systemctl restart odoo10.service
root@shailapf40SGE1819:~# su odoo
odoo@shailapf40SGE1819:/root$ cd ~
odoo@shailapf40SGE1819:~$ ls
ll_15_2018copia.sql          odoo                  server-tools            ???02_01_2019???copia.sql
account-financial-reporting odoo-server.conf      webservice              ???02_08_2019???copia.sql
copia.sh                   odoo.log              wkhtmltox-0.12.1_linux-trusty-amd64.deb ???11_16_2018???copia.sql
copia.sql                  odoo.server.conf      ???01_11_2019???copia.sql ???11_30_2018???copia.sql
copia_de_seguridad         partner-contact       ???01_18_2019???copia.sql ???12_07_2018???copia.sql
ll0n-spain                 reporting-engine      ???01_25_2019???copia.sql
odoo@shailapf40SGE1819:~$ exit
exit
root@shailapf40SGE1819:~# su odoo
odoo@shailapf40SGE1819:/root$ cd /home/odoo/addons
bash: cd: /home/odoo/addons: No such file or directory
odoo@shailapf40SGE1819:/root$ cd /home/odoo/odoo/addons/
```

Crear directorio con el módulo que queremos crear → Librería.

Hay un comando que te permite crear la estructura del módulo. → comando scaffold → hay que pasarle el nombre del módulo y el directorio donde lo vamos a alojar.
Ir al directorio donde esté Odoo-bin. Hay que entrar como Odoo porque si no, no tenemos los permisos suficientes.

```
odoo@shailapf40SGE1819:~$ ls
ll_15_2018copia.sql          odoo                  server-tools          ???02_01_2019???copia.sql
account-financial-reporting  odoo-server.conf     webservice            ???02_08_2019???copia.sql
copia.sh                    odoo.log              wkhtmltox-0.12.1_linux-trusty-amd64.deb ???11_16_2018???copia.sql
copia.sql                   odoo.server.conf      ???01_11_2019???copia.sql ???11_30_2018???copia.sql
copia_de_seguridad          partner-contact       ???01_18_2019???copia.sql ???12_07_2018???copia.sql
ll0n-spain                  reporting-engine      ???01_25_2019???copia.sql

odoo@shailapf40SGE1819:~$ cd home
-bash: cd: home: No such file or directory
odoo@shailapf40SGE1819:~$ cd /home
odoo@shailapf40SGE1819:/home$ ls
examenleval  odoo
odoo@shailapf40SGE1819:/home$ cd /odoo
-bash: cd: /odoo: No such file or directory
odoo@shailapf40SGE1819:/home$ cd odoo
odoo@shailapf40SGE1819:~$ ls
ll_15_2018copia.sql          odoo                  server-tools          ???02_01_2019???copia.sql
account-financial-reporting  odoo-server.conf     webservice            ???02_08_2019???copia.sql
copia.sh                    odoo.log              wkhtmltox-0.12.1_linux-trusty-amd64.deb ???11_16_2018???copia.sql
copia.sql                   odoo.server.conf      ???01_11_2019???copia.sql ???11_30_2018???copia.sql
copia_de_seguridad          partner-contact       ???01_18_2019???copia.sql ???12_07_2018???copia.sql
ll0n-spain                  reporting-engine      ???01_25_2019???copia.sql

odoo@shailapf40SGE1819:~$ cd /home/odoo/odoo
odoo@shailapf40SGE1819:~/odoo$ ls
CONTRIBUTING.md  LICENSE      Makefile      addons      doc      odoo-bin      setup      setup.py
COPYRIGHT        MANIFEST.in  README.md    debian      odoo      requirements.txt  setup.cfg
```

./odoo-bin scaffold librería addons

```
odoo@shailapf40SGE1819:~$ ls
ll_15_2018copia.sql          odoo                  server-tools          ???02_01_2019???copia.sql
account-financial-reporting  odoo-server.conf     webservice            ???02_08_2019???copia.sql
copia.sh                    odoo.log              wkhtmltox-0.12.1_linux-trusty-amd64.deb ???11_16_2018???copia.sql
copia.sql                   odoo.server.conf      ???01_11_2019???copia.sql ???11_30_2018???copia.sql
copia_de_seguridad          partner-contact       ???01_18_2019???copia.sql ???12_07_2018???copia.sql
ll0n-spain                  reporting-engine      ???01_25_2019???copia.sql

odoo@shailapf40SGE1819:~$ cd home
-bash: cd: home: No such file or directory
odoo@shailapf40SGE1819:~$ cd /home
odoo@shailapf40SGE1819:/home$ ls
examenleval  odoo
odoo@shailapf40SGE1819:/home$ cd /odoo
-bash: cd: /odoo: No such file or directory
odoo@shailapf40SGE1819:/home$ cd odoo
odoo@shailapf40SGE1819:~$ ls
ll_15_2018copia.sql          odoo                  server-tools          ???02_01_2019???copia.sql
account-financial-reporting  odoo-server.conf     webservice            ???02_08_2019???copia.sql
copia.sh                    odoo.log              wkhtmltox-0.12.1_linux-trusty-amd64.deb ???11_16_2018???copia.sql
copia.sql                   odoo.server.conf      ???01_11_2019???copia.sql ???11_30_2018???copia.sql
copia_de_seguridad          partner-contact       ???01_18_2019???copia.sql ???12_07_2018???copia.sql
ll0n-spain                  reporting-engine      ???01_25_2019???copia.sql

