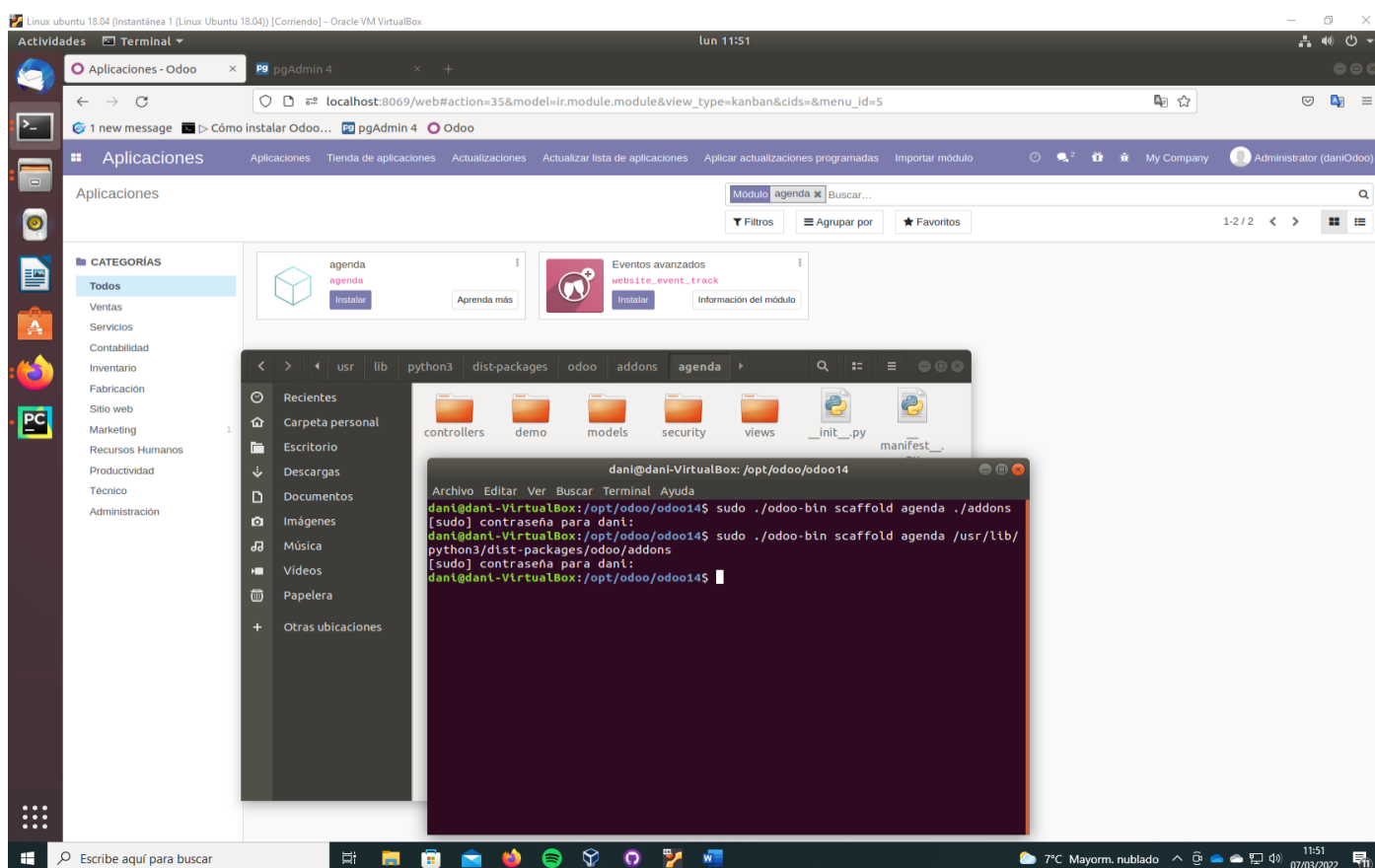


En esta tarea se nos pide crear un módulo que gestione una agenda telefónica con unas características dadas y haciendo uso del paradigma modelo-vista-controlador.

