

Xestión de permisos para os usuarios

Explicación da xestión de permisos

Pasos a realizar

As definicións dos 3 primeiros pasos están no arquivo **xestion_usuarios.xml** e a do cuarto no arquivo **ir.model.access.csv** ambos están no cartafol **security**.

1. Definir a categoría para asignar os usuarios aos grupos
2. Definir os grupos onde se asignarán os usuarios
3. Definir os usuarios
4. Asignar os permisos mediante grupos de usuarios aos modelos do noso módulo

1. Creación da categoría para asignar a cada usuario o seu grupo de usuario.

The screenshot shows the Odoo user management interface. The browser address bar indicates the URL: `localhost:8069/web#id=2&action=68&model=res.users&view_type=form&menu_id=4`. The navigation bar includes 'Ajustes', 'Tablero', 'Usuarios y compañías', 'Traducciones', and 'Opciones Generales'. The breadcrumb trail is 'Tablero / Usuarios / Administrator'. Below the breadcrumb are 'Guardar' and 'Descartar' buttons. A message bar says 'Enviar instrucciones de restablecimiento de la contraseña'. The user profile section shows a placeholder for a user icon, the name 'Administrador', and the email 'odoo@odoo.com'. There are tabs for 'Permisos de acceso' and 'Preferencias'. The 'Permisos de aplicaciones' section is expanded, showing a list of permissions. The 'Xestión de odoo_basico.' permission is highlighted with a red box, and its 'Grupo:' field is highlighted with a green box, showing 'Grupo Lectura' and 'Grupo Escritura' as selected options. Other permissions listed include 'Contabilidad y Finanzas' and 'Administración'.

Creamos unha categoría no modelo **"ir.module.category"** co id externo **"Grupos_para_odoo_basico"**

```

<record model="ir.module.category" id="Grupos_para_odoo_basico"> <!-- id externo para id
    <field name="name">Xestión de odoo_basico. Grupo:</field><!-- Se visualiza na par
    <field name="description">Xestión de odoo basico</field>
    <field name="sequence">2</field>
</record>

```

Mediante o campo name poñemos o literal desexado no noso caso Xestión de odoo_basico Grupo:

2. Creación dos grupos para asignar os usuarios

Creamos no modelo **res.groups** dous grupos.

1. Mediante o id. externo **grupo_lectura** creamos un grupo para os usuarios que só terá permiso de lectura
2. Mediante o id. externo **grupo_escritura** creamos outro para os usuarios que terán todos os permisos .

```

<record model="res.groups" id="grupo_lectura"> <!-- id externo para identificar o rexistro
    <field name="name">Grupo Lectura</field> <!-- Nome para o grupo de usuarios que
    <field name="category_id" ref="Grupos_para_odoo_basico"/><!-- Se relaciona a través
    <field name="implied_ids" eval="[(4, ref('base.group_user'))]" /> <!-- many2many l
    <field name="users" eval="[(4, ref('base.user_root'))]" /> <!-- https://odoo-develop
</record>

```

```

<record model="res.groups" id="grupo_escritura"> <!-- id externo para identificar o
    <field name="name">Grupo Escritura</field> <!-- Nome para o grupo de usuarios que
    <field name="category_id" ref="Grupos_para_odoo_basico"/><!-- Se relaciona a través
    <field name="implied_ids" eval="[(4, ref('odoo_basico.grupo_lectura'))]" /> <!-- r
</record>

```

Explicación de **ref** e **eval** no seguinte link: <https://fundamentos-de-desarrollo-en-odoo.readthedocs.io/capitulos/serializacion-de-datos-y-datos-del-modulo.html>

- Mediante **ref** pasamos o **id externo** do rexistro que queiramos referenciar.
- Lista dos posibles valores de como traballará **eval** <https://odoo-development.readthedocs.io/en/latest/dev/py/x2many.html>

Mediante o campo **category_id** relacionamos o grupo que estamos creando co **id externo** d categoría (**Grupos_para_odoo_basico**) que xa creamos no apartado anterior

localhost:8069/web?id=2&action=68&model=res.users&view_type=form&menu_id=4

Ajustes Tablero Usuarios y compañías Traducciones Opciones Generales

Tablero / Usuarios / Administrator

Enviar instrucciones de restablecimiento de la contraseña

Nombre: Administrator

Dirección de Email: odoo@odoo.com

Permisos de acceso Preferencias

Permisos de aplicaciones

Xestión de odoo_basico. Grupo: Contabilidad y Finanzas Administración

Grupo Lectura

Grupo Escritura

Ajustes

Vemos no pgadmin os grupos:

Query Editor Query History

1 select * from res_groups order by name

Scratch Pad

id	name	comment	category_id	color	share	create_uid	create_date	write_uid	write_date
7	13	Analytic Accounting T...	[null]	11	[null]	1	2020-04-08 12:59:06.642978	1	2020-04-08 12:59:06.642978
8	19	Billing	[null]	40	[null]	1	2020-04-08 12:59:21.790714	1	2020-04-08 12:59:21.790714
9	23	Billing Manager	[null]	40	[null]	1	2020-04-08 12:59:21.790714	1	2020-04-08 12:59:21.790714
10	7	Contact Creation	[null]	59	[null]	1	2020-04-08 12:55:36.116836	1	2020-04-08 12:55:36.116836
11	39	Grupo Escritura	[null]	66	[null]	1	2020-05-24 16:27:35.024346	1	2020-05-24 16:27:35.024346
12	38	Grupo Lectura	[null]	66	[null]	1	2020-05-24 16:27:35.024346	1	2020-05-24 16:27:35.024346
13	1	Internal User	[null]	42	[null]	[null]	[null]	1	2020-04-08 12:59:21.790714
14	11	Manage Multiple Units ...	[null]	11	[null]	1	2020-04-08 12:58:56.925605	1	2020-04-08 12:58:56.925605