odoo@shailapf40SGE1819:~$ cd /home/odoo/odoo
odoo@shailapf40SGE1819:~/odoo$ ls
CONTRIBUTING.md  LICENSE      Makefile      addons      doc      odoo-bin      setup      setup.py
COPYRIGHT        MANIFEST.in  README.md    debian      odoo      requirements.txt  setup.cfg
odoo@shailapf40SGE1819:~/odoo$ ./odoo-bin scaffold librer addons
odoo@shailapf40SGE1819:~/odoo$ ./odoo-bin scaffold librer addons librer^C
odoo@shailapf40SGE1819:~/odoo$ ./odoo-bin scaffold practicasfct addons
```

Init.py → Python lee este archivo y este archivo le indica que otros archivos hay que leer.

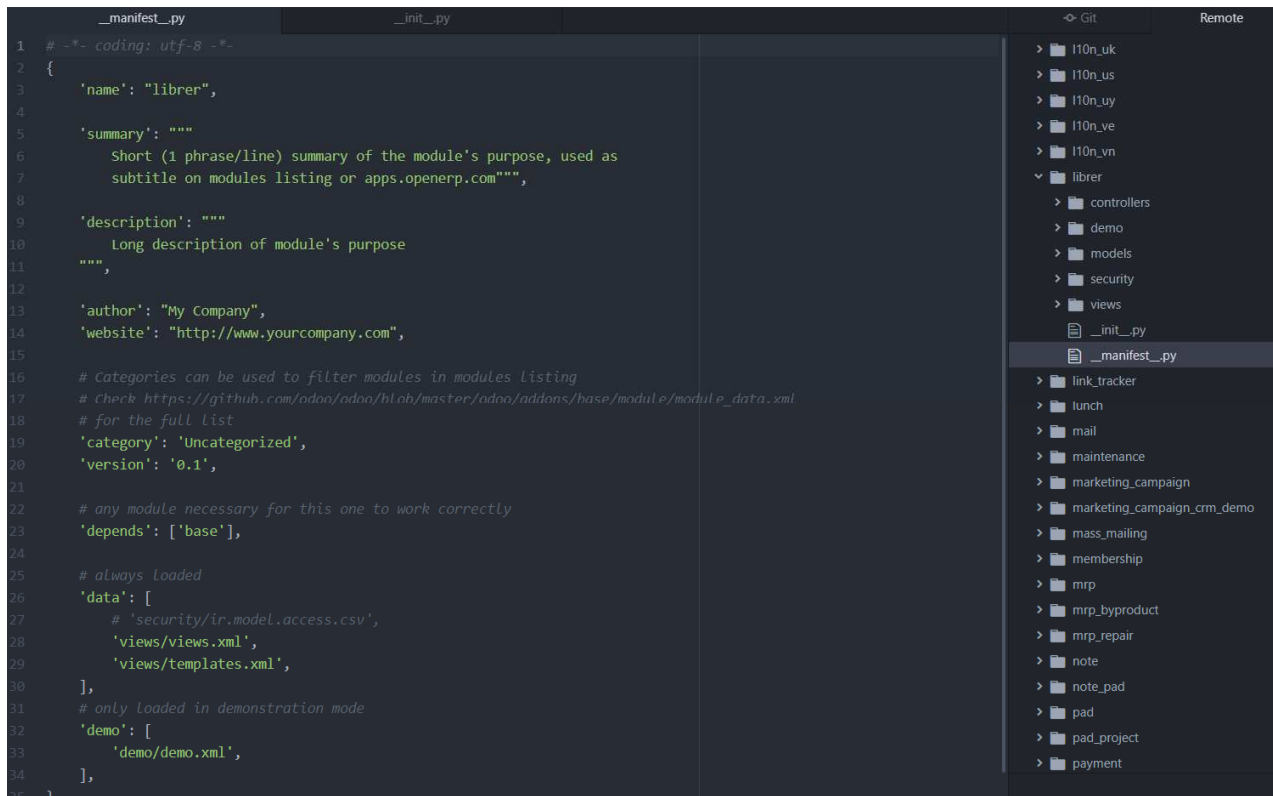
Manifest.py → describe el módulo → diccionario clave-valor que le dice a Odoo para rellenar la página principal del ERP.

Sigue el modelo modelo-vista (xml) -controlador.

Demo → tenemos los datos de demostración. (Meter 3 o 4 libros para probar). **NO ES NECESARIO.**

Security → Permisos

Manifest.py → Es el primero a editar → describe el módulo → diccionario clave-valor que le dice a Odoo para rellenar la página principal del ERP.



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

    # Escribir en varias líneas
    'summary': ""
        Módulo para gestionar una librería"",

    'description': ""
        Description larga línea 1
        Description larga línea 2
    "",

    'author': "IES JJCM",
    'website': "http://www.iesjuanjosecalvomiguel.com",

    # Categories can be used to filter modules in modules listing
    # Check https://github.com/odoo/odoo/blob/master/odoo/addons/base/module/module_data.xml
    # for the full list
    'category': 'Uncategorized', #NO TOCAMOS
    'version': '0.1', # Poner las versiones

    # any module necessary for this one to work correctly
    # IMPORTANTE
    'depends': ['base'],

    # always loaded
    'data': [#Lo dejamos como está
        # 'security/ir.model.access.csv',
        'views/views.xml',
        'views/templates.xml',
    ],
    # only loaded in demonstration mode
    'demo': [
        'demo/demo.xml',
    ],
}
```

Para probarlo mejor nos creamos una nueva base de datos:



Warning, your Odoo database manager is not protected. Please set a master password to secure it.

InformaticaSotrondio	Backup Duplicate Delete
Nautica	Backup Duplicate Delete
examen	Backup Duplicate Delete
prueba	Backup Duplicate Delete
shaila	Backup Duplicate Delete

[Create Database](#) [Restore Database](#) [Set Master Password](#)

Create Database

Odoo is up and running!
Create a new database by filling out the form, you'll be able to install your first app in a minute.

