

Herdanza nos modelos

Herdanza por extensión: persoa con res.partner

Exemplo de Herdanza por extensión

Neste exemplo veremos:

1. Como se modifica o código python.
2. Como se modifica a vista XML.
3. Unha aclaración importante para poder usar o exemplo. (ler antes de modificar o código)

1. Modificando o código

Neste suposto de herdanza por extensión, poderíamos engadir e modificar métodos existentes, ou engadir novos atributos (que serán novos campos na táboa xa existente). Para elo imos crear unha clase nova (persoa) no noso módulo.

No seguinte exemplo imos engadir en `res_partner` dous campos novos:

- **apelidos**: Orixinalmente en Odoo, só temos o campo "name" de `res_partner` para almacenar os valores de nome e apelidos. A idea é crear o campo "apelidos" para telos separados de "name" (onde almacenaremos só o nome).
- **visible**: É un campo que usaremos para poder filtrar os valores de `res_partner` en función deste campo boolean, para elo usaremos **domain**.

Por exemplo podemos ver no modelo **pedido.py** :

```
persoa_id = fields.Many2one('res.partner', ondelete='set null', domain="[('visible'
```

Definimos a clase **persoa** no arquivo **persoa.py**:

```
class persoa (models.Model):
    _inherit = 'res.partner' # como non lle proporcionamos _name facemos herdanza p
                             # por tanto non se crea unha nova táboa e os atributos
                             # No xml temos que estender as vistas da clase pai

    apellidos = fields.Char(required=True,size=60,string="Apellidos")
    visible = fields.Boolean(string="¿Visible?", default=False)

    # O campo visible serve para visualizar só os rexistros que nos interesan doma
    # Engadimos na relación many2one ou many2many unha condición

    # Temos que dar permisos en ir.model.access.csv para o model res_partner.
    # Engadir a liña:
    # odoo_basico_res_partner,odoo_basico.res_partner,model_res_partner,base.group_u
```

Agora xa poderíamos ter os valores nome e apellidos separados, pero como tamén queremos telos xuntos noutro campo, o que imos facer é sobrescribir o método existente:

```
def name_get(self):
```

Cando Odoo nos mostra os rexistros de res_partner (mediante unha relación) o que visualizamos é o contido do campo "display_name". Que é un campo computed baseado no contido do campo "name" e "company_id"

Como vemos a continuación, na clase persoa.py sobrescribimos o método **def name_get(self)**: Coa idea de concatenar os campos "name" e "apellidos". Así actualizarase o campo "display_name" e poderemos visualizar xuntos os campos "name" e "apellidos".

```
def name_get(self): #sobrescribimos o método name_get da clase res.partner
    # Por defecto o campo display_name é company, name
    # O metodo name_get actualiza o campo display_name(que é un
    resultado = []
    for rexistro in self:
        if rexistro.apellidos:
            resultado.append((rexistro.id, str(rexistro.name) + " " + str(rexistro.
        else:
            resultado.append((rexistro.id, str(rexistro.name)))
    return resultado
```

The screenshot shows a web application interface. At the top, there is a navigation bar with 'Menú Nivel 0' and 'Menú Nivel 1'. Below this, the page title is 'A acción de pedido / Pedido 1'. There are two buttons: 'Guardar' (Save) and 'Descartar' (Discard). Below the buttons, there are two tabs: 'Actualiza o campo Sexo do modelo información' and 'Actualiza Hora Timezone do Usuario en todos os rexistros do modelo información'. The main content area is divided into two sections. On the left, there is a large, empty, light gray rectangular area. On the right, there is a form titled 'Identificador' with a dropdown menu showing 'Persoa'. Below this, there is a section titled 'Visualizar e engadir' (View and add) with a sub-section 'A Descrición' (The description) containing three lines of text: 'línea 1 de pedido 1', 'línea 2 de pedido 1', and 'Agregar línea'. To the right of this section, there is a dropdown menu with the following options: 'Pedido 1', 'Noam Chomsky', 'Nome Escritura', 'Nome Lectura', 'Nome Persoa Apelidos Persoa', 'nome apelido', and 'Crear y editar...'. The 'Noam Chomsky' option is highlighted with a red box.

2. Modificando a vista XML

Temos que modificar a vista XML de `res_partner` para determinar onde solicitar os novos campos "apelidos" e "visible".