Para la creación del módulo que se nos pide en esta tarea he seguido los pasos que vienen en la documentación oficial de Odoo que se encuentra en la siguiente dirección: <https://www.odoo.com/documentation/14.0/es/developer/howtos/backend.html>

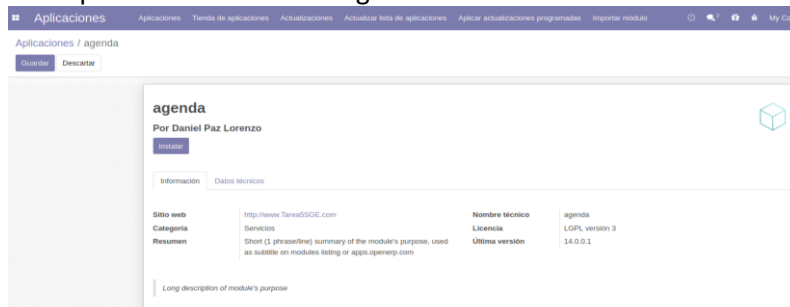
Controlador

Comenzamos creando el controlador. Abrimos la terminal y nos dirigimos a la carpeta donde tenemos instalado Odoo, desde ahí vamos a crear la estructura base de carpetas y ficheros haciendo uso del comando Scaffold (*\$ odoo-bin scaffold <module name> <where to put it>*):



Después de ejecutar la sentencia en la terminal podemos observar que dentro de la carpeta addons se nos ha creado la carpeta llamada "agenda" que contiene los archivos básicos del módulo, también podemos ver si abrimos Odoo y actualizamos lista de aplicaciones, que ya reconoce "agenda" como nuevo módulo

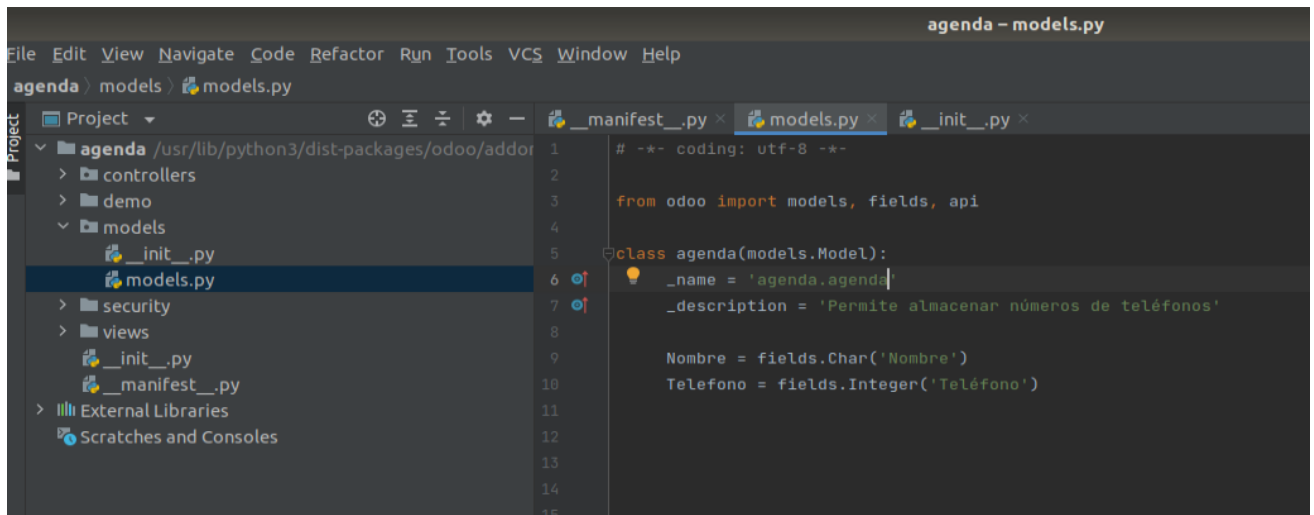
Modificamos un poco el archivo manifest del módulo para que incluya información del mismo, si abrimos desde Odoo la info del módulo podemos observar lo siguiente:



Modelo

Ahora que tenemos creado el módulo nos disponemos a añadirle los campos: Nombre y Teléfono e incorporar el nuevo objeto creado en la base de datos de Odoo:

1. Haciendo uso del IDE “PyCharm”, como en tareas anteriores, abro como proyecto la carpeta agenda dentro de addons de Odoo. Me voy al navegador de la aplicación y abro el fichero de nombre “models.py”. Modificamos el fichero que nos quedará así:



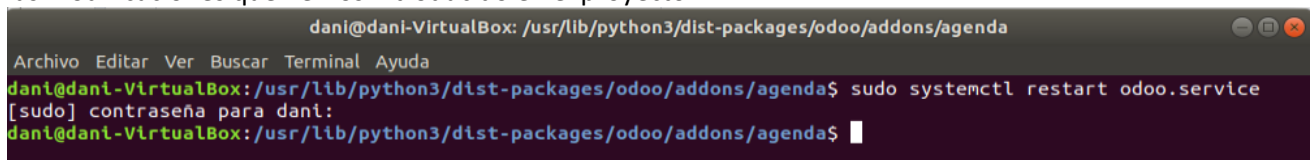
```

1  # -*- coding: utf-8 -*-
2
3  from odoo import models, fields, api
4
5  class agenda(models.Model):
6      _name = 'agenda.agenda'
7      _description = 'Permite almacenar números de teléfonos'
8
9      Nombre = fields.Char('Nombre')
10     Telefono = fields.Integer('Teléfono')
11
12
13
14
15

```

Importamos las clases base models, fields y api. Creamos una nueva clase llamada “agenda” y dentro de esta añadimos los dos nuevos campos (fields): “Nombre” y “Agenda”. Guardamos los cambios

2. Antes de abrir Odoo nuevamente para actualizar el módulo reseteamos el servidor de odoo para que coja bien las modificaciones que hemos introducido en el proyecto:

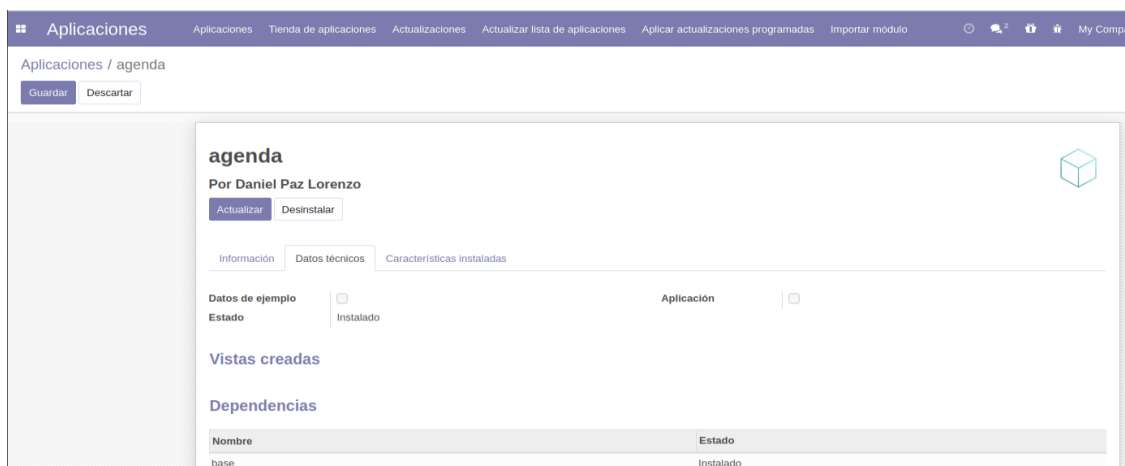


```

dani@dani-VirtualBox: /usr/lib/python3/dist-packages/odoo/addons/agenda
Archivo Editar Ver Buscar Terminal Ayuda
dani@dani-VirtualBox: /usr/lib/python3/dist-packages/odoo/addons/agenda$ sudo systemctl restart odoo.service
[sudo] contraseña para dani:
dani@dani-VirtualBox: /usr/lib/python3/dist-packages/odoo/addons/agenda$

```

Ahora ya nos disponemos a actualizar el módulo que previamente habíamos instalado y que no contenía nada:



3. Una vez actualizado vamos a comprobar que se nos ha creado bien el nuevo objeto junto con los campos:

Modelos / Permite almacenar números de teléfonos

Editar Crear Imprimir Acción

Descripción del modelo: Permite almacenar números de teléfonos. Tipo: En las aplicaciones. Objeto base: agenda.