Database Name

Email

Password

Language

Country

☐ Load demonstration data (Check this box to evaluate Odoo)

[Continue](#)

Vamos a la consola y desde root reiniciamos → `systemctl restart odoo10.service`

Vamos a odoo y en configuración → activamos modo desarrollador → actualizar aplicaciones

Aplicaciones Configuración



Aplicaciones
Tienda de aplicaciones
Actualizaciones
Actualizar lista de aplicaciones
Aplicar actualizaciones programadas

Actualizar módulo

Resultado de actualización de módulo

Haga clic en Actualizar para empezar el proceso

[Actualizar](#) [Cancelar](#)

Aplicaciones
Configuración

Aplicaciones
Tienda de aplicaciones
Actualizaciones
Actualizar lista de aplicaciones
Aplicar actualizaciones programadas

Aplicaciones

Librería
Módulo para gestionar una librería
librer

Instalar

Aplicaciones
Configuración

Aplicaciones
Tienda de aplicaciones
Actualizaciones
Actualizar lista de aplicaciones
Aplicar actualizaciones programadas

Aplicaciones / Librería

Acción ▼

Librería
Por IES JJCM

Instalar

Información
Datos técnicos

Sitio web	http://www.iesjuanjosecalvomiguel.c...	Nombre técnico	librer
Categoría	Sin categoría	Licencia	LGPL versión 3
Resumen	Módulo para gestionar una librería	Última versión	10.0.0.1

Description larga línea 1 Description larga línea 2

Desarrollo Componentes Odoo 10 Modelos.

Models → models.py → van declarados ahí las tablas de la BBDD.

Cada objeto en Odoo corresponde con una clase Python → class librería_categoria(models.Model)

(Lo podemos nombrar: nombre del modulo y nombre del objeto).

Todas las clases de Odoo extienden siempre de Model.

Todas las clases de Odoo tienen un campo privado name → _name = "nombredelmodulo.nombredelobjeto"

("librería_categoria").

Siempre tener OTRO campo name para facilitar las búsquedas → name =

fields.Char(string="Nombre", requerido=True, help="Introduce el nombre de la categoría")

C:\Users\Shaila\Documents\DAM\2 DAM\SGE\CHULETA SGE\models.py (prueba) - Sublime Text (UNREGISTERED)

File Edit Selection Find View Goto Tools Project Preferences Help

OPEN FILES
models.py

FOLDERS
prueba
e2e
node_modules
src
app
add-usuario
detalles-usuario
hello-world
lista-usuarios
modelo
usuario
usuario.component.css
usuario.component.html
usuario.component.ts
app-routing.module.ts
app.component.css
app.component.html
app.component.spec.ts
app.components.ts
app.module.ts
servicio-usuarios.service.spec.ts
servicio-usuarios.service.ts
assets
environments
browserslist
favicon.ico
index.html
karma.conf.js
main.ts

```

1  # -*- coding: utf-8 -*-
2  #Primero hay que tocar manifest
3  from odoo import models, fields, api
4
5  class libreria_categoria(models.Model):
6      _name = 'libreria_categoria'
7
8      #Vamos a crear los campos
9      #siempre tiene que tener un campo name OBLIGATORIO, el resto de los campos da igual el nombre
10     name = fields.Char(string="Nombre categoría", help="Introduce el nombre de la categoría", required=True)
11     #poner parametro help --> salta la ayuda
12     #required=True --> campo requerido, si no lo ponemos creemos que va en false
13     descripcion = fields.Text(string="Descripción", help="Introduce la descripción de la categoría", required=True)
14
15     # value = fields.Integer()
16     # value2 = fields.Float(compute="_value_pc", store=True)
17     # descripcion = fields.Text()
18
19     # @api.depends('value')
20     # def _value_pc(self):
21     #     self.value2 = float(self.value) / 100
22
23     class libreria_libro(models.Model):
24         _name = 'libreria.libro'
25
26         #Vamos a crear los campos
27         #siempre tiene que tener un campo name OBLIGATORIO, el resto de los campos da igual el nombre
28         name = fields.Char(string="Titulo", help="Introduce el titulo del libro", required=True) #string="Nombre"
29         #poner parametro help --> salta la ayuda
30         #required=True --> campo requerido, si no lo ponemos creemos que va en false
31         precio = fields.Float(string="Precio", help="Introduce el precio del libro", required=True) #string="Nombre"

```

Line 2, Column 32

Tab Size: 4 Python

Create Database

Odoo is up and running!
Create a new database by filling out the form, you'll be able to install your first app in a minute.

Database Name

Email

Password

Language

Country

ANTES DE TODO → vamos a Putty → si estamos con el usuario odoo → exit → vamos a root → systemctl restart odoo10.service

Vamos a configuración → activamos modo desarrollador → actualizamos lista de aplicaciones → instalamos nuestro módulo

The screenshot shows the Odoo 'Aplicaciones' (Applications) page. On the left is a sidebar with the Odoo logo and links: 'Aplicaciones', 'Tienda de aplicaciones', 'Actualizaciones', 'Actualizar lista de aplicaciones', 'Aplicar actualizaciones programadas'. The main area is titled 'Aplicaciones' and contains a card for the 'practicasfct' module. The card shows a 3D cube icon, the module name 'practicasfct', a description 'Módulo para gestionar alumnos y empresas de prácticas.', and an 'Instalar' (Install) button. A search bar at the top right shows 'Módulo pract' and 'Buscar...'. The top navigation bar has 'Aplicaciones' and 'Configuración' tabs.

