Actividad 8. Creación de un módulo de Odoo.

Enunciado

Se precisa crear un componente o módulo que sea capaz de gestionar una **agenda telefónica** con las siguientes características:

- ➔ Objeto llamado *agenda* con, al menos, dos campos: *Nombre* y *Teléfono*.
- ➔ Menú en la aplicación que enlace al objeto.
- ➔ Vista formulario con los datos del objeto.
- ➔ Vista árbol con los datos del objeto.8

NOTAS Jose:

Cada vez que modificamos un fichero .py debemos reiniciar el servicio de odoo. Pero si modificamos un fichero XML solo necesitamos actualizar el modulo desde aplicaciones.

Tarea

Crea el módulo y entrega un informe con todas las consideraciones oportunas que se necesiten para entender cómo has realizado la tarea, así como fragmentos de código fuente.

La evaluación de la tarea se hará según el siguiente esquema:

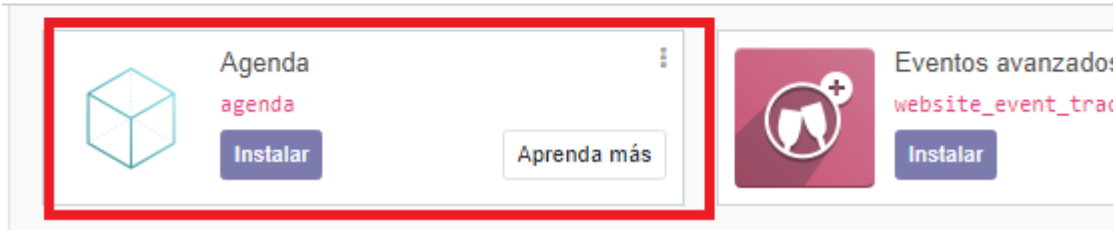
- ➢ Creación de la estructura de la carpeta del módulo *Agenda/*. (***10 ptos***)

Para esta tarea use Visual studio code para realizar una conexión ssh ya que Atom está fuera de ciclo de vida. Realizamos la conexión con el usuario odoo, nos posicionamos dentro de la carpeta odoo y lanzamos el comando siguiente:

```
odoo@jose-VirtualBox:~/odoo$ ./odoo-bin scaffold agenda ./addons/
```

Carpeta "addons": Contiene los módulos instalados en Odoo.
Carpeta "models": Contiene los archivos Python que definen los modelos de datos.
Carpeta "views": Contiene los archivos XML que definen las vistas de usuario.
Carpeta "static": Almacena archivos estáticos como imágenes, hojas de estilo y scripts.
Carpeta "security": Contiene archivos XML relacionados con la seguridad y los permisos del módulo.

Desde odoo ya podemos visualizar el módulo para realizar la instalación



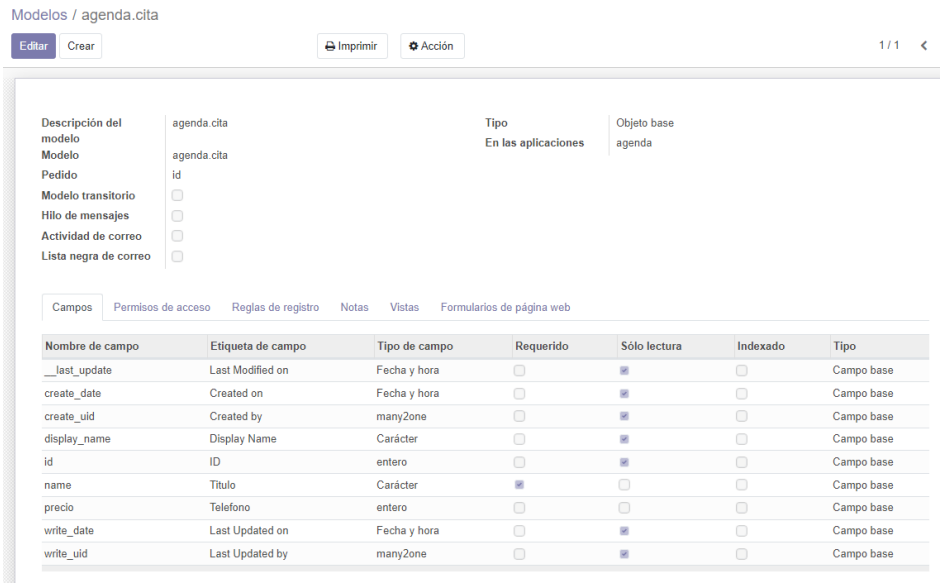
- Modelo correctamente definido *Agenda/models/models.py* y cargado correctamente en Odoo. (***10 ptos***).

```
from odoo import models, fields, api

class agenda_cita(models.Model):
    _name = 'agenda.cita'

    name = fields.Char(string="Nombre", required= True,help="Nombre de la cita")
    telefono = fields.Char(string='Teléfono', help="Número de teléfono de contacto")
```

