

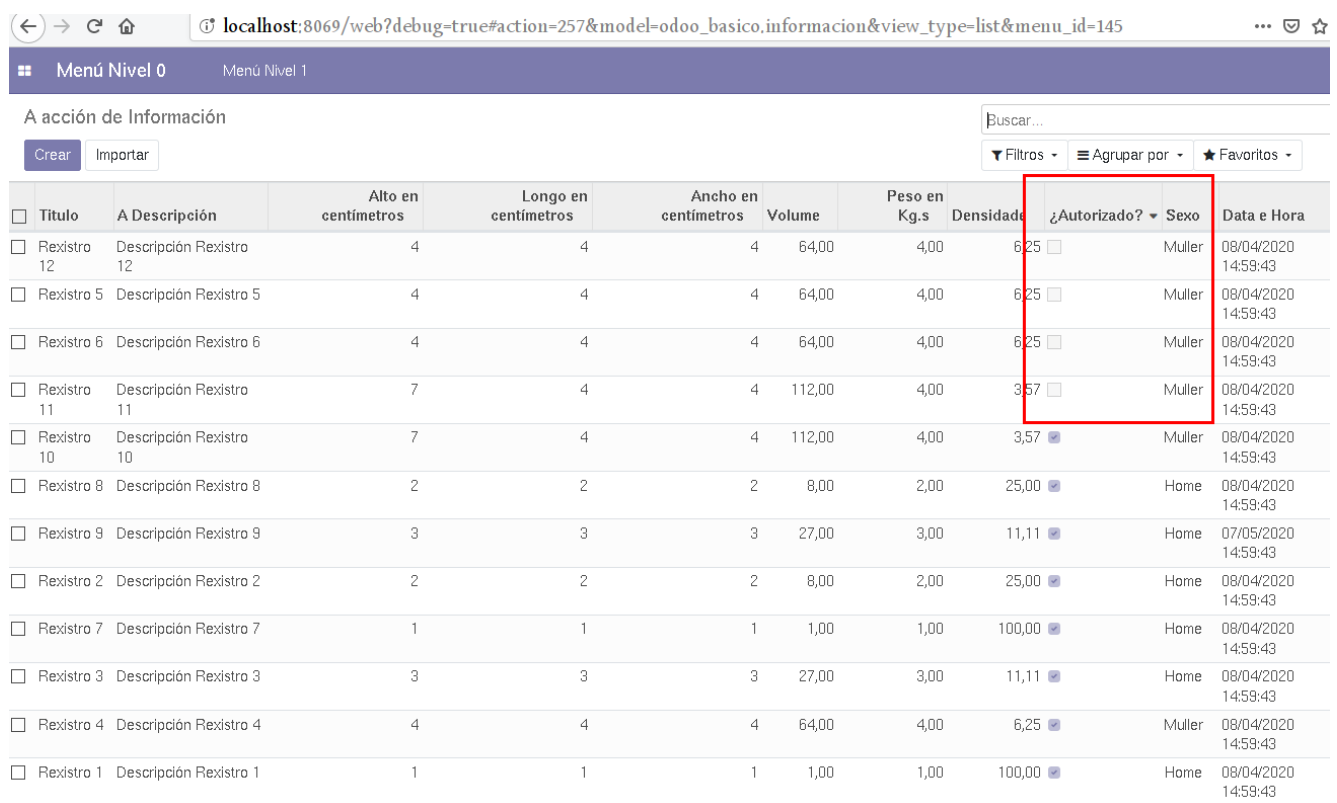
## Acceso dende un modelo ao código doutro modelo

## Acceso dende un modelo ao código doutro modelo

### Acceso dende un modelo ao código doutro modelo

A idea é actualizar dende o modelo "pedido" datos do modelo "informacion".