Para comprobarlo:

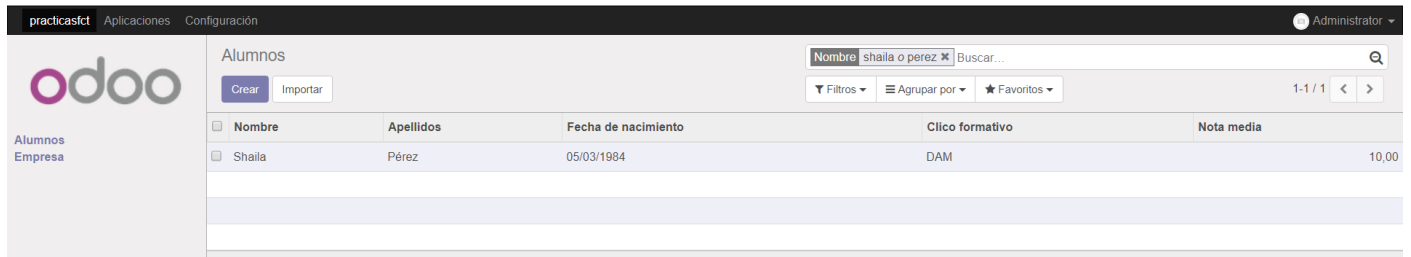
The screenshot shows the Odoo 'Modelos' (Models) page. The sidebar on the left has links: 'Tablero', 'Usuarios', 'Usuarios', 'Grupos', 'Interfaz de usuario', 'Estructura de la base d...', and 'Modelos' (highlighted). The main area is titled 'Modelos' and shows a table of models created by the 'practicasfct' module. The table has columns: 'Modelo', 'Descripción del modelo', 'Tipo', and 'Modelo transitorio'. There are two models listed: 'practicasfct alumno' and 'practicasfct empresa', both of type 'Objeto base'. A search bar at the top right shows 'Modelo pract' and 'Buscar...'. The top navigation bar has 'Aplicaciones' and 'Configuración' tabs.

ASÍ BIEN:

The first screenshot shows the 'Alumnos' (Students) view in the 'practicasfct' module. The sidebar on the left has links: 'Alumnos' and 'Empresa'. The main area is titled 'Alumnos' and shows a table with columns: 'Nombre', 'Apellidos', 'Fecha de nacimiento', 'Clico formativo', and 'Nota media'. There is one record: 'Shaila Pérez' with birth date '05/03/1984' and a note of '10,00'. A search bar at the top right shows 'Buscar...'. The top navigation bar has 'practicasfct', 'Aplicaciones', and 'Configuración' tabs.

The second screenshot shows the same 'Alumnos' view but with a filter applied. A dropdown menu is open, showing 'notamedia' as the selected filter. The table now shows the same record, but the 'Nota media' column is highlighted. The top navigation bar has 'practicasfct', 'Aplicaciones', and 'Configuración' tabs.

FILTRAR POR DOS CAMPOS:



Configuración de Vistas en Odoo 10.

Van En la carpeta Views

Vamos a crear vistas de tipo árbol y formulario

Creación de Relaciones en Odoo 10.

```
1 # coding: utf-8
2
3 from odoo import models, fields, api
4
5 class practicasfct_alumno(models.Model):
6     _name = 'practicafct.alumno'
7
8     name = fields.Char(string="Nombre", help="Introduce el nombre del alumno", required=True)
9     apellidos = fields.Char(string="Apellidos", help="Introduce los apellidos del alumno", required=True)
10    fechanacimiento = fields.Date(string="Fecha de nacimiento", help="Introduce la fecha de nacimiento", required=True)
11    cicloformativo = fields.Selection([('0', 'DAM'), ('1', 'DAW'), ('2', 'ASIR')], default='0', string="Ciclo formativo", help="Selecciona el ciclo que estás cursando")
12
13    notamedia = fields.Float(string="Nota media", help="Introduce la nota media del alumno.", required=False)
14    #cadena de texto, campo calculado como se indica más abajo
15    #notamediatexto = fields.Float(string="Nota media", help="Introduce la nota media del alumno", required=False)
16    #empresa
17    #manyToOne un estudiante solo puede tener una empresa de prácticas, (contra objeto que queremos la relacion, nombre, campo requerido, que pasa cuando se elimina el objeto)
18    #tira sobre name, por eso hay que ponerlo siempre
19    empresa = fields.Many2one("practicafct.empresa", string="Empresa", required=True, ondelete="cascade")
20
21    # @api.depends('value')
22    # def _value_pc(self):
23    #     self.value2 = float(self.value) / 100
24
25 class practicasfct_empresa(models.Model):
26     _name = 'practicafct.empresa'
27
28     name = fields.Char(string="Nombre", help="Introduce el nombre de la empresa.", required=True)
29     direccion = fields.Char(string="Dirección", help="Introduce la dirección de la empresa.", required=False)
30     #ListaAlumnos
31     #muchos a uno, aquí si hay que poner contra el campo con el que lo lanzamos
32     alumnos = fields.One2Many("practicafct.alumno", "Empresa", string="Alumnos")
```

```
7 <field name="name">Alumno</field>
8 <field name="model">practicafct.alumno</field>
9 <field name="arch" type="xml">
10     <tree>
11         <field name="name"/>
12         <field name="apellidos"/>
13         <field name="fechanacimiento"/>
14         <field name="cicloformativo"/>
15         <field name="notamedia"/>
16     </tree>
17 </field>
18 </record>
19
20 <record model="ir.ui.view" id="practicafct.alumno_form">
21     <field name="name">Alumno</field>
22     <field name="model">practicafct.alumno</field>
23     <field name="arch" type="xml">
24         <form>
25             <group>
26                 <field name="name"/>
27                 <field name="apellidos"/>
28                 <field name="fechanacimiento"/>
29                 <field name="cicloformativo"/>
30                 <field name="notamedia"/>
31                 <field name="empresa"/>
32             </group>
33         </form>
34     </field>
35 </record>
36
37
38 <record model="ir.ui.view" id="practicafct.empresa_tree">
39     <field name="name">Empresas</field>
40     <field name="model">practicafct.empresa</field>
41     <field name="arch" type="xml">
42         <tree>
43             <field name="name"/>
44             <field name="direccion"/>
45             <field name="alumnos"/>
46         </tree>
47     </field>
48 </record>
```

