1 Desenvolvemento de componentes

Índice

1.1 Sumario

1.2 Introdución a Python

Para crear novos módulos en Odoo utilizamos a linguaxe de programación Python.

Introdución a la programación con Python - Andrés Marzal e Isabel Gracia

The Python Language Reference

The Python Standard Library

- Practicamos creando un pequeno programa, *par.py*, que recolle números do teclado e responde se son pares ou impares:

```
 GNU nano 2.2.6                File: par.py

#!/usr/bin/python
#coding: latin-1

print 'Insire "0" para finalizar'
print 'Insire un número:',
numero = int(raw_input())

while numero != 0:

        print 'O número', numero, 'é',
        if (numero % 2 == 0):
                print 'par.\n'
        else:
                print 'impar.\n'

        print 'Insire un número:',
        numero = int(raw_input())
```

```
root@R-US-0:/home/administrador# python par.py
Insire "0" para finalizar
Insire un número: 7
O número 7 é impar.

Insire un número: 4
O número 4 é par.

Insire un número: 0
root@R-US-0:/home/administrador#
```

- Creamos unha segunda versión cunha función propia:

```
 GNU nano 2.2.6                File: par_f.py

#!/usr/bin/python
#coding: latin-1

def par (n):
        if (n % 2 == 0):
                resposta = 'par'
        else:
                resposta = 'impar'
        return resposta


print 'Insire "0" para finalizar'
print '\nInsire un número:',
numero = int(raw_input())

while numero != 0:
        print 'O número', numero, 'é', par(numero)

        print '\nInsire un número:',
        numero = int(raw_input())
```

```
root@R-US-0:/home/administrador# python par_f.py
Insire "0" para finalizar

Insire un número: 99
O número 99 é impar

Insire un número: 88
O número 88 é par

Insire un número: 0
root@R-US-0:/home/administrador#
```

- Para finalizar, creamos unha terceira versión formada por:
  - Un módulo de funcións propias, chamado *funcions_novagalaxia.py* onde gardamos a función creada no punto anterior.
  - O programa *par_final.py* que importa e utiliza esa función.

```
 GNU nano 2.2.6          File: funcions_novagalaxia.py

#!/usr/bin/python
#coding: latin-1

def par (n):
        if (n % 2 == 0):
                resposta = 'par'
        else:
                resposta = 'impar'
        return resposta
```

```
 GNU nano 2.2.6              File: par_final.py

#!/usr/bin/python
#coding: latin-1

from funcions_novagalaxia import par

print 'Insire "0" para finalizar'
print '\nInsire un número:',
numero = int(raw_input())

while numero != 0:
        print 'O número', numero, 'é', par(numero)

        print '\nInsire un número:',
        numero = int(raw_input())
```

1.3 Creación de módulos Odoo

Vamos crear un módulo novo: unha axenda telefónica.

- Movémonos ao directorio */usr/lib/python2.7/dist-packages/openerp/addons*, onde están todos os módulos da aplicación.
- Creamos o directorio *axenda*.
- Dentro del, creamos o ficheiro *__init__.py* que chama ao ficheiro principal do noso paquete: *axenda.py*. Grazas a *__init__.py*, o módulo é recoñecido como tal por Odoo.

```
 GNU nano 2.2.6              File: __init__.py

import axenda
```

- Creamos o ficheiro *__openerp__.py* coa descrición do módulo (dicionario):

```
 GNU nano 2.2.6          File: __openerp__.py                    Modified

#!/usr/bin/python
#coding: utf-8

{
    'name': 'Axenda',
    'author': 'Marcos',
    'category': 'Uncategorized',
    'version': '1.0',
    'description': 'Módulo personalizado para xestionar unha axenda telefónica',
    'depends': ['base'],
    'init_xml': [],
    'update_xml': ['axenda_view.xml'],
    'active': False,
    'installable': True,
}
```

- Creamos o ficheiro *axenda.py* (modelo e controlador). Define a clase *axenda* que é unha táboa na BD con varias columnas:

```
GNU nano 2.2.6                File: axenda.py

#!/usr/bin/python
#coding: utf-8

from openerp.osv import osv, fields

class axenda(osv.osv):
    _name = 'axenda'
    _columns = {
        'nome': fields.char('Nome', size=64, required=True),
        'telefono': fields.char('Teléfono', size=16, required=True),
        'sexo': fields.selection([('h','Home'),('m','Muller')], 'Sexo'),
        'facturacion': fields.float('Facturación anual media'),
        'dispoñible': fields.boolean('Dispoñible')
    }

axenda()
```

- Unha forma de comprobar que non haxa erros nos ficheiros **.py*, é executándoos con *python*:

```
python __init__.py
python __openerp__.py
python axenda.py
```

- Creamos a vista en *axenda_view.xml* (dunha forma parecida a como fixeramos en Creación de vistas en Odoo), formada por tres elementos:

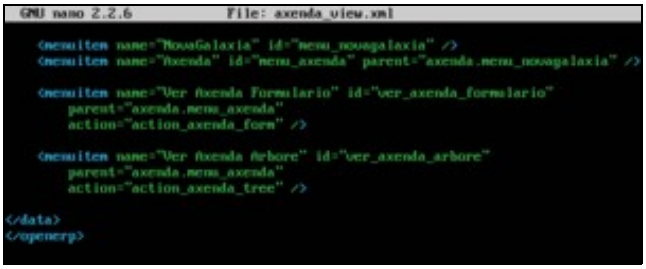
- ♦ Unha vista de formulario e outra vista de árbore:

```
GNU nano 2.2.6                File: axenda_view.xml

<?xml version="1.0" encoding="utf-8" ?>
<openerp>
<data>

    <record model="ir.ui.view" id="axenda_vista_formulario">
        <field name="name">axenda.vista.formulario</field>
        <field name="model">axenda</field>
        <field name="type">form</field>
        <field name="priority" eval="5" />
        <field name="arch" type="xml">
            <form string="Axenda - Vista Formulario">
                <field name="nome" select="1" string="Nome" />
                <field name="telefono" select="1" string="Teléfono" />
                <field name="sexo" select="1" string="Sexo" />
                <field name="facturacion" select="1" string="Facturación anual $
                <field name="dispoñible" select="1" string="Dispoñible" />
            </form>
        </field>
    </record>

    <record model="ir.ui.view" id="axenda_vista_arbore">
        <field name="name">axenda.vista.arbore</field>
        <field name="model">axenda</field>
        <field name="type">tree</field>
        <field name="priority" eval="5" />
        <field name="arch" type="xml">
            <tree string="Axenda - Vista Arbore">
                <field name="nome" select="1" string="Nome" />
                <field name="telefono" select="1" string="Teléfono" />
                <field name="sexo" select="1" string="Sexo" />
                <field name="facturacion" select="1" string="Facturación anual $
                <field name="dispoñible" select="1" string="Dispoñible" />
            </tree>
        </field>
    </record>
```

- ♦ Accións. Ao facer clic nunha opción do menú, a acción vai abrir a vista correspondente):

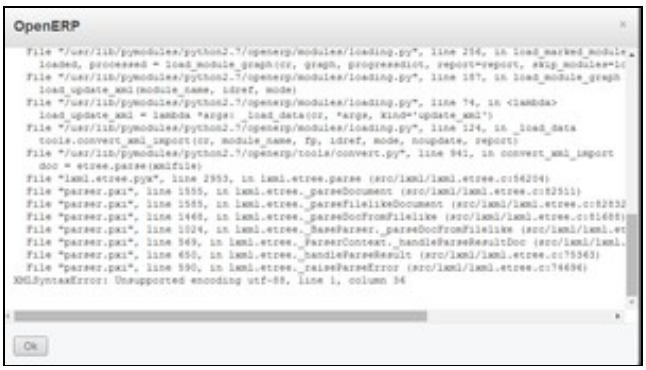
```
GNU nano 2.2.6                File: axenda_view.xml

    <record model="ir.actions.act_window" id="action_axenda_form">
        <field name="name">Información Axenda (Formulario)</field>
        <field name="type">ir.actions.act_window</field>
        <field name="res_model">axenda</field>
        <field name="view_type">form</field>
        <field name="view_mode">tree,form</field>
    </record>

    <record model="ir.actions.act_window" id="action_axenda_tree">
        <field name="name">Información Axenda (Arbore)</field>
        <field name="type">ir.actions.act_window</field>
        <field name="res_model">axenda</field>
        <field name="view_type">tree</field>
        <field name="view_mode">tree,form</field>
    </record>
```

- ♦ Menú, submenú e opcións:



- En Odoo: actualizamos a lista de módulos, facemos clic en *Módulos locales*, procuramos o módulo *axenda* e o instalamos.

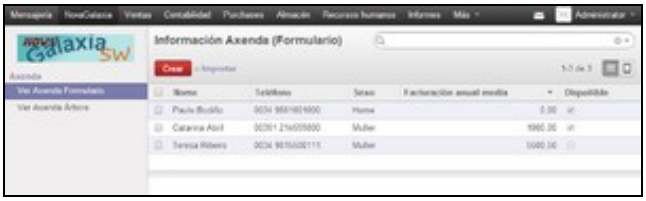
- Se houber erros no ficheiro *xml*, saberémolo neste momento (a mensaxe de erro é moi longa pero temos que fixarnos na última liña). Neste exemplo especificouse mal a codificación *utf-8*:



- Outro erro típico: "Ha ocurrido un error mientras se validaban los campos arch: Invalid XML for View Architecture!". A causa é esquecer inserir espazos en branco antes das "/" de cerre.

- Se todo vai ben, deber aparecer o novo menú *Axenda* coas dúas vistas deseñadas. Inserimos datos na axenda:

- E para finalizar, comprobamos o resultado:



Fontes: [1] [2] [3]