



**Apellidos, Nome: Iglesias Nieto, Rodrigo**



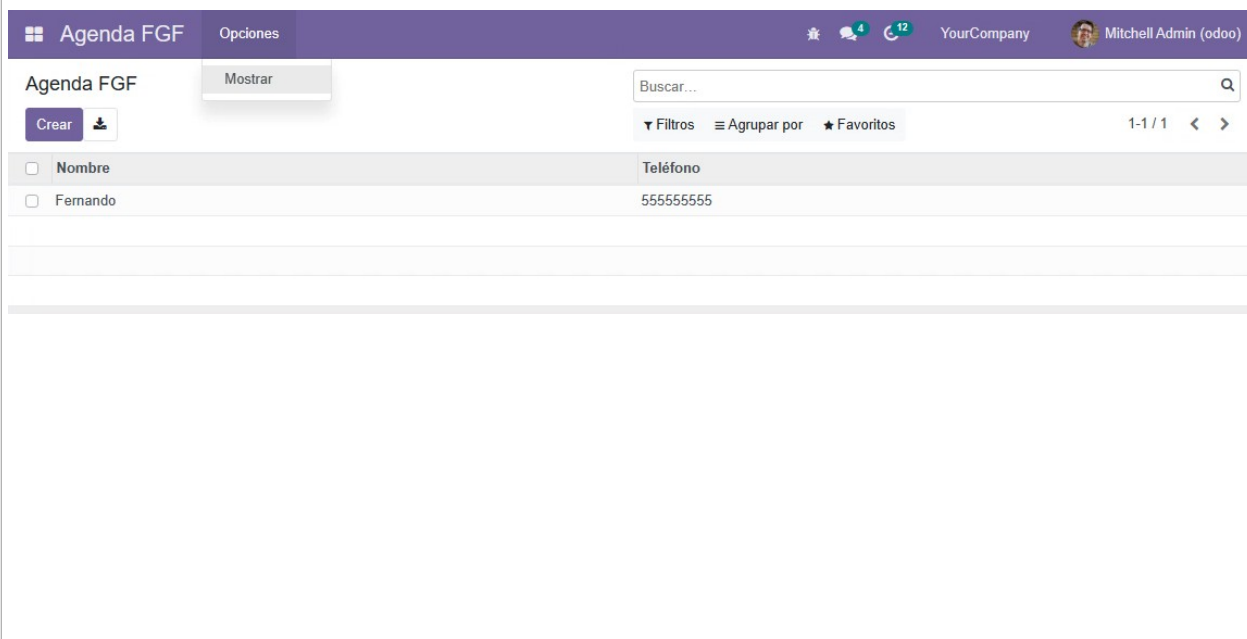
1. Proxecto de desenvolvemento de compoñentes de Odoo en implantación cliente/servidor. Incluír documentación e capturas de pantalla que sexan necesarias.

Crea un compoñente ou módulo que xestione unha axenda telefónica coas seguintes características:

- Modelo e controlador:
  - Obxecto na aplicación cuxa nomenclatura siga o patrón axendaIniciaisNomeCompleto (por exemplo, axendaFGF)
  - A axenda debe proporcionar polo menos dous campos: name e telephone.
  - Táboa na base de datos cos datos do obxecto.
- Vista:
  - Menú na aplicación que ligazón ao obxecto. O nome do mesmo debe seguir o patrón AxendaIniciais NomeCompleto (por exemplo, Axenda FGF).
  - Vista árbore cos datos do obxecto.
  - Vista formulario cos datos do obxecto.

Módulo de exemplo:

1. Vista en forma de árbore:



## 2. Vista de entrada de datos:

Agenda FGF Opciones

Agenda FGF / Nuevo

Guardar Descartar

Nombre

Teléfono

## 3. Descripción do módulo:

Aplicaciones / Agenda FGF

Guardar Descartar 2 / 3 < >

### Agenda FGF

Por **Fernando Gómez Folgar**

Actualizar Desinstalar

Información Datos técnicos Características instaladas

Sitio web	http://www.ieslosada.com	Nombre técnico	agenda_fgf
Categoría	Sin categoría	Licencia	LGPL versión 3
Resumen	Módulo de gestión de Agenda	Última versión	15.0.0.1

Agenda FGF permite gestionar los nombres y los números de teléfono de los contactos

Para implementar o módulo hai que ter en conta as instrucións seguintes:

(CA5.1 Recoñecéronse as sentenzas dá linguaxe propia do sistema ERP-CRM. 15% ME)

1. Crear o persoal do módulo mediante a opción scaffold de Odoo.

(CA5.2 Utilizáronse vos elementos de programación dá linguaxe para crear compoñentes de manipulación de datos. 20% ME)

2. Implementar o módulo empregando os elementos da linguaxe de programación (python, xml, csv).

(CA5.3 Modificáronse compoñentes de software para lle engadir funcionalidade nova ao sistema. 20% ME)

3. Modificar o compoñente creado definir a icona do módulo e a súa información personalizada en relación ao autor do módulo, sitio web, resumo curto e información ampliada indicativa do propósito do mesmo .

