

Sistemas de Gestión Empresarial

UD05 - Desarrollo de componentes

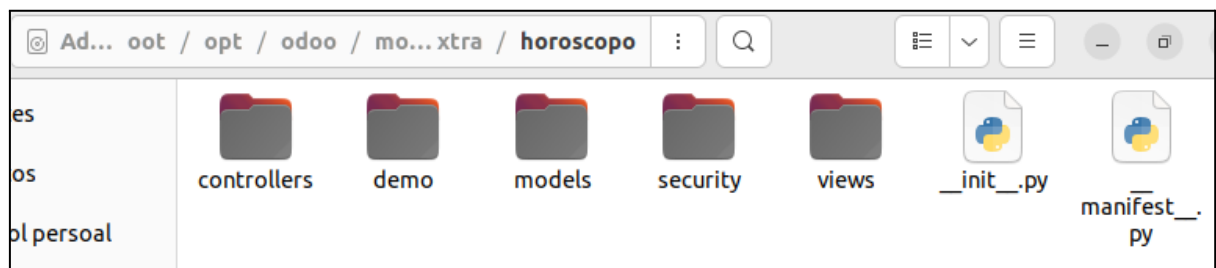
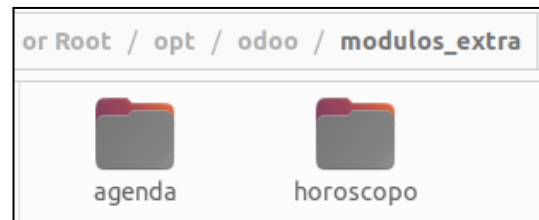
0. Preparación del escenario	2
1. Modificación de __manifest__.py	3
2. Modificación de models.py	4
3. Modificación de views.xml	5
4. Código para mostrar los datos	6
4.1. Cálculo de la edad	6
4.2. Cálculo del signo del zodiaco	6

0. Preparación del escenario

Del mismo modo que en su momento creamos el módulo `agenda`, ahora vamos a crear horoscopo. Para ello, desde la terminal ejecutamos la siguiente línea:

```
operador_odoo@odoo-VirtualBox:~$ /opt/odoo/odoo-bin scaffold horoscopo /opt/odoo/modulos_extra
```

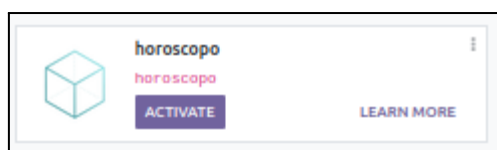
Una vez ejecutado, podemos comprobar que se ha creado en el destino indicado y que dentro tenemos el formato esperado:



Tras confirmar que tenemos todos los ficheros deseados, procedemos a reiniciar el servicio.

```
odoo@odoo-VirtualBox:~$ sudo service odoo16 restart
[sudo] Contraseña de odoo:
odoo@odoo-VirtualBox:~$ sudo service odoo16 status
● odoo16.service - Odoo como servicio
   Loaded: loaded (/etc/systemd/system/odoo16.service; enabled; vendor preset: enabled)
   Active: active (running) since Fri 2023-05-12 19:16:40 CEST; 26s ago
     Main PID: 4808 (python3)
        Tasks: 6 (limit: 9457)
      Memory: 101.4M
         CPU: 3.695s
       CGroup: /system.slice/odoo16.service
              └─4808 python3 /opt/odoo/odoo-bin

Mai 12 19:16:40 odoo-VirtualBox systemd[1]: Started Odoo como servicio.
odoo@odoo-VirtualBox:~$
```



Tras actualizar la lista de aplicaciones, el módulo creado figura disponible para instalar.

Crear un nuevo addon con un modelo "horoscopo" que herede de res.partner.

Añadir los campos nuevos

Añadir la parte de programación en python para calcular edad y signo cuando el usuario modifique la fecha de nacimiento.

Modificar la vista formulario para que el usuario pueda introducir esta información en una nueva page dentro del notebook.

Modificar la vista lista para que incorpore el signo del zodiaco en la segunda columna

Modificar la vista kanban para que incorpore el signo del zodiaco justo debajo del nombre.

Comprobar en la base de datos que la información se está guardando correctamente

1. Modificación de __manifest__.py

Una vez editado, el archivo figura así:

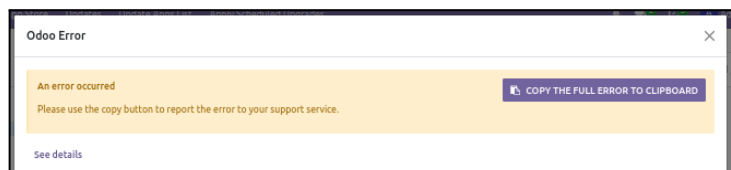
```
1 # -*- coding: utf-8 -*-
2 {
3     'name': "horoscopo",
4
5     'summary': ""
6     Tarea de la UD55
7     "",
8
9     'description': ""
10     Este módulo añade funcionalidad a la aplicación de contactos con 3 nuevos campos:
11     - Fecha de nacimiento
12     - Edad (calculado automáticamente)
13     - Signo del zodiaco (calculado automáticamente)
14     "",
15
16     'author': "Iván Estévez",
17     'website': "https://github.com/Mskina",
18
19     # Categories can be used to filter modules in modules listing
20     # Check https://github.com/odoo/odoo/blob/16.0/odoo/addons/base/data/ir\_module\_category\_data.xml
21     # for the full list
22     'category': 'Uncategorized',
23     'version': '0.1',
24
25     # any module necessary for this one to work correctly
26     'depends': ['base', 'contacts'],
27
28     # always loaded
29     'data': [
30         # 'security/ir.model.access.csv',
31         'views/views.xml',
32         'views/templates.xml',
33     ],
34     # only loaded in demonstration mode
35     'demo': [
36         'demo/demo.xml',
37     ],
38 }
```

2. Modificación de models.py

Nuestro primer paso es escribir el código del módulo

```
1 # -*- coding: utf-8 -*-
2
3 from odoo import models, fields, api
4
5 class horoscopo(models.Model):
6     _inherit = "res.partner"
7
8     fechaNacimiento = fields.Date("Fecha de nacimiento")
9     edad = fields.Integer(
10         string="Edad", readonly=True, compute="_calcula_edad", store=True
11     )
12
13     signo = fields.Char(
14         string="Signo zodiaco", readonly=True, compute="_calcula_signo", store=True
15     )
16
17     @api.depends("fechaNacimiento")
18     def _calcula_edad(self):
19         for registro in self:
20             edad = 99
21             registro.edad = edad
22
23     @api.depends("fechaNacimiento")
24     def _calcula_signo(self):
25         self.ensure_one() # si pasamos de esta línea es que solo ha llegado un registro
26         signo_zodiaco = "Mi signo"
27         self.signo = signo_zodiaco
```

Tras guardarlo, intentamos instalar el módulo. Nos devuelve un error porque, cuando en la línea 25 de `models.py` escribimos `self.ensure_one()`, estamos indicando que únicamente se ejecute cuando le llegue un registro. Sin embargo, le están llegando todos los contactos creados.



Para solucionarlo, modificamos el código de forma que capturemos las excepciones y solo

```
17 @api.depends("fechaNacimiento")
18 def _calcula_edad(self):
19     for registro in self:
20         if registro.fechaNacimiento: # si existe el registro...
21             # códigoPropio
22             edad = 99
23             registro.edad = edad
24
25 @api.depends("fechaNacimiento")
26 def _calcula_signo(self):
27     try:
28         self.ensure_one()
29         # códigoPropio
30         self.signo = "Signo TBD"
31     except Exception:
32         print("Varios registros en el dataset de _calcula_signo")
```

realice la función definida cuando realmente le llegue un registro. Una vez efectuado el cambio, reiniciamos el servicio y procedemos a instalar el módulo. Podemos comprobar que ha salido bien ya que en la tabla figuran los tres nuevos campos.

edad
signo
fechaNacimiento

3. Modificación de views.xml

Mediante la modificación de este archivo, lograremos visualizar las funciones desarrolladas en esta tarea.

```
1 <odoo>
2   <record id="res_partner_zodiaco_form" model="ir.ui.view">
3     <field name="name">res.partner.form.inherit.base</field>
4     <field name="model">res.partner</field>
5     <field name="inherit_id" ref="base.view_partner_form" />
6     <field name="arch" type="xml">
7       <xpath expr="//notebook" position="inside">
8         <page string="Zodiaco">
9           <group>
10            <field name="fechaNacimiento" string="Fecha de nacimiento" />
11            <field name="edad" string="Edad" />
12            <field name="signo" string="Signo del zodiaco" />
13          </group>
14        </page>
15      </xpath>
16    </field>
17  </record>
18
19  <record id="res_partner_zodiaco_tree" model="ir.ui.view">
20    <field name="name">res.partner.tree.inherit.base</field>
21    <field name="model">res.partner</field>
22    <field name="inherit_id" ref="base.view_partner_tree" />
23    <field name="arch" type="xml">
24      <xpath expr="//tree/field[@name='display_name']" position="after">
25        <field name="signo" string="Signo del zodiaco" />
26      </xpath>
27    </field>
28  </record>
29
30  <record id="res_partner_zodiaco_kanban" model="ir.ui.view">
31    <field name="name">res.partner.kanban.inherit.base</field>
32    <field name="model">res.partner</field>
33    <field name="inherit_id" ref="base.res_partner_kanban_view" />
34    <field name="arch" type="xml">
35      <xpath expr="//kanban/templates/t/div/div[2]/div/strong" position="after">
36        <p></p>
37        <field name="signo" />
38      </xpath>
39    </field>
40  </record>
41 </odoo>
```

Tras corregir un par de errores de escritura y reinstalar el módulo, podemos comprobar que lo que actualmente está desarrollado se ejecuta correctamente:

Contacts & Addresses		Sales & Purchase	Invoicing	Internal Notes	Zodiaco
Fecha de nacimiento ?	05/01/2023				
Edad ?	99				
Signo del zodiaco ?	Signo TBD				



A Lira de Ribadavia
Signo TBD
Ribadavia, Spain
banda@lira.gal

<input type="checkbox"/>	Name	Signo del zodiaco	Phone
<input type="checkbox"/>	A Lira de Ribadavia	Signo TBD	+34988658976
<input type="checkbox"/>	A Lira de Ribadavia, Ramón Estévez Casasnovas		+34988471710

4. Código para mostrar los datos

En esta tarea se solicita escribir el código necesario para dos tareas. Sería el siguiente

4.1. Cálculo de la edad

```
15     @api.depends("fechaNacimiento")
16     def _calcula_edad(self):
17         for registro in self:
18             if registro.fechaNacimiento:
19                 hoy = date.today()
20                 edad = hoy.year - registro.fechaNacimiento.year - ((hoy.month, hoy.day) <
21                 (registro.fechaNacimiento.month, registro.fechaNacimiento.day))
22                 registro.edad = edad
```

Itero sobre cada registro recibido (`self`). Si la fecha de nacimiento está registrada, calculo la diferencia de años entre la fecha actual y la de nacimiento. Posteriormente, resto (o no) 1 en función del mes y día, mediante un comparador `true-false`.

4.2. Cálculo del signo del zodiaco

```
23     @api.depends("fechaNacimiento")
24     def _calcula_signo(self):
25         for registro in self:
26             try:
27                 self.ensure_one()
28                 if registro.fechaNacimiento:
29                     mes = registro.fechaNacimiento.month
30                     dia = registro.fechaNacimiento.day
31
32                     if (mes == 1 and dia >= 20) or (mes == 2 and dia <= 18):
33                         registro.signo = "Acuario"
34                     elif (mes == 2 and dia >= 19) or (mes == 3 and dia <= 20):
35                         registro.signo = "Piscis"
36                     elif (mes == 3 and dia >= 21) or (mes == 4 and dia <= 19):
37                         registro.signo = "Aries"
38                     elif (mes == 4 and dia >= 20) or (mes == 5 and dia <= 20):
39                         registro.signo = "Tauro"
40                     elif (mes == 5 and dia >= 21) or (mes == 6 and dia <= 20):
41                         registro.signo = "Géminis"
42                     elif (mes == 6 and dia >= 21) or (mes == 7 and dia <= 22):
43                         registro.signo = "Cáncer"
44                     elif (mes == 7 and dia >= 23) or (mes == 8 and dia <= 22):
45                         registro.signo = "Leo"
46                     elif (mes == 8 and dia >= 23) or (mes == 9 and dia <= 22):
47                         registro.signo = "Virgo"
48                     elif (mes == 9 and dia >= 23) or (mes == 10 and dia <= 22):
49                         registro.signo = "Libra"
50                     elif (mes == 10 and dia >= 23) or (mes == 11 and dia <= 21):
51                         registro.signo = "Escorpio"
52                     elif (mes == 11 and dia >= 22) or (mes == 12 and dia <= 21):
53                         registro.signo = "Sagitario"
54                     else:
55                         registro.signo = "Capricornio"
56             except:
57                 registro.signo = "Signo TBD"
58         except Exception:
59             print("Error al calcular el signo zodiacal.")
```

En este otro caso, primero me aseguro de recibir únicamente un único registro; en caso contrario, lanzo una excepción. Tras recibirlo, comparo contra el mes y día de la fecha de nacimiento y le asigno su signo correspondiente. Añado una condición final donde dejo Signo TDB; si en algún momento figurase ese signo significaría que la lógica superior no sería correcta. Lo escribí mientras probaba el código y lo dejé como seguro final, aunque no sería necesario.

En esta imagen vemos que solo se ejecuta el código sobre un contacto.



En esta otra vista vemos los tres campos ya informados.

Contacts & Addresses	Sales & Purchase	Invoicing	Internal Notes	Zodiaco
Fecha de nacimiento ?	05/22/1987			
Edad ?	35			
Signo del zodiaco ?	Géminis			

En la lista figuran también los signos de las fechas de nacimiento informadas:

<input type="checkbox"/>	Name	Signo del zodiaco	Phone
<input type="checkbox"/>	A Lira de Ribadavia	Géminis	+34988658976
<input type="checkbox"/>	A Lira de Ribadavia, Ramón Estévez Casasnovas		+34988471710
<input type="checkbox"/>	Administrator	Tauro	
<input type="checkbox"/>	Festas Galicia		+34981987654
<input type="checkbox"/>	Festas Galicia, Sabela Sabucedo Rego		
<input type="checkbox"/>	IES de Teis		+34886120486
<input type="checkbox"/>	Queremos Charangueo		986123456
<input type="checkbox"/>	Queremos Charangueo, Marisol Arias Rego		+34986658976

Por último, comprobamos cómo se ve la información en la base de datos:

Query

Query History

1

select id, edad, signo, "fechaNacimiento"

2

from res_partner;

Data Output

Messages

Notifications