Actualizaremos de xeito masivo o campo `sexo_traducido` co valor "Hombre", todos os rexistros que teñan o valor do campo `autorizado` a false.



localhost:8069/web?debug=true#action=257&model=odoo\_basico.informacion&view\_type=list&menu\_id=145

Menú Nivel 0 Menú Nivel 1

A acción de Información

Crear Importar

Buscar...

Filtros Agrupar por Favoritos

	Título	A Descripción	Alto en centímetros	Longo en centímetros	Ancho en centímetros	Volume	Peso en Kg.s	Densidade	¿Autorizado? ▾	Sexo	Data e Hora
<input type="checkbox"/>	Rexistro 12	Descripción Rexistro 12	4	4	4	64,00	4,00	6,25	<input type="checkbox"/>	Muller	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 5	Descripción Rexistro 5	4	4	4	64,00	4,00	6,25	<input type="checkbox"/>	Muller	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 6	Descripción Rexistro 6	4	4	4	64,00	4,00	6,25	<input type="checkbox"/>	Muller	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 11	Descripción Rexistro 11	7	4	4	112,00	4,00	3,57	<input type="checkbox"/>	Muller	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 10	Descripción Rexistro 10	7	4	4	112,00	4,00	3,57	<input checked="" type="checkbox"/>	Muller	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 8	Descripción Rexistro 8	2	2	2	8,00	2,00	25,00	<input checked="" type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 9	Descripción Rexistro 9	3	3	3	27,00	3,00	11,11	<input checked="" type="checkbox"/>	Home	07/05/2020 14:59:43
<input type="checkbox"/>	Rexistro 2	Descripción Rexistro 2	2	2	2	8,00	2,00	25,00	<input checked="" type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 7	Descripción Rexistro 7	1	1	1	1,00	1,00	100,00	<input checked="" type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 3	Descripción Rexistro 3	3	3	3	27,00	3,00	11,11	<input checked="" type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 4	Descripción Rexistro 4	4	4	4	64,00	4,00	6,25	<input checked="" type="checkbox"/>	Muller	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 1	Descripción Rexistro 1	1	1	1	1,00	1,00	100,00	<input checked="" type="checkbox"/>	Home	08/04/2020 14:59:43

Podemos acceder dende un modelo ao código (métodos) que temos definido noutro modelo.

Por exemplo, imos acceder dende o modelo "pedido" ao método "`_cambia_campo_sexo`" que temos definido no modelo "informacion".

- No modelo "informacion" teremos definido o metodo "`_cambia_campo_sexo`" que será chamado dende o modelo "pedido"

será chamado dende o modelo "pedido".

- Definiremos un botón na vista "pedido.xml", coa finalidade de poder executar dende o modelo "pedido", a chamada ao método "\_cambia\_campo\_sexo", que está definido no modelo "informacion"
- No modelo "pedido" faremos a chamada ao método "\_cambia\_campo\_sexo", que está definido no modelo "informacion"

Definimos no modelo **"informacion"** o método que será chamado dende o modelo "pedido".

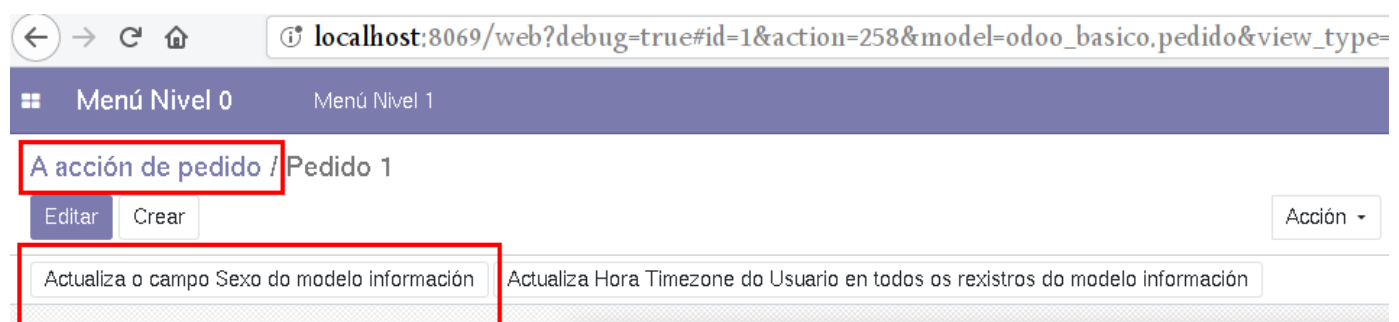
```
def _cambia_campo_sexo(self, rexistro):  
    rexistro.sexo_traducido = "Hombre"
```

