

Herdanza nos modelos

Herdanza nos modelos

Herdanza nos modelos

A hora de traballar con modelos, podemos ter tres tipos de herdanza.

<https://fundamentos-de-desarrollo-en-odoo.readthedocs.io/capitulos/herencia-extendiendo-funcionalidad-aplicaciones-existentes.html>

1. Herdanza por extensión: Na que podemos engadir e modificar métodos existentes, ou engadir novos atributos (que serán novos campos na táboa xa existente).
 - Veremos o exemplo de herdanza entre clases: persoa con res.partner

- ```
class persoa (models.Model):
 _inherit = 'res.partner'
```

2. Herdanza por delegación: Na que se creará unha táboa nova na que se engadirán os novos atributos definidos. Neste caso só se accede aos atributos da clase da que se herda NON se accede aos métodos.

- exemplo res\_users con res\_partner. Uns campos están almacenados en res\_users e outros en res\_partner.

- ```
class User(models.Model):  
    _name = 'res.users'  
    _inherits = {'res.partner': 'partner_id'}  
    partner_id = fields.Many2one('res.partner')
```

3. Herdanza por prototipo. Neste caso podemos acceder aos métodos da clase da que herdamos e se crea unha nova táboa na que se copian todos os campos da táboa da que herdamos. Pero as táboas serán independentes, non estarán relacionadas.

O normal é usala para herdar de modelos abstractos(plantillas). Non de modelos que teñan datos porque poderíamos ter os mesmos datos almacenados en dúas táboas distintas.

```
○ class TodoTask(models.Model):  
    _name = 'todo.task'  
    _inherit = 'mail.thread'
```

A ter en conta!!! Se especificamos **_name** crearase unha nova táboa (co nome que lle facilitamos). Se especificamos **_inherit** temos acceso aos métodos da clase que herdamos. Se especificamos **_inherits** non temos acceso aos métodos e manteremos unha relación entre as táboas.

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