Modelo visto desde odoo.



➢ Elemento de menú *Agenda* correctamente visualizado en la Odoo a través de su definición en el fichero *Agenda/views/views.xml* (***10 ptos***).

Fichero views.xml con el menú creado

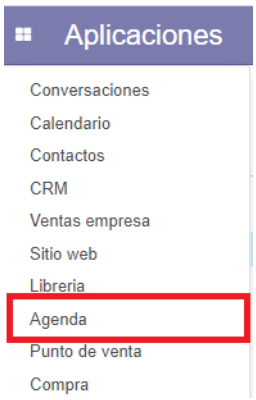
```
<!-- actions opening views on models -->

    <record model="ir.actions.act_window" id="agenda.cita_action_window">
      <field name="name">agenda.cita.action_window</field>
      <field name="res_model">agenda.cita</field>
      <field name="view_mode">tree,form</field>
    </record>


  <!-- Menu -->

    <menuitem name="Agenda" id="agenda.menu_root"/>
    <menuitem name="Citas" id="agenda.citas_menu" parent="agenda.menu_root" action="agenda.cita_action_window"/>
```

Menú desde odoo:



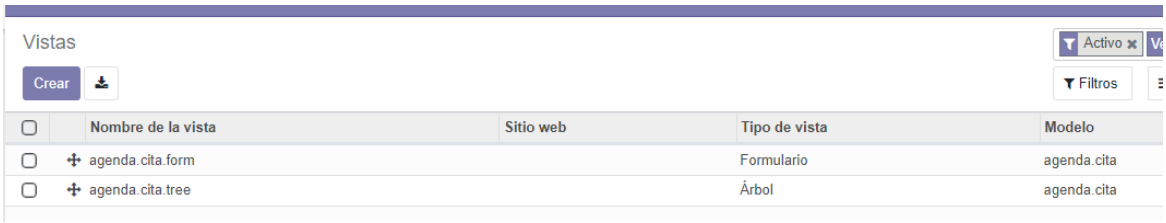
¡!OJO. Debemos dar permisos sobre el modelo para poder visualizar el menú:

➢ Vistas de formulario y de árbol definidas correctamente en el fichero *Agenda/views/views.xml* (***10 ptos***).