Como vemos o método pon o valor "Hombre" no campo "sexo\_traducido", do rexistro que lle pasamos como parámetro.

Vemos que na definición do método temos dous parámetros, mentras que cando fagamos chamada só lle pasaremos un (que será o rexistro a actualizar):

1. self : "En python,o primeiro parámetro de todos os métodos, sempre é unha referencia a obxeto que chama ao método. Por convención chámase "self" <https://pythontips.com/2013/08/07/the-self-variable-in-python-explained/>
2. rexistro: Será o rexistro que queremos actualizar.

Definimos no "header" dentro da vista **"pedido.xml"**, o "botón" que usaremos para executar o método do modelo "pedido" que nos interese:



	Pedido 1
	<b>Visualizar e engadir</b>
	A Descrición
	linea 1 de pedido1
	linea 2 de pedido 1

Vemos na definición do botón que o valor de name nos indica o método dentro do model "pedido" que vai ser chamado ao premer o botón.

```
<button name="actualizadorSexo" type="object" string="Actualiza o campo Sexo do mode
```

```
<record model="ir.ui.view" id="pedido_form_view">
  <field name="name">pedido.form</field>
  <field name="model">odoo_basico.pedido</field>
  <field name="arch" type="xml">
    <form string="Formulario pedido">
      <header>
        <button name="actualizadorSexo" type="object" string="Actualiza o campo
        <button name="actualizadorHoraTimezone" type="object" string="Actualiza
      </header>
    <sheet>
      <field name="name"/>
      <separator string="Visualizar e engadir"/>
      <field name="lineapedido_ids"/>
        <!-- Non permite editar -->
        <!-- <separator string="Só visualizar"/>-->
        <!-- <field name="lineapedido_ids" readonly="1"/> -->
    </sheet>
  </form>
</field>
</record>
```

No modelo "**pedido**" facemos a chamada ao método  
"**\_cambia\_campo\_sexo**", definido no modelo "informacion"

Como vimos na vista "pedido.xml" o nome do método que é chamado dende o botón chamase: `actualizadorSexo()`

```
def actualizadorSexo(self):  
    informacion_ids = self.env['odoo_basico.informacion'].search([('autorizado', '=',  
    for rexistro in informacion_ids:  
        self.env['odoo_basico.informacion']._cambia_campo_sexo(rexistro)
```

Como vemos no código do método, facemos uso do "environment" mediante "self.env" para acceder aos datos do modelo "informacion".

Temos varios funcionamentos predefinidos para "self.env", como podemos ver no seguinte enlace: <https://odoo-new-api-guide-line.readthedocs.io/en/latest/environment.html#common-actions>

Neste caso usamos "search", aplicandolle un filtro (por exemplo que o campo autorizado sexa false).

Isto nos devolve todos os rexistros do modelo "informacion" que cumplan esa condición, almacenamos o resultado na variable "informacion\_ids"

Finalmente, mediante un bucle accedemos a cada rexistro almacenado en "informacion\_ids".