(CA5.4 Integráronse vos novos compoñentes de software non sistema ERP-CRM. 20% ME)

4. Instalar o módulo nunha instalación cliente-servidor de Odoo.

(CA5.5 Verifícase ou correcto funcionamento dous compoñentes creados. 15%)

5. Deseñar, documentar e realizar probas para verificar o correcto funcionamento do compoñente.

(CA5.6 Documentáronse todos vos compoñentes creados ou modificados 10%)

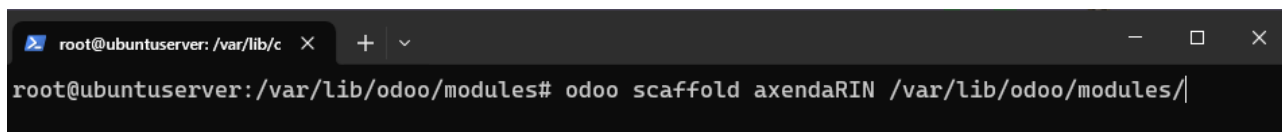
6. Documentar o código fonte que se implementou tanto para a vista como o modelo e o controlador.

Forma de entrega:

- Entrega o proxecto comprimido en formato zip empregando a nomenclatura UD5\_Apelido1\_A pelido2\_Nome.zip coa seguinte estrutura interna:
  - Módulo distribuíble nun arquivo comprimido coa nomenclatura UD5\_Apelido1\_A pelido2\_Nome.tar.gz
  - Documentación do proxecto en formato pdf coa nomenclatura UD5\_Apelido1\_A pelido2\_Nome.pdf

(Completar...)

1)



```
root@ubuntuuser: /var/lib/c  X  +  v
root@ubuntuuser:/var/lib/odoo/modules# odoo scaffold axendaRIN /var/lib/odoo/modules/
```

3)

Para añadir una imagen personalizada de icono al módulo de Odoo necesitamos tener una imagen en formato .png en una ruta específica. Esta ruta es **<nombre-módulo>/static/description/icon.png**.

Para cambiar la información específica del módulo debemos modificar el archivo **\_\_manifest\_\_.py**. Este es el archivo del módulo ya modificado:

```

GNU nano 6.2                                __manifest__.py
# -*- coding: utf-8 -*-
{
    'name': "axendaRIN",

    'summary': """
        Módulo de gestión de Agenda RIN""",

    'description': """
        Agenda RIN permite gestionar los nombres y los números de teléfono de los contactos
        """,

    'author': "Rodrigo Iglesias Nieto",
    'website': "https://www.kike.com",

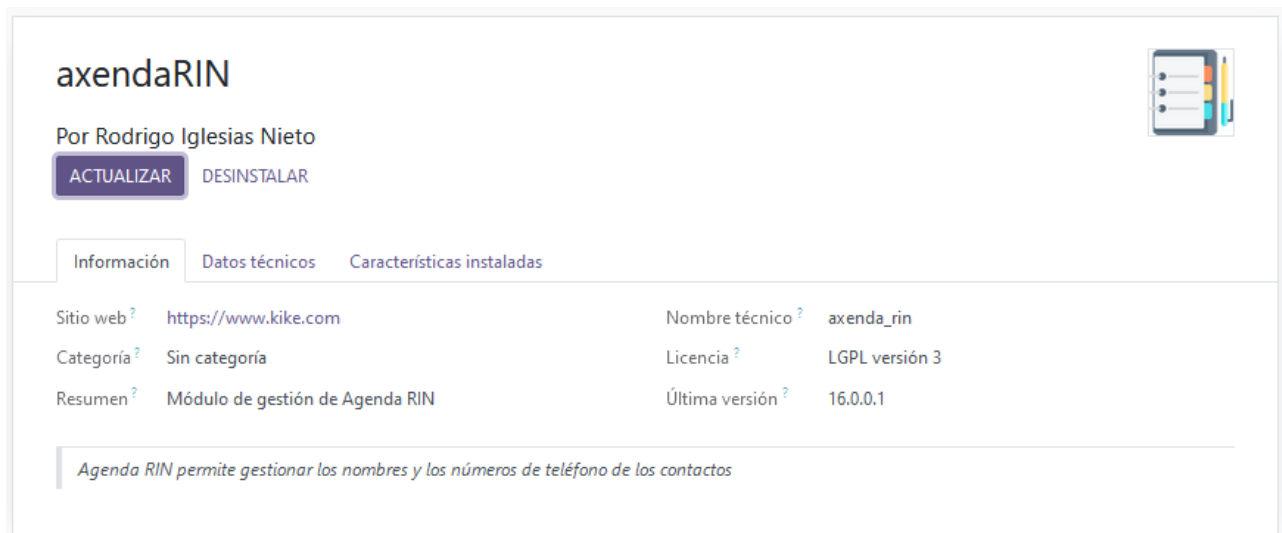
    # Categories can be used to filter modules in modules listing
    # Check https://github.com/odoo/odoo/blob/16.0/odoo/addons/base/data/ir_module_category_data.xml
    # for the full list
    'category': 'Uncategorized',
    'version': '0.1',

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

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