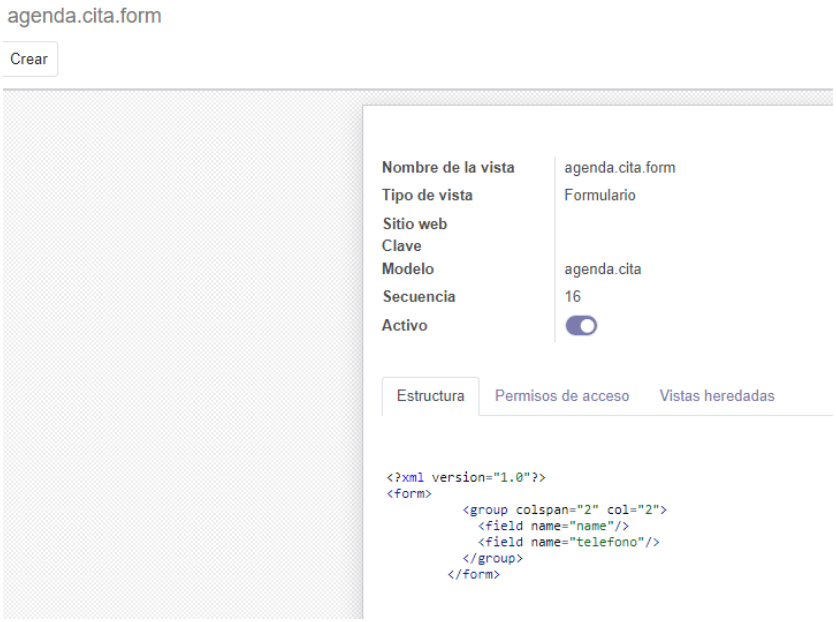
Ficheros views.xml

```xml
<odoo>
  <data>
<!-- Vista Arbol -->
 <record model="ir.ui.view" id="agenda.cita_tree">
  <field name="name">agenda.cita.tree</field>
      <field name="model">agenda.cita</field>
      <field name="arch" type="xml">
        <tree>
          <field name="name"/>
          <field name="telefono"/>
        </tree>
      </field>
  </record>
<!-- Vista Formulario -->
  <record model="ir.ui.view" id="agenda.cita_form">
  <field name="name">agenda.cita.form</field>
      <field name="model">agenda.cita</field>
      <field name="arch" type="xml">
        <form>
          <group colspan="2" col="2">
            <field name="name"/>
            <field name="telefono"/>
          </group>
        </form>
      </field>
  </record>
```

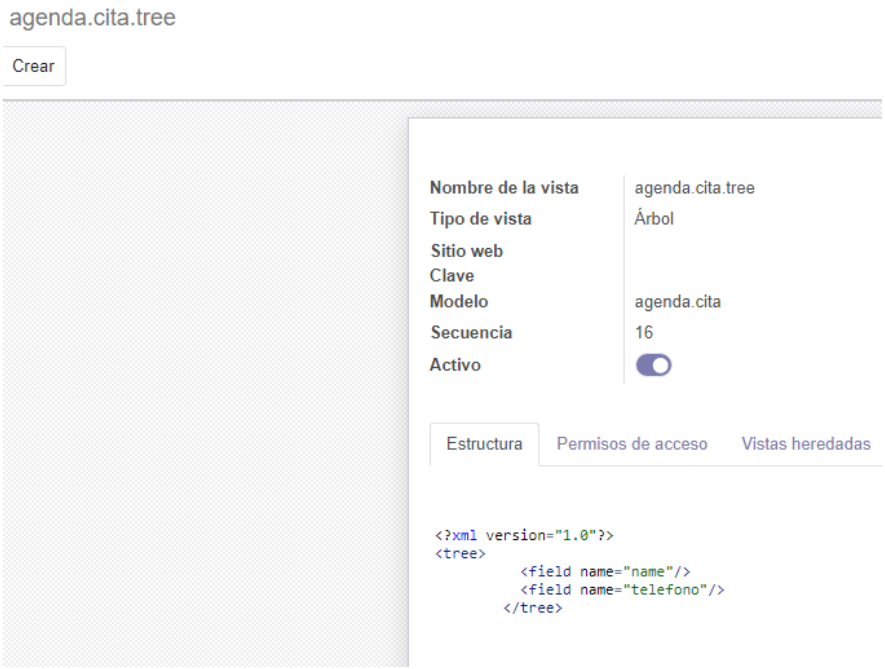
Vistas correctamente creadas una vez actualizamos el módulo:



Formulario:



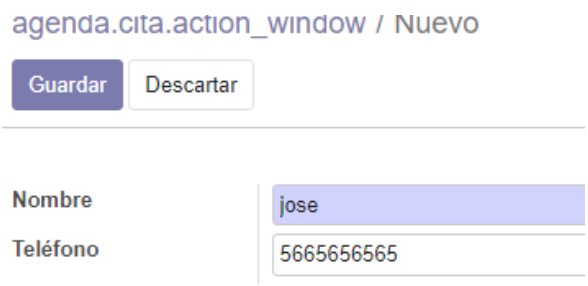
Árbol:



➢ Demostración del funcionamiento adecuado del módulo *agenda* (***20 ptos***).