#**muchos a uno**, aquí si hay que poner contra el campo con el que lo lanzamos --> empresa como el parametro que pusimos antes en las relaciones → estaríamos en el campo empresa → aquí si que hay que indicar contra que campo tira → en este caso empresa

`alumnos = fields.One2many("practicafct.alumno", "empresa", string="Alumnos")`

#Many2one un estudiante solo puede tener una empresa de prácticas, (contra objeto que queremos la **relación**, nombre → que va a salir en la vista, campo requerido → un alumno siempre va a tener que tener una empresa de prácticas, que pasa cuando se borra)
#tira sobre name, por eso hay que ponerlo siempre → estaríamos en el campo alumno → tira contra el campo name,
#por eso no lo hemos puesto

```
empresa = fields.Many2one("practicassfct.empresa",string="Empresa", required=True,onDelete="cascade",
help="Selecciona la empresa de prácticas.")
```

Ahora iríamos a las vistas y pondríamos en la de formulario un campo más con el nombre que tenemos antes puesto:

```
<record model="ir.ui.view" id="practicassfct.empresa_tree">
  <field name="name">Empresa</field>
  <field name="model">practicassfct.empresa.tree</field>
  <field name="arch" type="xml">
    <tree>
      <field name="name"/>
      <field name="direccion"/>
      <field name="alumnos"/>
    </tree>
  </field>
</record>

<record model="ir.ui.view" id="practicassfct.empresa_form">
  <field name="name">Empresa</field>
  <field name="model">practicassfct.empresa.form</field>
  <field name="arch" type="xml">
    <form>
      <group colspan="2" col="2">
        <field name="name"/>
        <field name="direccion"/>
        <field name="alumnos"> <!--Esta es la que tengo que quitar para que no se me muestre como
coleccion -->
<!--Lo dejamos abierto y metemos una vista tree igual no va aquí -->
          <tree>
            <field name="name"/>
            <field name="apellidos"/>
          </tree>
        </field>
      </group>
    </form>
  </field>
</record>
```

Vistas de búsqueda

ACUERDATE EN ODOO DARLE A LA LUPA PARA FILTRAR

Vamos a vistas:

```
<record model="ir.ui.view" id="practicassfct.alumno_search"> <!--Pablo pone practicassfct.alumno_search_view-->
  <field name="name">Alumnos</field> <!--Pablo pone practicassfct.alumno.search -->
  <field name="model">practicassfct.alumno</field> <!--Modelo sobre el que tiramos -->
  <field name="arch" type="xml">
    <search>
      <field name="name" string="Nombre"/> <!--Campos por los que buscar-->
      <field name="apellidos" string="Apellidos"/> <!-- &lt; o &gt; son los códigos XML para <-->
      <filter name="notamedia" domain="[('notamedia', '&gt;', 9)]"/> <!-- ponemos un filtro!! escribir un
dominio en el que el primer campo será el campo sobre el que se escribe la condición, luego menor igual o mayor
OJO ENTRE COMILLAS SIMPLES y luego por lo que se quiere buscar, en este caso nota media mayor igual que 9 --
    >
      </search>
    </field>
  </record>

<record model="ir.ui.view" id="practicassfct.empresa_search">
```

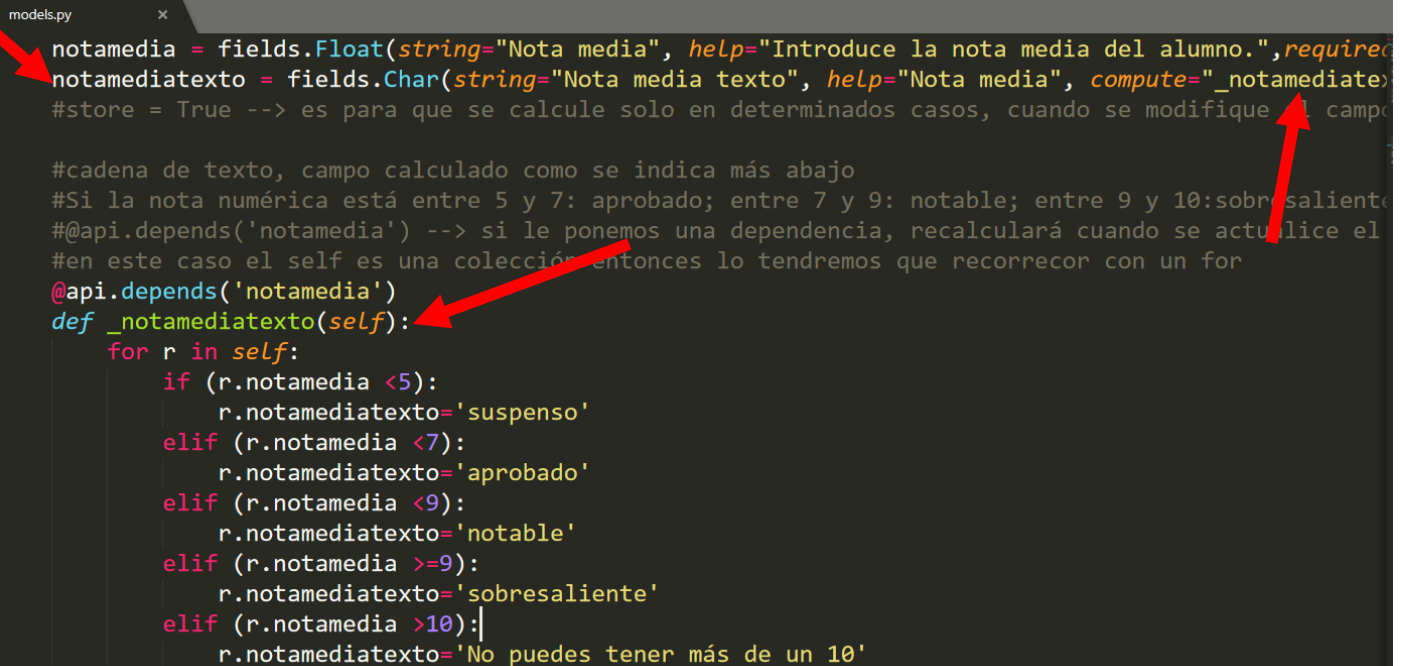
```

<field name="name">Empresa</field>
<field name="model">practicafct.empresa</field> <!--Modelo sobre el que tiramos -->
<field name="arch" type="xml">
  <search>
    <field name="name" string="Nombre"/> <!--Campos por los que buscar-->
    <!--<filter name="baratos" dominio="['precio','&lt;=';5)']"/>-->
  </search>
</field>
</record>