```

Que se vería de la siguiente manera el la interfaz de Odoo:

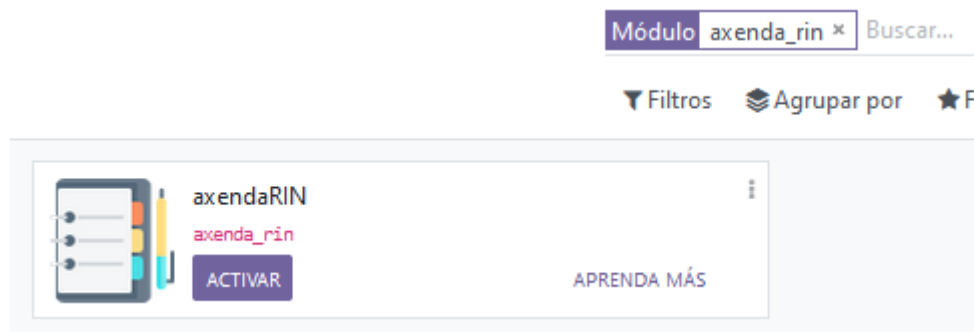


En este módulo la imagen tiene un tamaño de 64x64 pixeles

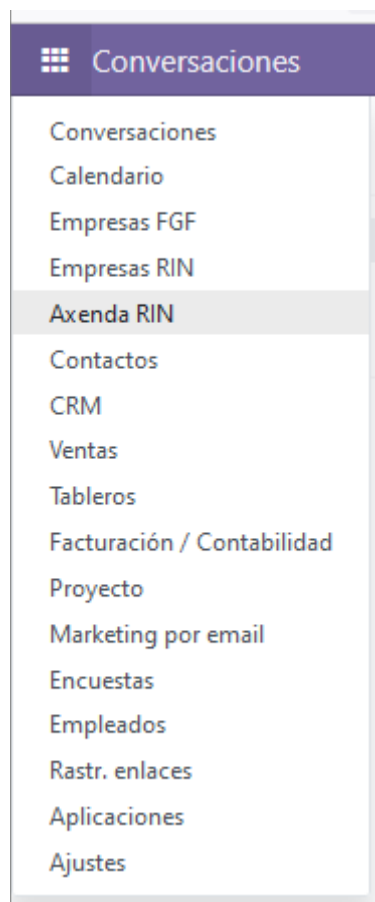
4)

Para poder instalar nuestro módulo en Odoo debemos tener la carpeta en la ruta especificada en la configuración dónde se encuentran los módulos. En este caso **/var/lib/odoo/modules/**. Ahora debemos reiniciar el servicio de Odoo en el servidor con el comando **systemctl restart odoo**, una vez hecho el reinicio debemos ir al **menú Aplicaciones>Actualizar la lista de aplicaciones** para que cargue nuestro módulo.

Ahora podemos buscar por el nombre del módulo:



Para instalarlo debemos darle al botón **Activar** y ahora en el menú podremos ver nuestra aplicación:

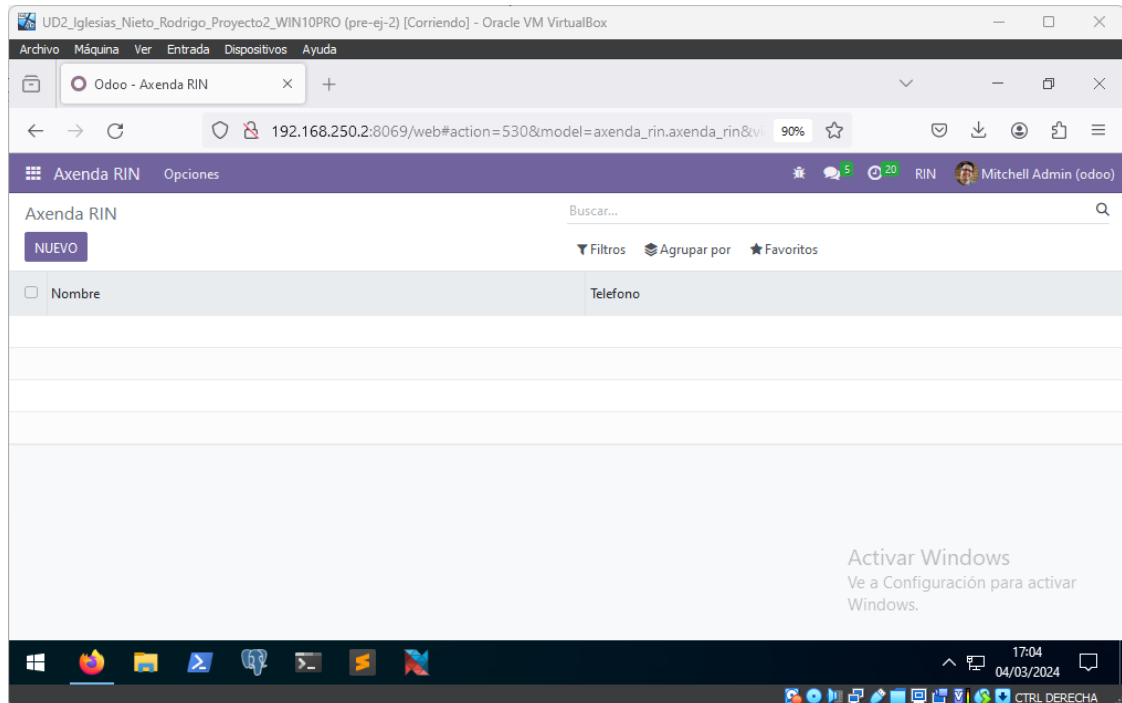


5)

