

Importación de datos

Tipo de datos binary

Tipo de datos binary (imaxes,adxuntos)

Os atributos (name,descripcion,autorizado...) que definimos no modelo('odoo_basico.informacion') convértense en campos na táboa **odoo_informacion**.

```
name = fields.Char(string="Título",required=True,size=20)
descripcion = fields.Text(string="A Descrición")
autorizado = fields.Boolean(string="¿Está Autorizado?",default=True)
sexo_traducido = fields.Selection([('Hombre','Home'),('Mujer','Muller'),('Otros','Outros')])
alto_en_cms = fields.Integer(string="Alto en Cms.")
```

The screenshot shows the pgAdmin interface with a query executed on the 'public.odoo_basico_informacion' table. The query is: `SELECT * FROM public.odoo_basico_informacion ORDER BY id ASC`. The results are displayed in a table with the following columns: id (PK integer), name (character varying (20)), descripcion (text), autorizado (boolean), and sexo_traducido (character varying). The data rows are as follows:

id	name	descripcion	autorizado	sexo_traducido
1	Registro 1	Descrición Registro 1	true	Hombre
2	Registro 2	Descrición Registro 2	true	Hombre
3	Registro 3	Descrición Registro 3	true	Hombre
4	Registro 4	Descrición Registro 4	true	Mujer
5	Registro 5	Descrición Registro 5	false	Mujer
6	Registro 6	Descrición Registro 6	false	Mujer

Excepto os campos **binary** que se almacenan na táboa **ir_attachement**

```
foto = fields.Binary(string='Foto')
adxunto_nome = fields.Char(string="Nome Adxunto")
adxunto = fields.Binary(string="Archivo adxunto")
```

```
adxunto = Files.Binary(Setting= Archivo adxunto )
```

<ul style="list-style-type: none"> ir_act_server ir_act_server_group_rel ir_act_url ir_act_window ir_act_window_group_rel ir_act_window_view ir_actions ir_actions_todo ir_attachment Columns Constraints Indexes RLS Policies Rules Triggers ir_config_parameter ir_cron ir_default ir_demo ir_demo_failure ir_demo_failure_wizard ir_exports ir_exports_line ir_filters ir_logging ir_mail_server 	Data Output Explain Messages Notifications				
	id [PK] integer	name character varying	description text	res_model character varying	res_field character varying
253	253	web.assets_frontend.css	[null]	ir.ui.view	[null]
254	254	web.assets_common_minima...	[null]	ir.ui.view	[null]
255	255	web.assets_frontend_minima...	[null]	ir.ui.view	[null]
256	256	web.assets_common_lazyjs	[null]	ir.ui.view	[null]
257	257	web.assets_frontend_lazyjs	[null]	ir.ui.view	[null]
258	259	web.assets_common.js	[null]	ir.ui.view	[null]
259	261	web_editor.summernote.css	[null]	ir.ui.view	[null]
260	262	web_editor.summernote.js	[null]	ir.ui.view	[null]
261	263	web_editor.assets_wysiwyg.css	[null]	ir.ui.view	[null]
262	264	web_editor.assets_wysiwyg.js	[null]	ir.ui.view	[null]
263	267	foto	[null]	odoo_basico.informacion	foto
264	268	adxunto	[null]	odoo_basico.informacion	adxunto
265	269	foto	[null]	odoo_basico.informacion	foto
266	270	foto	[null]	odoo_basico.informacion	foto
267	271	foto	[null]	odoo_basico.informacion	foto
268	272	foto	[null]	odoo_basico.informacion	foto
269	273	foto	[null]	odoo_basico.informacion	foto
270	274	web.assets_backend.css	[null]	ir.ui.view	[null]
271	275	web.assets_backend.js	[null]	ir.ui.view	[null]

Na vista usamos o widget "image" para as fotos (por exemplo coa clase oe_avatar facemos a fot máis pequena)

No caso do arquivo adxunto definimos un campo tipo char (adxunto_nome) que é onde se almacena o nome do arquivo e que se relaciona na vista co campo adxunto mediante filename.

Vemos que o campo adxunto_nome ten que estar na vista pero con invisible="1" conseguimos que non se vexa.

```
<field name="foto" widget="image" class="oe_avatar"/>
<field name="foto" widget="image"/>
<field name="adxunto_nome" invisible="1"/>
<field name="adxunto" filename="adxunto_nome"/>
```

Obra publicada con [Licencia Creative Commons Reconocimiento Compartir igual 4.0](https://creativecommons.org/licenses/by/4.0/)