```

Campos calculados en Odoo 10.

Para ello vamos a models → creamos el campo calculado → notamediaindex = ponemos field y el tipo de dato que nos va a “devolver”, en este caso es un Char, en el de librería un float



```

models.py
notamedia = fields.Float(string="Nota media", help="Introduce la nota media del alumno.", required=True)
notamediaindex = fields.Char(string="Nota media texto", help="Nota media", compute="_notamediaindex", store=True)
#store = True --> es para que se calcule solo en determinados casos, cuando se modifique el campo

#cadena de texto, campo calculado como se indica más abajo
#Si la nota numérica está entre 5 y 7: aprobado; entre 7 y 9: notable; entre 9 y 10: sobresaliente
@api.depends('notamedia') --> si le ponemos una dependencia, recalculará cuando se actualice el campo
#en este caso el self es una colección entonces lo tendremos que recorrer con un for
@api.depends('notamedia')
def _notamediaindex(self):
    for r in self:
        if (r.notamedia <5):
            r.notamediaindex='suspense'
        elif (r.notamedia <7):
            r.notamediaindex='aprobado'
        elif (r.notamedia <9):
            r.notamediaindex='notable'
        elif (r.notamedia >=9):
            r.notamediaindex='sobresaliente'
        elif (r.notamedia >10):
            r.notamediaindex='No puedes tener más de un 10'

```

EJEMPLO LIBERÍA:

```

importetotal = fields.Float(string="Importe total", help="Importe total", compute="_importetotal",store=True)
#store = True --> es para que se calcule solo en determinados casos, cuando se modifique el campo
#compute, lo ponemos privado con un guion bajo, solo uno, es lo que llevará luego el decorador con la función
#calculada → va a ser un método que va a recibir un self, el objeto

```

```

#cadena de texto, campo calculado como se indica más abajo
#Si la nota numérica está entre 5 y 7: aprobado; entre 7 y 9: notable; entre 9 y 10: sobresaliente.
#@api.depends('precio','ejemplares')--> si le ponemos una dependencia, recalculará cuando se actualice el campo
#entre paréntesis, cuando debe llamarse, es decir si cambiar los valores de los datos por los que se calcula
#en este caso el self es una colección entonces lo tendremos que recorrer con un for
@api.depends('precio','ejemplares')
def _importetotal(self):
    for r in self:
        r.importetotal = r.ejemplares*r.precio

```

Luego metemos el campo en la vista

```

<field name="importetotal"/>

```

Informes en Odoo 10.

Meter un directorio nuevo → reports y un xml → reports_libro.xml → luego habrá que ir al Manifest y meterlo
OJO CON EL DIV PAG → HAY QUE SACARLO DEL FOR

```
<odoo>
<data>
  <record id="ejemplo" model="report.paperformat">
    <field name="name">ejemplo</field>
    <field name="default" eval="True"/>
    <field name="orientation">Portrait</field>
    <field name="header_line" eval="False"/>
  </record>
  <report
    id="report_empresa"
    model="practicafct.empresa"
    string="Informe empresa/s"
    name="practicafct.report_empresa_view"
    file="practicafct.report_empresa_view"
    report_type="qweb-pdf"
    paperformat="ejemplo"/>

  <template id="practicafct.report_empresa_view">
    <t t-call="report.html_container">
      <t t-call="report.external_layout">
        <div class="page"> <!-- OJO -- >
          <t t-foreach="docs" t-as="empresa">
            <h2 t-field="empresa.name" />
            <div>
              <strong>Direccion empresa:</strong>
              <span t-field="empresa.direccion"/>
            </div>
            <div>
              <strong>ALUMNOS</strong>
              <t t-foreach="empresa.alumnos" t-as="alumno">
                <h5>-----</h5>
                <p><strong>Nombre: </strong><span t-field="alumno.name"/></p>
                <p><strong>Apellidos: </strong><span t-field="alumno.apellidos"/></p>
                <p><strong>Fecha nacimiento: </strong><span t-field="alumno.fechanacimiento"/></p>
                <p><strong>Ciclo formativo: </strong> <span t-field="alumno.cicloformativo"/></p>
                <p><strong>Nota media: </strong> <span t-field="alumno.notamedia"/></p>
              </t>
            </div>
          </t>
        </div>
      </t>
    </template>
  </data>
</odoo>
```

```

port_empresa.xml x _manifest_.py x
'author': "Shaila",
'website': "http://www.shaila.com",

# Categories can be used to filter modules in modules listing
# Check https://github.com/odoo/odoo/blob/master/odoo/addons/base/module/module_data.xml
# for the full list
'category': 'Uncategorized',
'version': '0.1',

# any module necessary for this one to work correctly
'depends': ['base', 'report']

# always loaded
'data': [
    'security/security.xml',
    'security/ir.model.access.csv',
    'views/views.xml',
    'views/templates.xml',
    'reports/report_empresa.xml',
    'data/data.xml'
],
# only loaded in demonstration mode
'demo': [
    'demo/demo.xml',
]

```

Permisos en Odoo 10.