Neste bucle chamamos ao método do modelo "informacion" que nos interese (neste caso "\_cambia\_campo\_sexo") usando "self.env"

Podedes ver que lle pasamos como parámetro a ese método o "rexistro" que terá que actualizar. E como decíamos no primeiro apartado non é necesario pasar o parámetro "self" (x que python o pasa automaticamente), pero como neste caso "self" fai referencia ao obxecto da clase "pedido" non nos serve e temos que pasar "rexistro" que é onde temos a referencia a obxecto da clase "informacion".

```
self.env['odoo_basico.informacion']._cambia_campo_sexo(rexistro)
```

Podemos ver como queda despois de actualizar:



<input type="checkbox"/>	Título	A Descrición	Alto en centímetros	Longo en centímetros	Ancho en centímetros	Volume	Peso en Kg.s	Densidade	¿Autorizado? ▼	Sexo	Data e Hora
<input type="checkbox"/>	Rexistro 5	Descrición Rexistro 5	4	4	4	64,00	4,00	6,25	<input type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 12	Descrición Rexistro 12	4	4	4	64,00	4,00	6,25	<input type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 6	Descrición Rexistro 6	4	4	4	64,00	4,00	6,25	<input type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 11	Descrición Rexistro 11	7	4	4	112,00	4,00	3,57	<input type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 10	Descrición Rexistro 10	7	4	4	112,00	4,00	3,57	<input checked="" type="checkbox"/>	Muller	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 8	Descrición Rexistro 8	2	2	2	8,00	2,00	25,00	<input checked="" type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 9	Descrición Rexistro 9	3	3	3	27,00	3,00	11,11	<input checked="" type="checkbox"/>	Home	07/05/2020 14:59:43
<input type="checkbox"/>	Rexistro 2	Descrición Rexistro 2	2	2	2	8,00	2,00	25,00	<input checked="" type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 7	Descrición Rexistro 7	1	1	1	1,00	1,00	100,00	<input checked="" type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 3	Descrición Rexistro 3	3	3	3	27,00	3,00	11,11	<input checked="" type="checkbox"/>	Home	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 4	Descrición Rexistro 4	4	4	4	64,00	4,00	6,25	<input checked="" type="checkbox"/>	Muller	08/04/2020 14:59:43
<input type="checkbox"/>	Rexistro 1	Descrición Rexistro 1	1	1	1	1,00	1,00	100,00	<input checked="" type="checkbox"/>	Home	08/04/2020 14:59:43

En resumo, mediante "self.env" podemos acceder a outro modelo, e podemos executar o métodos nel definidos ou unha serie de métodos xa predefinidos como os que vimos no enlac anteriormente exposto.

## Crear un novo rexistro

Para crear un novo rexistro engadimos no header da vista **pedido.xml** un botón que chamará ao método **creaRexistroInformacion**

E no modelo **pedido.py** engadimos o seguinte código:

```
def creaRexistroInformacion(self):
    creado_id = self.env['odoo_basico.informacion'].create({'name': 'Creado den
    creado_id.descripcion = "Creado dende o modelo pedido"
    creado_id.autorizado = False
```

Usamos **self.env** para **odoo\_basico.informacion**, que é o modelo no que queremos crear un novo rexistro.

E mediante o metodo **create** creamos un novo rexistro pasandolle como parámetro o name que vai ter.

Este novo obxecto creado almacenámolo en **creado\_id** e mediante este obxecto nodedemos

Este novo obxecto creado almacenámolo en `creado_id` e mediante este obxecto podemos actualizar os seus campos.

## Actualizar un rexistro existente

De novo engadimos no header da vista **pedido.xml** un botón que chamará neste caso ao método **actualizaRexistroInformacion**

E no modelo **pedido.py** engadimos o seguinte código:

```
def actualizaRexistroInformacion(self):
    informacion_id = self.env['odoo_basico.informacion'].search([('name', '=', '
    if informacion_id:
        informacion_id.name = "Actualizado ..."
        informacion_id.descripcion = "Actualizado dende o modelo pedido"
        informacion_id.sexo_traducido = "Mujer"
```

E mediante `self.env['odoo_basico.informacion'].search` seleccionamos o rexistro que queremos modificar.

E se existe facemos os cambios que queiramos nos campos dese rexistro.