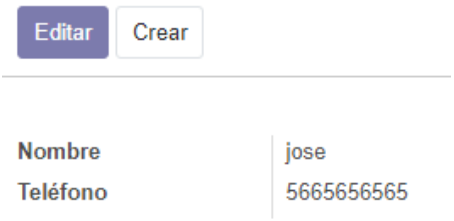
Entramos en el menú y pulsamos crear

Rellenamos los campos necesarios:

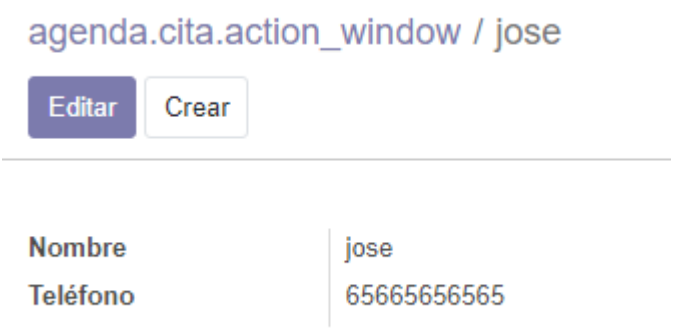


Y ya tenemos el registro creado:

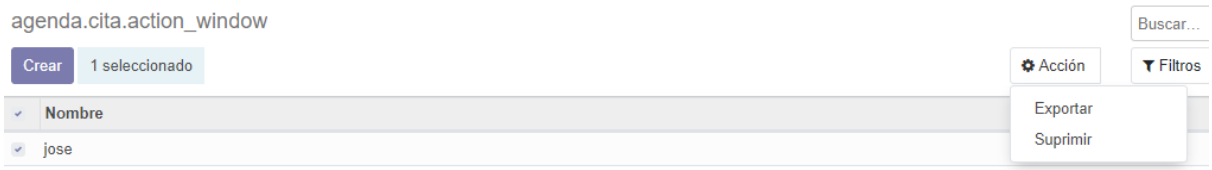
Podemos editar



Registro editado:



Podemos borrar los registro:



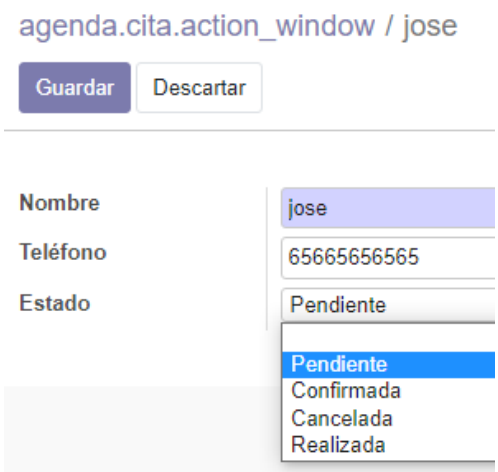
➢ Implementación de alguna funcionalidad, modelo o vista adicional a lo pedido en el enunciado (***20 ptos***).

1.Primera funcionalidad extra. Estados para las citas:

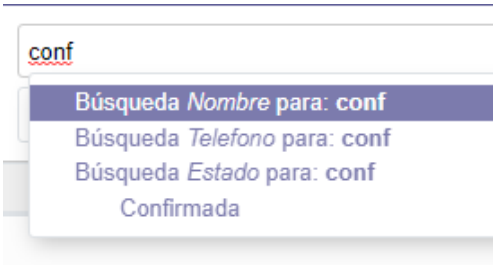
```
class agenda_cita(models.Model):
    _name = 'agenda.cita'

    name = fields.Char(string="Nombre", required= True,he
    telefono = fields.Char(string='Teléfono', help="Númer
    state = fields.Selection([
        ('pendiente', 'Pendiente'),
        ('confirmada', 'Confirmada'),
        ('cancelada', 'Cancelada'),
        ('realizada', 'Realizada')],
        string='Estado', default='pendiente')
```