Vamos a security → creamos xml security.xml

```

report_empresa.xml x _manifest_.py x security.xml x
<odoo>
  <data>
    <!-- Se utiliza para definir los grupos -->
    <!-- Escribimos un record por cada grupo -->
    <!-- model siempre es lo mismo, el id es contra que el que vamos a hacer los grupos -->
    <!-- Tenemos que definir una entrada por cada grupo o rol -->

    <record id="practicasfct_coordinador" model="res.groups">
      <field name="name"> Practicas / Coordinador</field>
    </record>

    <record id="practicasfct_profesor" model="res.groups">
      <field name="name">Practicass / Profesor</field>
    </record>

  </data>
</odoo>

```

Para establecer los permisos vamos al CSV

```

security.xml x ir.model.access.csv x
1 id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
2 practicasfct_coordinador_alumno,practicassfct coordinador
  alumno,model_practicassfct_alumno,practicassfct_coordinador,1,1,1,1
3 practicasfct_coordinador_empresa,practicassfct coordinador
  empresa,model_practicassfct_empresa,practicassfct_coordinador,1,1,1,1
4
5 practicasfct_profesor_alumno,practicassfct profesor
  alumno,model_practicassfct_alumno,practicassfct_profesor,1,0,0,0
6 practicasfct_profesor_empresa,practicassfct profesor
  empresa,model_practicassfct_empresa,practicassfct_profesor,1,0,0,0
7

```


La primera línea no la tocamos

1º) practicasfct_coordinador_alumno → sería el id, no puede llevar espacios y debe ser ÚNICO

2º) practicasfct coordinador alumno → sería el nombre para la regla, puede llevar espacios

3º) model_practicasfct_alumno → IMPORTANTE: objeto o tabla sobre la que dar permisos (los que creamos en el models.py → en el video de Pablo model_libreria_categoria)

4º) practicasfct_coordinador → IMPORTANTE: grupo sobre el que otorgamos los permisos, lo que hemos puesto en el id de security.xml.

5º) 1,1,1,1 → permisos → lectura, escritura, creación y borrado, por ese orden.

Ahora IMPORTANTE, vamos al manifest:

```
port_empresa.xml x _manifest_.py x

'author': "Shaila",
'website': "http://www.shaila.com",

# Categories can be used to filter modules in modules listing
# Check https://github.com/odoo/odoo/blob/master/odoo/addons/base/module/module_data.xml
# for the full list
'category': 'Uncategorized',
'version': '0.1',

# any module necessary for this one to work correctly
'depends': ['base', 'report'],

# always loaded
'data': [
    'security/security.xml',
    'security/ir.model.access.csv',
    'views/views.xml',
    'views/templates.xml',
    'reports/report_empresa.xml',
    'data/data.xml'
],
# only loaded in demonstration mode
'demo': [
    'demo/demo.xml',
]
```

Tienen que estar en ese orden para que no de problemas.

Ahora iremos a los menús y ponemos para que grupos tiene sentido que los muestre:

```
views.xml x

    }
  </field>
</record>
-->

<!-- Top menu item -->
<!-- No le ponemos padre por lo que saldrá en la parte de arriba -->
<menuitem name="practicasfct" id="practicasfct.menu_root" groups="
practicasfct_coordinador,practicasfct_profesor"/>

<!-- menu categories parent="practicasfct.menu_root" para que tenga padre-->

<menuitem name="Alumnos" id="practicasfct.alumno_menu" action="
practicasfct.alumno_action_window" parent="practicasfct.menu_root" groups="
practicasfct_coordinador,practicasfct_profesor"/>
<menuitem name="Empresa" id="practicasfct.empresa_menu" action="
practicasfct.empresa_action_window" parent="practicasfct.menu_root" groups="
practicasfct_coordinador,practicasfct_profesor"/>

<!-- actions -->
<!--
<menuitem name="List" id="practicasfct.menu_1_list" parent="practicasfct.menu_1"
action="practicasfct.action_window"/>
<menuitem name="Server to list" id="practicasfct" parent="practicasfct.menu_2"
action="practicasfct.action_server"/>
-->
</data>
</odoo>
```

OJO!!!! Si queremos poner más de un grupo debe ir entre comas.

Para comprobarlo vamos al módulo, vamos → Configuración → Usuarios → Admon → Editar → y darle los permisos.
F5

Retoques finales en Odoo 10. Poner Icono y datos precargados.

Icono en Odoo 10.

Debemos crear una carpeta vamos a WinSCP → odoo/odoo/addons/nombredelmodulo → crear dos carpetas que se tienen que llamar por narices así:

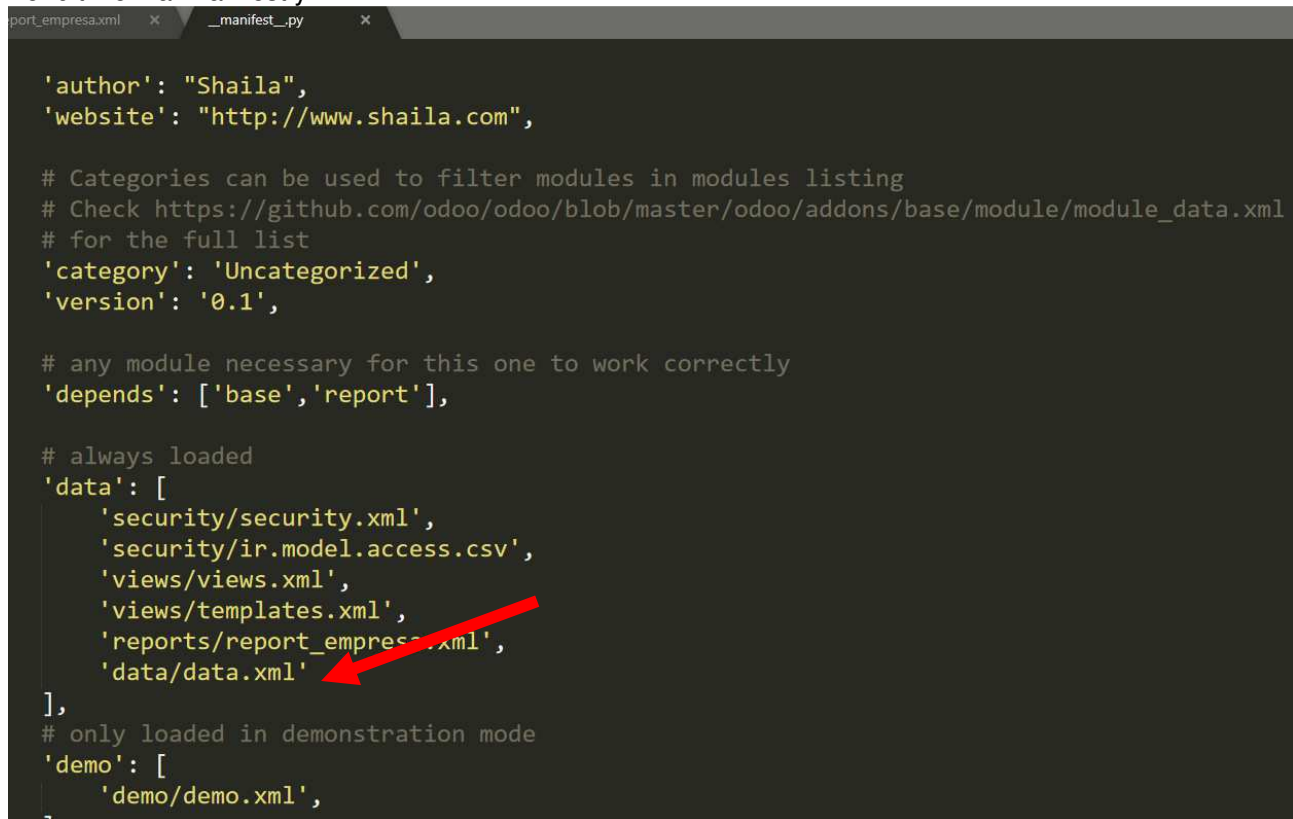
static → description → ya podemos meter nuestro icono → OJO TIENE QUE LLAMARSE POR NARICES
→ icon.png → para que funcione.

Datos precargados en Odoo 10.

Crear en atom un directorio que se llame data y un archivo que sea data.xml. OJO hay que poner los datos requeridos para que no casque.

```
<odoo>
<data>
  <record model="practicafct.alumno">
    <field name="name">Shaila</field>
    <field name="apellidos">Perez</field>
    <field name="fechanacimiento">05/03/1984</field>
    <field name="ciclo">0</field>
    <field name="notaMedia">9.00</field>
    <field name="notaMediaTexto">Sobresaliente</field>
  </record>
  <record model="practicafct.empresa">
    <field name="name">Alcampo</field>
  </record>
</data>
</odoo>
```

Por último ir al manifest y:



```
port_empresa.xml x _manifest_.py x

'author': "Shaila",
'website': "http://www.shaila.com",

# Categories can be used to filter modules in modules listing
# Check https://github.com/odoo/odoo/blob/master/odoo/addons/base/module/module_data.xml
# for the full list
'category': 'Uncategorized',
'version': '0.1',

# any module necessary for this one to work correctly
'depends': ['base', 'report'],

# always loaded
'data': [
  'security/security.xml',
  'security/ir.model.access.csv',
  'views/views.xml',
  'views/templates.xml',
  'reports/report_empresa.xml',
  'data/data.xml'
],
# only loaded in demonstration mode
'demo': [
  'demo/demo.xml',
]
```


Si queremos para que se cargue siempre ir a demo.xml

```
<odoo>
<data>
  <record id="empresa1" model="practicafct.empresa">
    <field name="name">Informatica Sotrondio</field>
    <field name="direccion">Calle Principal nº7 Bajo</field>
  </record>
  <record id="empresa2" model="practicafct.empresa">
    <field name="name">Informatica Langreo</field>
    <field name="direccion">Avenida Principal nº55 Bajo</field>
  </record>
  <record id="alumno1" model="practicafct.alumno">
    <field name="name">Pedro</field>
    <field name="apellidos">Fernandez Fernandez</field>
    <field name="fechanacimiento">10/11/2009</field>
    <field name="empresa" ref="empresa1"></field>
  </record>
  <record id="alumno2" model="practicafct.alumno">
    <field name="name">Jose</field>
    <field name="apellidos">Menendez Rodriguez</field>
    <field name="fechanacimiento">11/11/1997</field>
    <field name="empresa" ref="empresa1"></field>
  </record>
  <record id="alumno3" model="practicafct.alumno">
    <field name="name">Ana</field>
    <field name="apellidos">Perez Perez</field>
    <field name="fechanacimiento">10/12/1994</field>
    <field name="empresa" ref="empresa2"></field>
  </record>
  <record id="alumno4" model="practicafct.alumno">
    <field name="name">Eva</field>
    <field name="apellidos">Arias Romero</field>
    <field name="fechanacimiento">11/11/1994</field>
    <field name="empresa" ref="empresa2"></field>
  </record>
</data>
</odoo>
```