Para probar que funciona correctamente el módulo vamos a realizar unas pruebas.

### Prueba 1 - Añadir contacto:

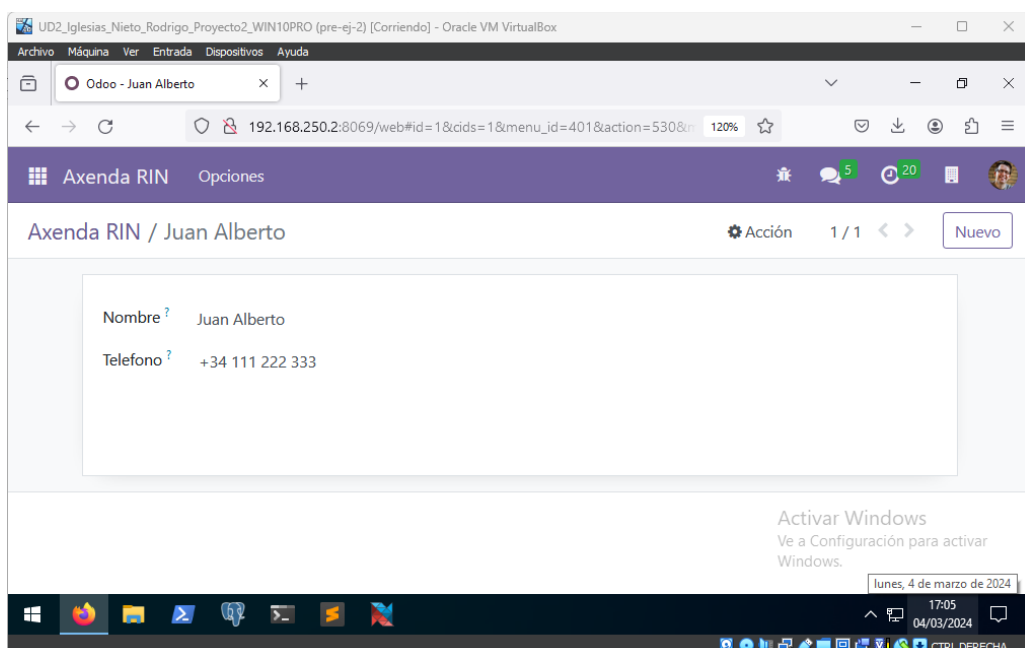
Para añadir un contacto iremos a Axenda RIN y pulsaremos el botón *Nuevo*



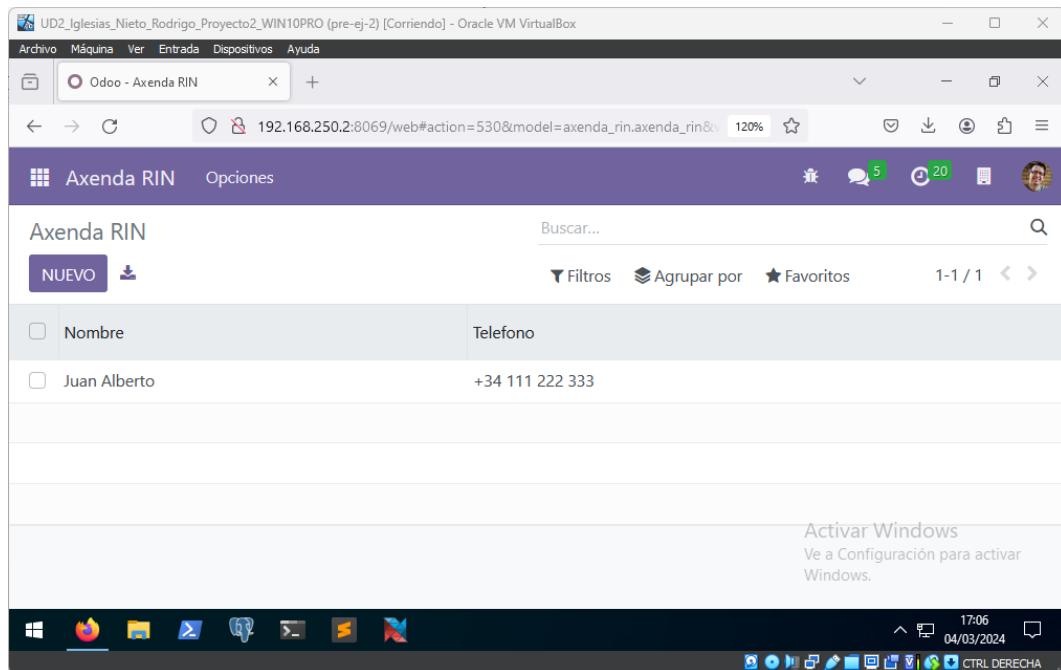
Para esta prueba insertaremos los siguientes datos:

- **Nombre:** Juan Alberto
- **Telefono:** +34 111 222 333

Una vez introducidos los datos le daremos a *Nuevo* para guardarlo. Una vez le damos al botón podemos ir para atrás para ver si se ha introducido correctamente:

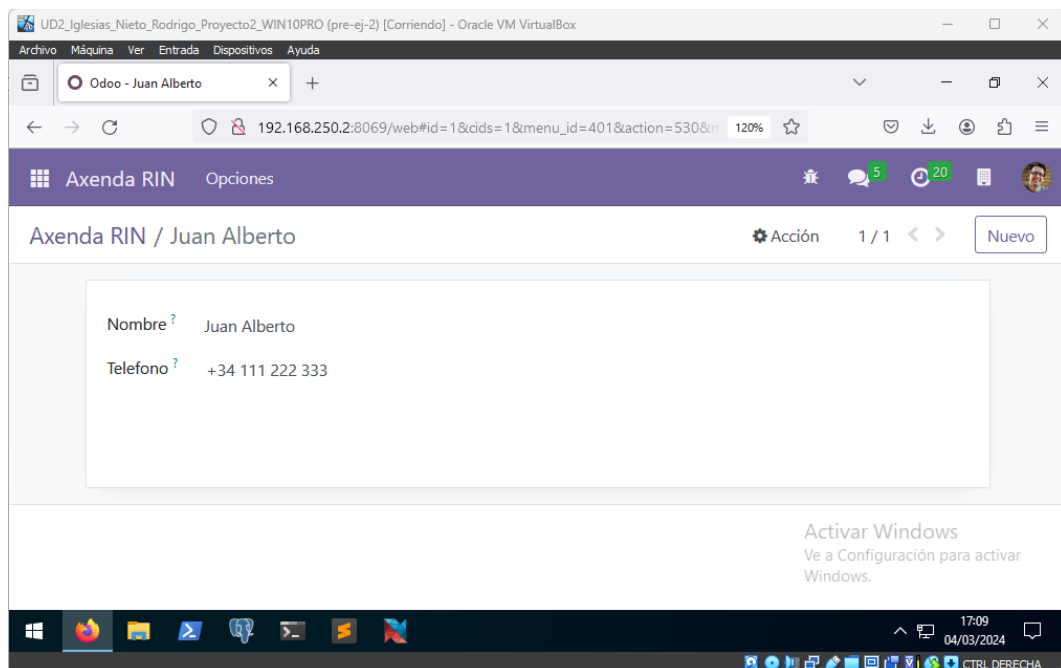


Si se ha introducido correctamente veremos lo siguiente en la vista:



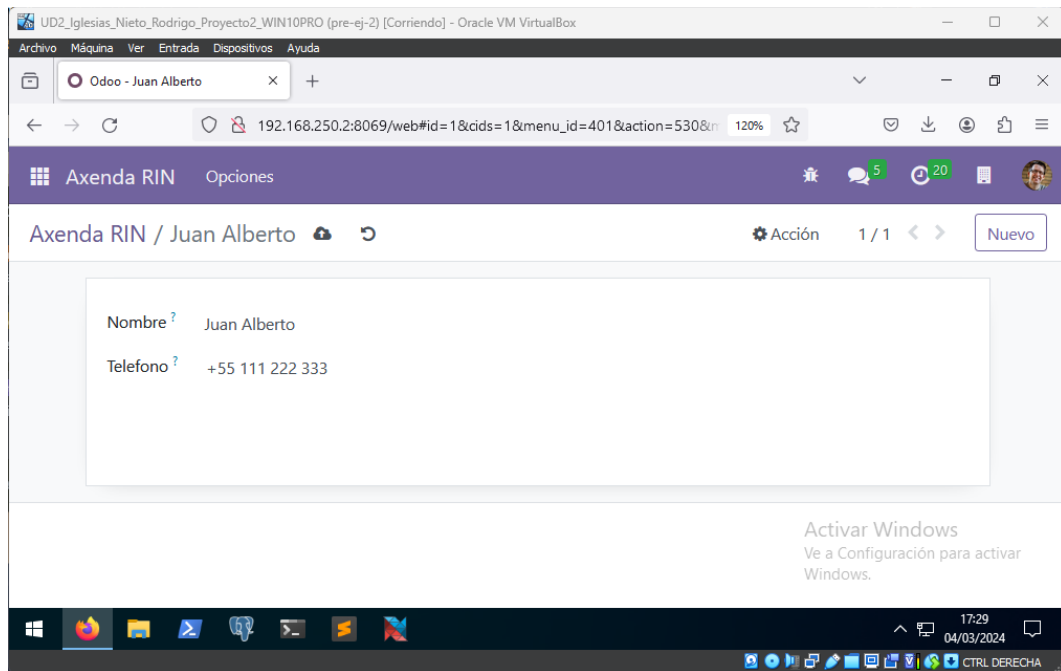
### Prueba 2 – Edición de contacto:

Para editar un contacto le daremos a nuestro contacto desde la vista y veremos el formulario con los datos del contacto:

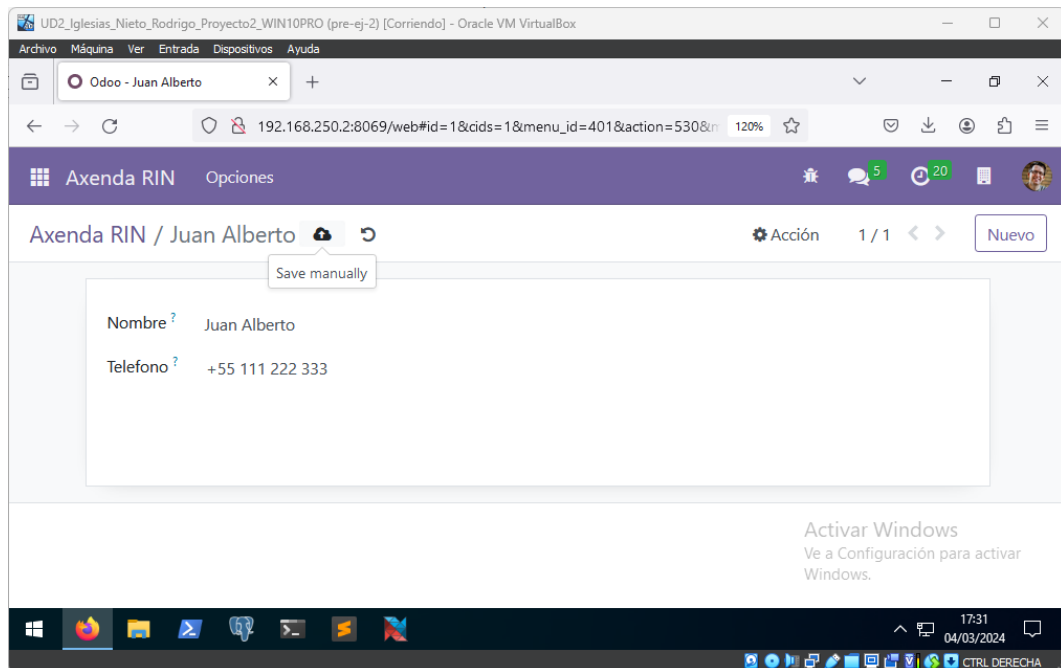


Para esta prueba de editar vamos a cambiar los datos a:

- **Nombre:** Juan Manuel
- **Telefono:** +55 111 222 333

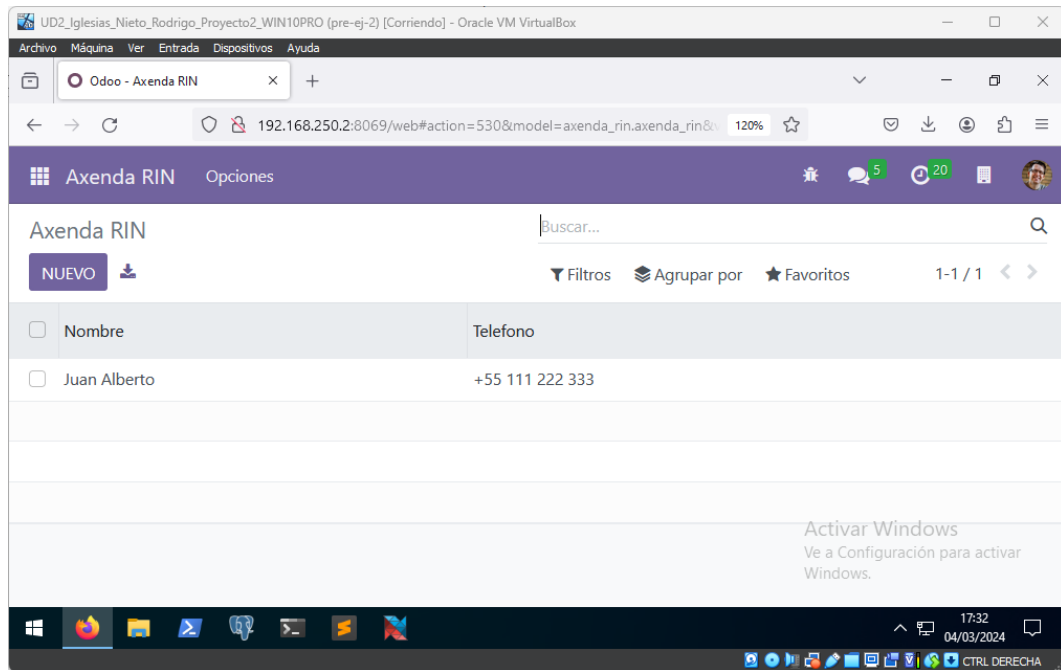


Para guardar los datos le daremos al botón de *Save manually*:



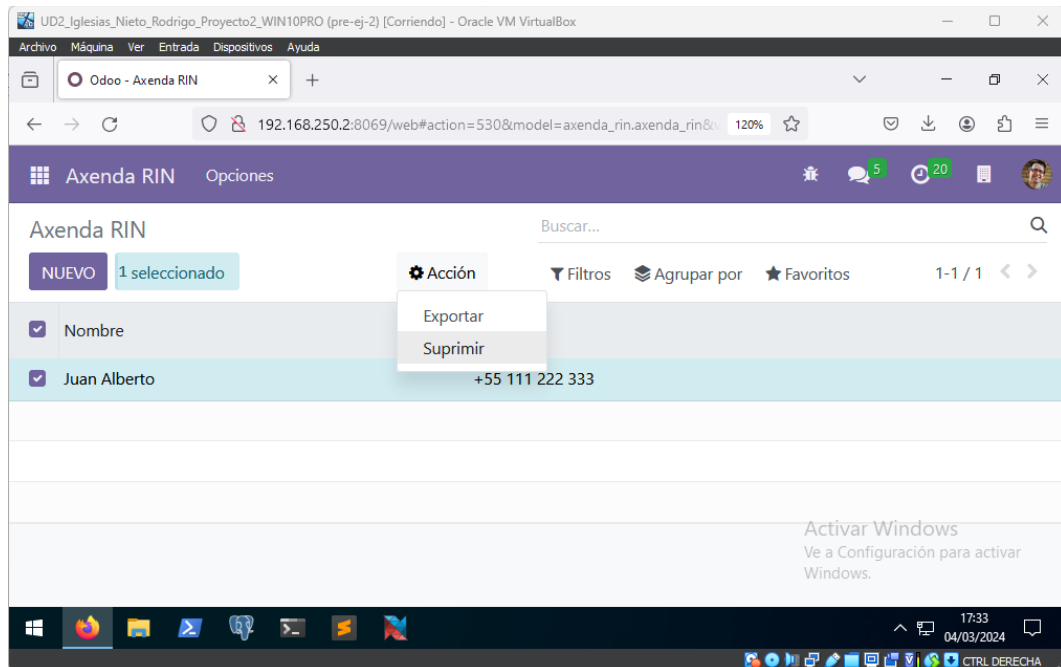


Ahora en la vista principal debemos ver los datos actualizados:

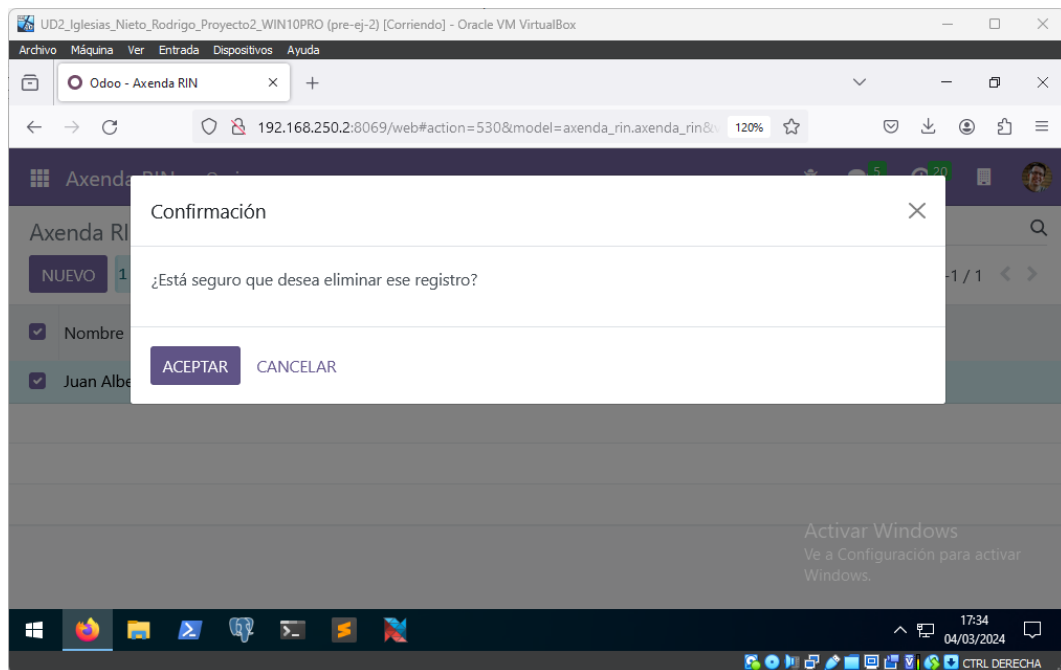


### Prueba 3 – Eliminar contacto:

Para eliminar un contacto debemos ir a la vista y seleccionar en el checkbox izquierdo de la fila el contacto deseado. Después debemos ir al menú *Acción* y darle al botón *Suprimir*



Borrar un contacto nos pedirá confirmación, para borrarlo definitivamente debemos darle a *Aceptar* o *Cancelar* si no queremos borrar el contacto seleccionado.