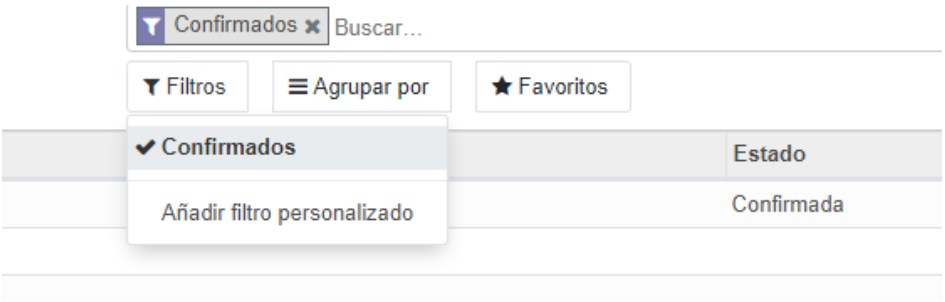
```
<field name="telefono"/>
<field name="state"/>
```



Segunda funcionalidad. Añadimos la posibilidad de filtrar por teléfono y estado.



Y un filtro que solo muestra las citas con estado confirmado



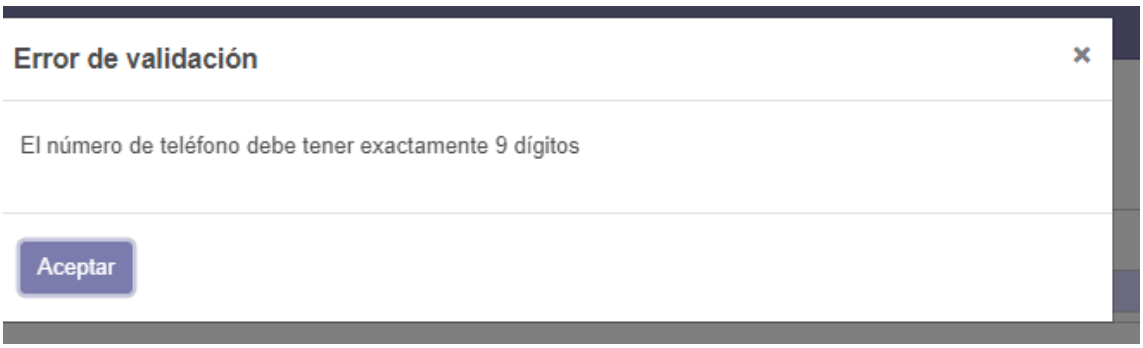
Para ello añadimos una vista búsqueda y añadimos los campos que queremos filtrar. Aparte el filter para configurar un filtro fijo según los datos que indicamos.

```
<!-- Vista busqueda -->
  <record model="ir.ui.view" id="agenda.cita_search_view">
  <field name="name">agenda.cita.search</field>
      <field name="model">agenda.cita</field>
      <field name="arch" type="xml">
              <search>
                <field name="name" string="Nombre"/>
                <field name="telefono" string="Telefono"/>
                <field name="state" string="Estado"/>
                <filter name="Confirmados" domain="[('state','=','confirmada')]"/>
              </search>
      </field>
  </record>
```

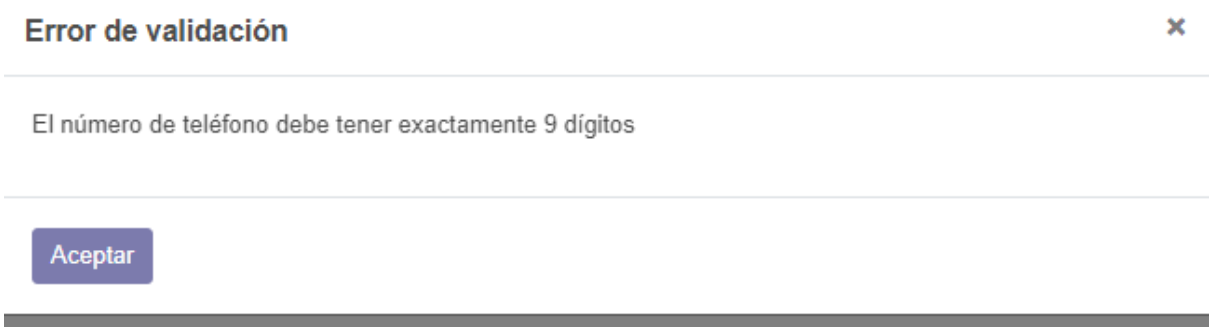
Tercera funcionalidad. Datos introducidos:

Validar teléfono y nombre:

- Validamos el número de teléfono tiene un tamaño correcto



- Validamos que nombre no contenga números



En el modelo generamos una función que valida teléfono y name. Odoo captura los raise y los muestra.

```
@api.constrains('telefono', 'name')
def _check_telefono(self):
    for record in self:
        if record.telefono:  # Solo verificar si el número de teléfono está presente
            if len(record.telefono) != 9 or not record.telefono.isdigit():
                raise ValidationError("El número de teléfono debe tener exactamente 9

            if record.name and bool(re.search(r'\d', record.name)):
                raise ValidationError("El nombre no puede contener números")
```

- **Cuarta funcionalidad:**

  Campo motivo de cancelación. Se activa cuando el estado de la cita es cancelado.

  Modelo

```
cancel_reason = fields.Text(string='Motivo de cancelación')  # Nuevo campo
```

  View:

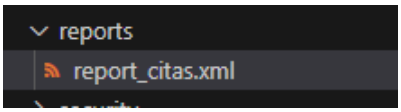
  Aquí indicamos que se muestre cuando estado es igual a cancelada.

```
<field name="state"/>
<field name="cancel_reason" attrs="{'invisible': [('state', '!=', 'cancelada')]}"/>
```

➢ Limpieza y claridad del informe (***20 ptos***).

Para el informe realice los siguientes cambios:

Primero se creó la carpeta reports y dentro los report que vamos usar:



Este report consta de los datos por decir así de configuración.

```xml
<odoo>
    <data>
        <report
            id="agenda.report_cita"
            model="agenda.cita"
            string="Imprimir Reporte de Citas"
            report_type="qweb-pdf"
            file="agenda.report_agenda"
            name="agenda.report_agenda"
        />
```

Y seguido la plantilla. Aquí indicamos los campos que queremos mostrar.

```xml
<template id="agenda.report_agenda">
    <t t-call="web.external_layout">
        <t t-foreach="docs" t-as="cita">
            <main>
                <div class="page">
                    <h2>Reporte de Citas</h2>
                    <table class="table table-condensed">
                        <thead>
                            <tr>
                                <th>Nombre</th>
                                <th>Teléfono</th>
                                <th>Estado</th>
                            </tr>
                        </thead>
                        <tbody>
                            <tr>
                                <td>
                                    <span t-field="cita.name"/>
                                </td>
                                <td>
                                    <span t-field="cita.telefono"/>
                                </td>
                                <td>
                                    <span t-field="cita.state"/>
                                </td>
                            </tr>
                        </tbody>
```

Tenemos que dar de alta en el manifest la nueva plantilla. En depends no es necesario añadir report ya que desde odoo 14 está incluido en la base.

```
    'depends': ['base'],

    # always loaded
    'data': [
        # 'security/ir.model.access.csv',
        'views/views.xml',
        'views/templates.xml',
        'reports/report_citas.xml'
    ],
```

Imprimir reports.

Report generado.

Your logo

My Company (San Francisco)
VIGO
BALAIDOS
36210 vIGO (Pontevedra)
España

Reporte de Citas

Nombre	Teléfono	Estado
Jose Alonso	321456789	Pendiente

Reporte de Citas

Nombre	Teléfono	Estado
Paco a	662134567	Pendiente

Reporte de Citas

Nombre	Teléfono	Estado
Luis	987653421	Pendiente

Reporte de Citas

Nombre	Teléfono	Estado
Marco Polo	113568565	Pendiente