Nombre de campo	Etiqueta de campo	Tipo de campo	Requerido	Sólo lectura	Indexado	Tipo
Nombre	Nombre	Carácter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
Telefono	Teléfono	entero	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
_last_update	Last Modified on	Fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
create_date	Created on	Fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
create_uid	Created by	many2one	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
display_name	Display Name	Carácter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
id	ID	entero	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
write_date	Last Updated on	Fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
write_uid	Last Updated by	many2one	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base

Nos dirigimos a Ajustes/Técnico/Modelos y buscamos “agenda”, lo abrimos y podemos ver que se nos han creado correctamente los nuevos campos

Solo nos queda ya comprobar que el objeto agenda ha sido incorporado a la base de datos como una tabla dentro del esquema de Odoo:

Actividades Navegador web Firefox

pgAdmin 4 Permite almacenar núm...

localhost/browser/#

1 new message pgAdmin 4 Cómo instalar Odoo... pgAdmin 4 Odoo Building a Module — Zona de Usuarios

pgAdmin Archivo Objeto Herramientas Ayuda

Navegador Tablero Propiedades SQL Estadísticas Dependencias Dependientes

agenda_agenda

id

Nombre

Telefono

create_uid

create_date

write_uid

write_date

Disparadores

RLS Policies

Reglas

Restricciones

Indice

Nombre

id

Nombre

Telefono

create_uid

create_date

write_uid

write_date

Comentario

Nombre

Teléfono

Created by

Created on

Last Updated by

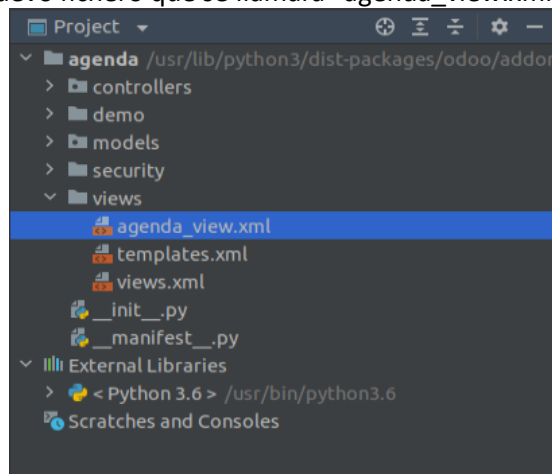
Last Updated on

Abrimos en una nueva ventana del navegador nuestro administrador “PgAdmin” que tiene conexión con la bbdd de PostgreSQL de Odoo y a través del navegador buscamos la nueva tabla creada “agenda” que vemos efectivamente se ha creado junto con las columnas “Nombre”, “Teléfono” y algunas otras que nos ha generado Odoo automáticamente

Vista

La última parte de la creación del módulo consiste en crear la vistas y menús con los que va a interactuar el usuario.

Lo primero que haremos será crear el **menú** que enlace al objeto. En nuestro proyecto de agenda, dentro de la carpeta “view” creamos un nuevo fichero que se llamará “agenda_view.xml”:



Abrimos el xml que hemos creado y lo editamos de la siguiente manera para que se añada un menú a las vistas de la aplicación:

```

1 <!-- DEFINICIÓN DE MENÚ -->
2 <!-- actions opening views on models -->
3 <record model="ir.actions.act_window" id="agenda.action_window">
4   <field name="name">Agenda Telefónica</field>
5   <field name="res_model">agenda.agenda</field>
6   <field name="view_mode">tree,form</field>
7 </record>
8
9 <!-- Top menu item -->
10 <menuitem name="Agenda" id="agenda.menu_root"/>
11
12 <!-- menu categories -->
13 <menuitem name="Agenda Telefónica" id="agenda.menu_agenda_tel" parent="agenda.menu_root" action="agenda.action_window"/>
14
15 </data>
16 </odoo>
  