Isto facemolo no arquivo "**persoa.xml**".

```
<record model="ir.ui.view" id="persoa_form_view">
  <field name="name">persoa.form</field>
  <field name="model">res.partner</field>
  <field name="inherit_id" ref="base.view_partner_form"/>
  <field name="type">form</field>
  <field name="arch" type="xml">
    <data>
      <field name="name" position="after">
        <group>
          <field name="apelidos" />
          <field name="visible" />
        </group>
      </field>
    </data>
  </field>
</record>
```

Herdamos a vista de res_partner mediante o seu id externo "base.view_partner_form".

Situamonos na vista para engadir os dous campos mediante:

```
<field name="name" position="after">
```

Vemos o resultado:

Crear: Persoa

Individual Compañía

0,00 Facturado 0 Cuentas ana... Activo

Noam

Apellidos Chomsky

¿Visible? ☒

Compañía

Dirección

Calle... Calle 2... Ciudad Provincia C.P. País

NIF Por ejemplo, ESA00000000

Puesto de trabajo Por ejemplo, Director de Ventas

Teléfono

Móvil

Correo electrónico

Sitio web e.j. https://www.odoo.com

Título

Idioma Spanish / Español

Etiquetas...

Contatos y direcciones Notas internas Ventas y Compras Facturación / Contabilidad

Añadir

Guardar Descartar

Se temos instalados módulos de Odoo con referencia a clientes,proveedores..., vemos como tamén se ve afectada a vista de res_partner:

Facturación / Contabilidad Clientes Proveedores Informes Configuración

Clientes / Noam Chomsky

Editar Crear

Acción

Noam

Apellidos Chomsky

¿Visible? ☒

Dirección

NIF

Puesto de trabajo

Teléfono

Móvil

Correo electrónico

Sitio web

Título

0,00 € Facturado



Mediante o arquivo **"templates.xml"** podemos engadir rexistros na táboa `res_partner`. (Ollo cos campos `"apelidos"` e `"visible"` que son os que creamos mediante a clase `persoa.py`).

```
<record model="res.partner" id="partner_persoa">
    <field name="name">Nome Persoa</field>
    <field name="apelidos">Apelidos Persoa</field>
    <field name="visible">true</field>
    <field name="display_name">Nome Persoa Apelidos Persoa</field>
    <field name="email">persoa@iesteis.com</field>
    <field name="active">true</field>
    <field name="company_id">1</field>
    <field name="lang">es_ES</field>
</record>
```

3. Aclaración para poder usar o exemplo

Para poder usar este exemplo, temos que ter en conta o seguinte:

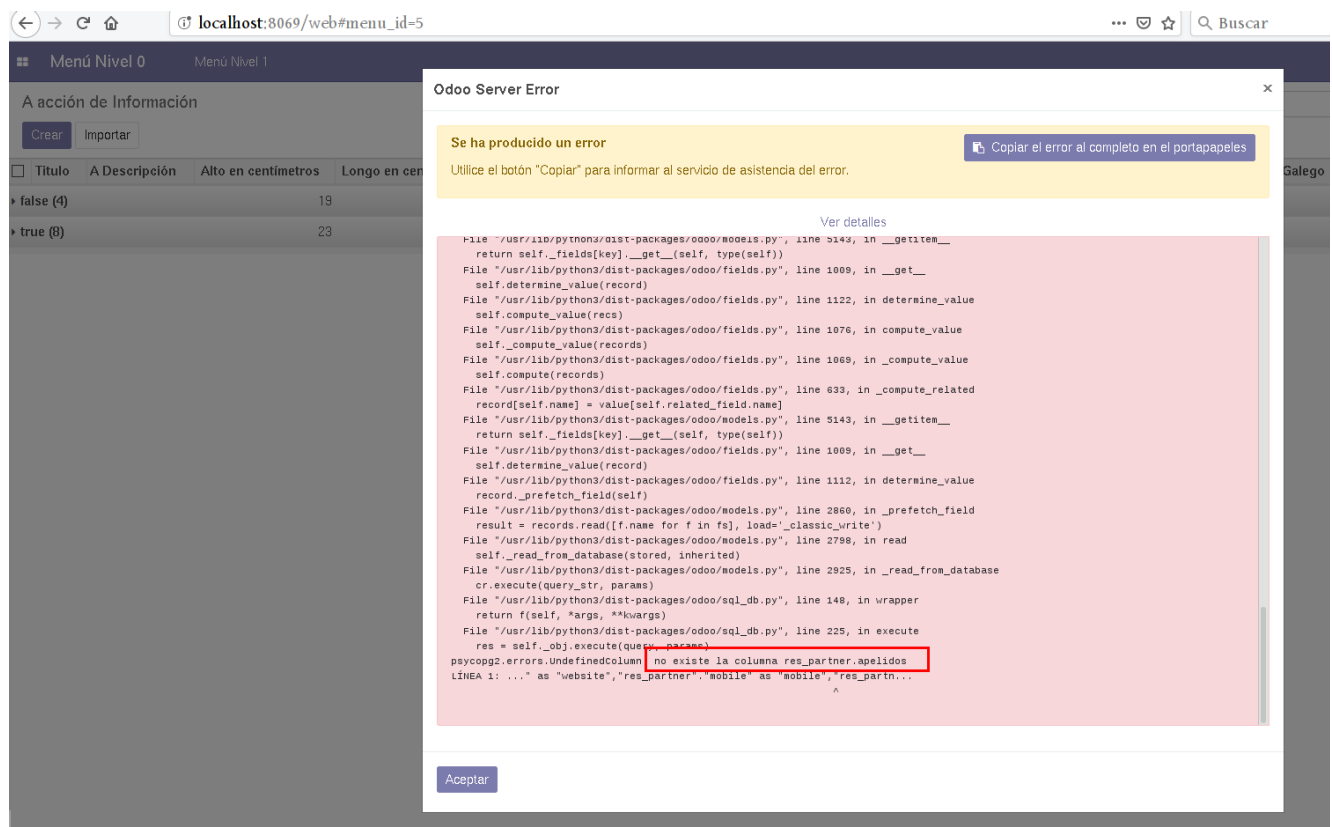
- No arquivo `__init__.py` está comentada a referencia a clase `"persoa"` `#from . import persoa`
- No apartado `'data'` do arquivo `__manifest__.py` está comentada a referencia a vista `#'views/persoa.xml'`,
- No arquivo **"templates.xml"** están comentadas as liñas do `<record model="res.partner"` que fan referencia aos campos `<field name="apelidos">` e `<field name="visible">`
- No arquivo **"ir.model.access.csv"** , engadir a seguinte liña:
`odoo_basico_res_partner,odoo_basico.res_partner,model_res_partner,base.group_u`
- No arquivo **pedido.py** na relación `persoa_id` podemos usar a definición na que usamos o campo `visible` que inicialmente está comentada:

usamos o campo `visible` que inicialmente esta comentada.

```
persoa_id = fields.Many2one('res.partner', ondelete='set null', domain="[('vis
```

Unha vez descomentado todo o anterior e engadida a liña en "ir.model.access.csv", teríamos que reiniciar o servizo odoo (systemctl restart odoo).

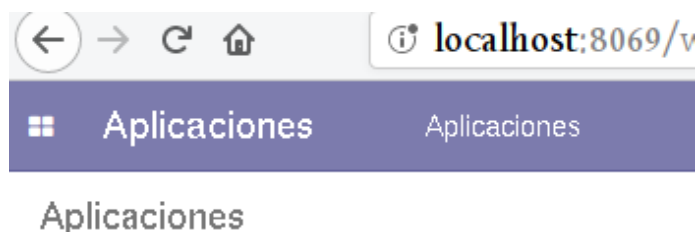
Pero ao facer isto, se queremos navegar polos menús, Odoo xerará un erro similar ao seguinte:



Xa que o noso código agora fai referencia aos campos "apelidos" e "visible" e como non actualizamos o módulo os campos aínda non existen.

Para elo temos dúas posibles solucións, **antes de reiniciar o servizo de Odoo**:

1. Desinstalar o módulo. Reiniciar o servizo. E facer unha nova instalación.
2. Se no momento de reiniciar o servizo de Odoo, xa estamos situados na pantalla de actualización do módulo, podremos actualizalo sen problemas.





Se non o fixemos ben e temos o erro. Temos que volver poñer o código como estaba, reiniciar de novo o servizo de odoo, tentar de novo e **facelo ben ;)**

Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](#)