6)

La lista de archivos que hemos modificado después de la creación del módulo con el comando scaffold de Odoo son los siguientes:

- models/models.py
- views/views.xml
- \_\_manifest\_\_.py

Primero comentaremos el model que se encuentra en el archivo **models/models.py** es un archivo Python que tiene los datos que queremos almacenar:

```
models.py 1 X
models > models.py > axenda_rin
1  # -*- coding: utf-8 -*-
2
3  from odoo import models, fields, api
4
5
6  class axenda_rin(models.Model):
7      _name = 'axenda_rin.axenda_rin'
8      _description = 'axenda_rin.axenda_rin'
9
10     name = fields.Char('Nombre', required=True)
11     telephone = fields.Char('Telefono', required=True)
```

Los imports y la clase ya vendrán creados, incluyendo las variables `_name` y `_description`. Nosotros creamos las variables que se llaman **name** y **telephone**. A estas variables se le

asigna con el método `fields.Char()` el nombre que tendrán en el model y si es requerido. Char en este caso vendría a ser una String de java. Telephone también se guardará como Char ya que podremos añadir al principio el prefijo del país del cual es el teléfono.

El siguiente archivo es la vista, la cual está en el fichero `views/views.xml`. Este es el código xml que tiene el fichero:

```
views.xml X
views > views.xml
1  <odoo>
2  <data>
3      <!-- explicit list view definition -->
4      <record model="ir.ui.view" id="axenda_rin.list">
5          <field name="name">axendaRIN list</field>
6          <field name="model">axenda_rin.axenda_rin</field>
7          <field name="arch" type="xml">
8              <tree>
9                  <field name="name"/>
10                 <field name="telephone"/>
11             </tree>
12         </field>
13     </record>
14
15     <!-- actions opening views on models -->
16     <record model="ir.actions.act_window" id="axenda_rin.action_window">
17         <field name="name">Axenda RIN</field>
18         <field name="res_model">axenda_rin.axenda_rin</field>
19         <field name="view_mode">tree,form</field>
20         <field name="view_id" ref="axenda_rin.list"/>
21     </record>
22
23     <!-- Top menu item -->
24     <menuitem name="Axenda RIN" id="axenda_rin.menu_root"/>
25     <!-- menu categories -->
26     <menuitem name="Opciones" id="axenda_rin.menu_1" parent="axenda_rin.menu_root"/>
27     <!-- actions -->
28     <menuitem name="Mostrar" id="axenda_rin.menu_1_list" parent="axenda_rin.menu_1"
29         action="axenda_rin.action_window"/>
30 </data>
31 </odoo>
32
```

En la etiqueta `tree` de la línea 8 añadimos las columnas del modelo que queremos ver en la vista, en nuestro caso **name** y **telephone**. Después, en la línea 17 pondremos el nombre del módulo. En nuestro caso Axenda RIN. También debemos añadir la línea número 20.

En la última parte del fichero creamos los menús que tendrá el módulo. El primero es el principal, que tiene como nombre Axenda RIN y será `axenda_rin.menu_root`. Después crearemos otros dos `menuitem` que ya se verán dentro del otro menú Axenda RIN. Con `parent` nos referimos al padre del menú, para el menú Opciones el padre será `root` = Axenda RIN y para el menú Mostrar el padre será Opciones.

Por último el fichero `__manifest__.py` que contiene la información de nuestro módulo.

```
__manifest__.py
1  # -*- coding: utf-8 -*-
2  {
3      'name': "agendaRIN",
4
5      'summary': ""
6      |   Módulo de gestión de Agenda RIN"",
7
8      'description': ""
9      |   Agenda RIN permite gestionar los nombres y los números de teléfono de los contactos
10     |   "",
11
12     'author': "Rodrigo Iglesias Nieto",
13     'website': "https://www.kike.com",
14
15     # Categories can be used to filter modules in modules listing
16     # Check https://github.com/odoo/odoo/blob/16.0/odoo/addons/base/data/ir_module_category_data.xml
17     # for the full list
18     'category': 'Uncategorized',
19     'version': '0.1',
20
21     # any module necessary for this one to work correctly
22     'depends': ['base'],
23
24     # always loaded
25     'data': [
26         'security/ir.model.access.csv',
27         'views/views.xml',
28         'views/templates.xml',
29     ],
30     # only loaded in demonstration mode
31     'demo': [
32         'demo/demo.xml',
33     ],
34 }
35
```

Modificaremos el `name` que es el nombre del módulo, `summary` que será un resumen, `description` que es una descripción más extensa de lo que hará el módulo, `author` el autor y `website` la página web.

También debemos descomentar una cosa que vendrá comentada que es la línea 26, debemos descomentar `'security/ir.model.access.csv'`. Para que haga comprobaciones de la seguridad.