```

Le estamos diciendo que cree un menú principal que se llame “Agenda” del que partirá un submenú llamado “Agenda telefónica” que a su vez tiene como atributo un action_window del tipo objeto agenda.agenda con vistas tree y form

Ya tenemos el menú ahora vamos a configurar la vista de **árbol** dentro del mismo fichero de la siguiente manera:

```

1  <!-- Definición de menú -->
2  <!-- actions opening views on models -->
3  <record model="ir.actions.act_window" id="agenda.action_window">
4    <field name="name">Agenda Telefónica</field>
5    <field name="res_model">agenda.agenda</field>
6    <field name="view_mode">tree,form</field>
7  </record>
8
9  <!-- Top menu item -->
10 <menuitem name="Agenda" id="agenda.menu_root"/>
11
12 <!-- menu categories -->
13 <menuitem name="Agenda Telefónica" id="agenda.menu_agenda_tel" parent="agenda.menu_root" action="agenda.action_window"/>
14
15 <!-- VISTA ÁRBOL DE AGENDA -->
16 <record model="ir.ui.view" id="agenda.list_view">
17   <field name="name">agenda.telefonica.view.tree</field>
18   <field name="model">agenda.agenda</field>
19   <field name="arch" type="xml">
20     <tree>
21       <field name="Nombre"/>
22       <field name="Telefono"/>
23     </tree>
24   </field>
25 </record>
26
27 </data>
28 </odoo>

```

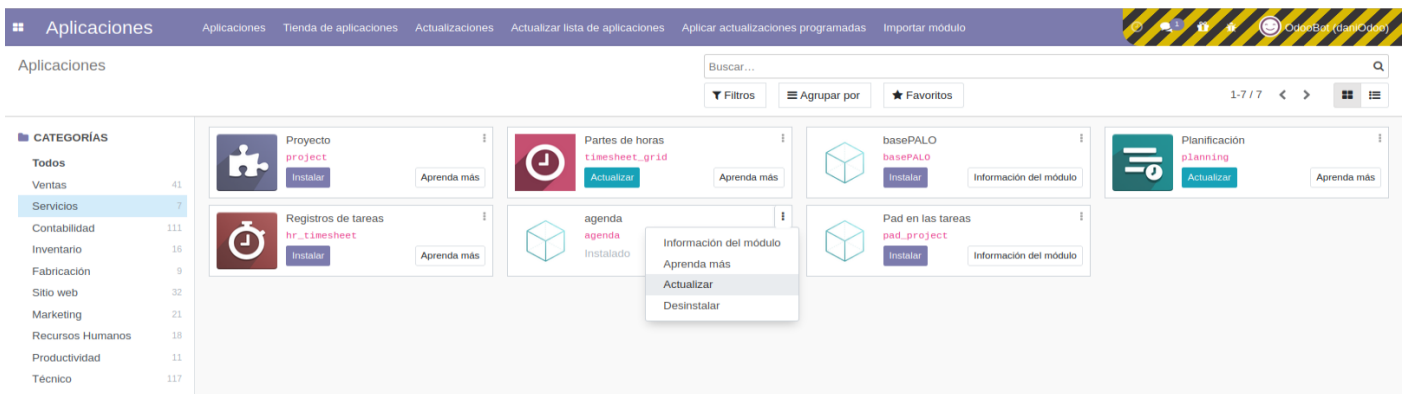
Dentro de las etiquetas `<record>` seleccionamos el nombre que tendrá la vista, al modelo que pertenece (`agenda.agenda`) y la arquitectura que le decimos es del tipo árbol con las etiquetas `<tree>` y los campos `Nombre` y `Teléfono`

Ahora que ya tenemos configurada una primera vista, abrimos Odoo y actualizamos la aplicación para comprobar que todo funciona como esperamos, pero antes tenemos que añadir al fichero `_manifest.py` la vista que acabamos de crear para que la incluya en el módulo:

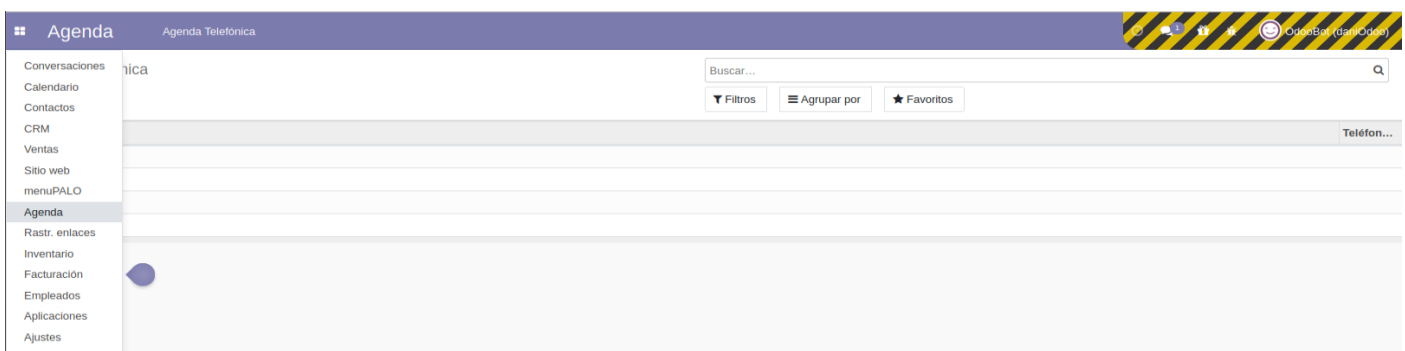
```

1  # -*- coding: utf-8 -*-
2  {
3    'name': 'Agenda',
4    'summary': 'Agenda telefónica',
5    'description': """Módulo que sirve para la gestión de la agenda telefónica""",
6    'author': "Daniel Paz Lorenzo",
7    'website': "http://www.Tarea5SGE.com",
8
9    # Categories can be used to filter modules in modules listing
10   # Check https://github.com/odoo/odoo/blob/14.0/odoo/addons/base/data/ir_module_category_data.xml
11   # for the full list
12   'category': 'Services',
13   'version': '0.1',
14
15   # any module necessary for this one to work correctly
16   'depends': ['base'],
17
18   # always loaded
19   'data': [
20     # 'security/ir.model.access.csv',
21     'views/views.xml',
22     'views/templates.xml',
23     'views/agenda_view.xml',
24   ],
25   # only loaded in demonstration mode
26   'demo': [
27     'demo/demo.xml',
28   ],
29 }

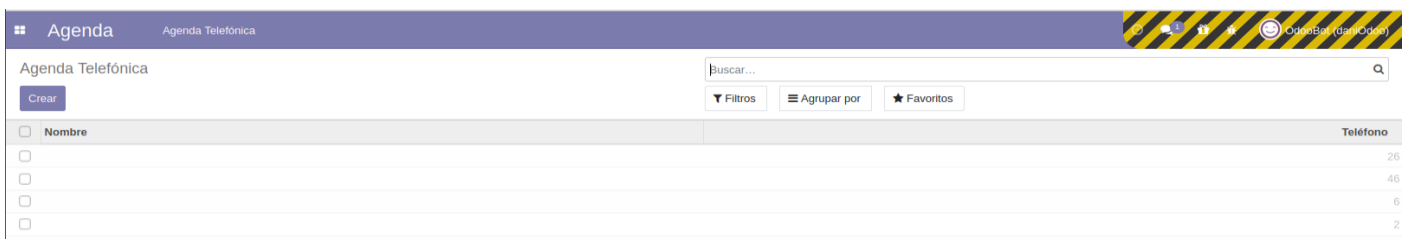
```



Pulsamos en actualizar el módulo y esperamos a que termine



Desplegamos el menú de la aplicación y ya podemos ver la etiqueta de “Agenda” pinchamos sobre ella y se abrirá la nueva vista



Arriba a la izquierda vemos la etiqueta del submenú que hemos creado y la vista de árbol que contiene los 2 campos: Nombre y Teléfono

Por último vamos a modificar la vista básica de formulario creada por Odoo. Añadimos las siguientes líneas al mismo fichero de vista:

```

18 <record model="ir.ui.view" id="agenda.list_view">
19   <field name="name">agenda.telefonica.view.tree</field>
20   <field name="model">agenda.agenda</field>
21   <field name="arch" type="xml">
22     <tree multi_edit="1" sample="1">
23       <field name="Nombre"/>
24       <field name="Telefono"/>
25     </tree>
26   </field>
27 </record>
28
29 <!-- VISTA FORMULARIO DE AGENDA -->
30 <record model="ir.ui.view" id="agenda.form_view">
31   <field name="name">agenda.telefonica.view.form</field>
32   <field name="model">agenda.agenda</field>
33   <field name="arch" type="xml">
34     <form string="Formulario de agenda telefonica">
35       <sheet>
36         <separator string="Contacto telefónico"/>
37         <group>
38           <group>
39             <field name="Nombre"/>
40           </group>
41           <group>
42             <field name="Telefono"/>
43           </group>
44         </group>
45       </sheet>
46     </form>
47   </field>
48 </record>
49
50 </data>
51 </odoo>

```

Creemos una vista del tipo formulario donde crearemos una plantilla (sheet) y dentro 2 grupos que se dividen al medio: a la izquierda quedará la etiqueta Nombre y a la derecha Teléfono

Volvemos a Odoo, actualizamos el módulo de agenda y nos vamos al submenú agenda telefónica, donde vamos a crear un nuevo contacto a través de la vista formulario que acabamos de crear:

Agenda Telefónica / Nuevo

Guardar Descartar

Contacto telefónico

Nombre: Telefono:

Rellenamos los 2 campos y guardamos para ver si funciona correctamente

Agenda	
Agenda Telefónica	
<input type="button" value="Crear"/>	<input type="text" value="Buscar..."/>
<input type="button" value="Filtros"/> <input type="button" value="Agrupar por"/> <input type="button" value="★ Favoritos"/>	
1-1 / 1	
Nombre	Teléfono
<input type="checkbox"/> Daniel	666.888.555

Comprobamos como se nos han guardado los datos

Ya solo nos queda comprobar que los datos se nos han guardado correctamente en la base de datos:

pgAdmin 4

localhost/browser/#

1 new message

Cómo instalar Odoo...

pgAdmin 4

Odoo

Building a Module ...

Zona de Usuarios

dpazlor@gmail.com (internal)

Navegador

account_reconcile_mio

account_reconcile_mo

account_reconcile_mo

account_resequence_v

account_setup_bank_n

account_tax

account_tax_filiation_n

account_tax_group

account_tax_repartitioi

account_tax_repartitioi

account_tax_repartitioi

account_tax_repartitioi

account_tax_report

account_tax_report_lin

account_tax_report_lin

account_tax_sale_adv

account_tax_sale_orde

account_tax_template

account_tax_template

account_tour_upload_t

account_tour_upload_t

account_unreconcile

agenda_agenda

Columns (7)

id

Nombre

Telefono

create_uid

create_date

write_uid

write_date

Disparadores

RLS Policies

Tablero

Propiedades

SQL

Estadísticas

Dependencias

Dependientes

public.agenda_agenda/daniOdoo/dani@dani-server

Sin limite

Query Editor

Historial de Consu...

1 SELECT * FROM public.agenda_agenda

2 ORDER BY id ASC

Salida de Datos

Explains

Mensajes

Notificaciones

id [PK] integer	Nombre character varying	Telefono integer	create_uid integer	create_date timestamp without time zone	write_uid integer	write_date timestamp without time zone
1	Daniel	666888555		2022-03-09 20:31:28.710364		2022-03-09 20:31:28.710364

Abrimos PgAdmin y visualizamos las columnas de la tabla donde podemos ver que los datos que hemos introducido en Odoo