O id. externo base.group_user fai referencia ao **grupo "Internal User"**

Ajustes Tablero Usuarios y compañías Traducciones Opciones Generales Técnico

Identificadores externos / Tipos de Usuario / Usuario interno

Editar Crear

Acción

base.group_user

Módulo: base

Identificador externo: group_user

No actualizable: ☒

Fecha de actualización: 08/04/2020 14:59:21

Fecha inicial:

Nombre mostrado: Tipos de Usuario / Usuario interno

Nombre del modelo: res.groups

ID de registro: 1

Registro: Tipos de Usuario / Usuario interno

Mediante o campo **implied_ids**:

- Os usuarios do "Grupo Lectura" formarán parte do grupo "Internal User"
- name="implied_ids" eval="[(4, ref('base.group_user'))]

- Os usuarios do "Grupo Escritura" formarán parte do grupo "Grupo Lectura"
`name="implied_ids" eval="[(4, ref('odoo_basico.grupo_lectura'))]`

Vemos que `implied_ids` é unha relación many2many entre `res.groups` e `res.groups`:

The screenshot shows the Odoo Query Editor with a query: `SELECT * FROM public.res_groups_implied_rel`. The Data Output table shows columns `gid` and `hid`, both of type `integer`. The table contains 11 rows of data. Below the table, two pop-up windows show the field definitions for `res_groups_implied_rel_gid_fkey` and `res_groups_implied_rel_hid_fkey`. Both fields reference the `id` column of the `public.res_groups` table.

	gid	hid
1	3	2
2	1	6
3	19	1
4	23	19
5	3	23
6	10	20
7	9	20
8	1	20
9	1	27
10	38	1
11	39	38

Agora imos ver os **usuarios** implicados: o **1** (`__system__`) o **16** (usuario lectura) e o **17** (usuario escritura)

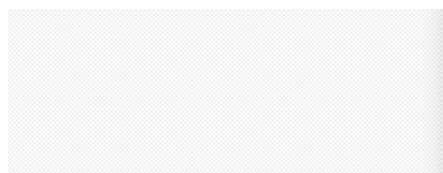
The screenshot shows the Odoo Query Editor with a query: `select * from res_users order by id`. The Data Output table shows columns `id`, `active`, `login`, `password`, `company_id`, and `partner_id`. The table contains 7 rows of data. Rows 1, 16, and 17 are highlighted with red boxes.

	id	active	login	password	company_id	partner_id
1	1	false	<code>__system__</code>	[null]	1	2
2	2	true	<code>odoo@odoo.com</code>	<code>\$pbkdf2-sha512\$2500...</code>	1	3
3	3	false	<code>default</code>	[null]	1	5
4	4	false	<code>public</code>	<code>\$pbkdf2-sha512\$2500...</code>	1	4
5	5	false	<code>portaltemplate</code>	[null]	1	6
6	16	true	<code>lectura@odoo.com</code>	<code>\$pbkdf2-sha512\$2500...</code>	1	23
7	17	true	<code>escritura@odoo.com</code>	<code>\$pbkdf2-sha512\$2500...</code>	1	24

O `id.` externo `base.user_root` fai referencia ao usuario con `id=1`, como vemos:

The screenshot shows the Odoo configuration page for "Identificadores externos / OdooBot". The "base.user_root" field is highlighted with a red box. The table below shows the configuration for the "base" module.

Módulo	Identificador externo	Nombre mostrado	Nombre del modelo
base	<code>user_root</code>	OdooBot	<code>res.users</code>



No actualizable
Fecha de
actualización
Fecha inicial

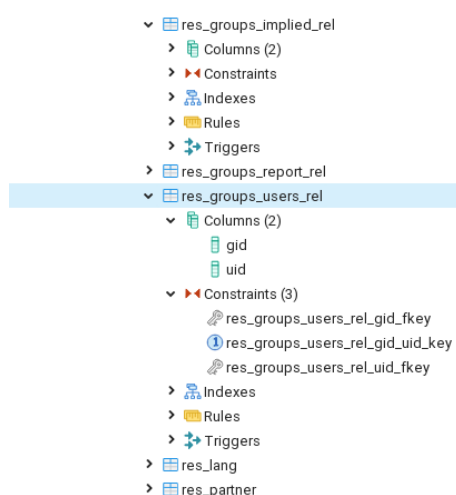
29/05/2020 14:37:15

ID de registro
Registro

1
OdooBot

Vemos como engadimos:

- O usuario do sistema (sempre id=1) **__system__** (mediante o seu id. extern **base.user_root**) no "Grupo Lectura"(38)
- ```
<field name="users" eval="[(4, ref('base.user_root')))]"/>
```
- Despois veremos como engadimos os usuarios en res.users ( `model="res.users` `id="usuario_lectura"`), nese momento o usuario lectura (no meu caso id=16) engadiremol no "Grupo Lectura" (38) mediante
- ```
<field name="groups_id" eval="[(6, 0, [ref('odoo_basico.grupo_lectura')]))]"/>
```
- Despois veremos como engadimos os usuarios en res.users (`model="res.users` `id="usuario_escritura"`), nese momento o usuario escritura (no meu caso id=17) engadiremolo no "Grupo Escritura" (39) mediante
- ```
<field name="groups_id" eval="[(6, 0, [ref('odoo_basico.grupo_escritura')]))]"/>
```



documentacion/odoo@127.0.0.1

Query Editor Query History

1 `select * from res_groups_users_rel where gid=1 or gid=38 or gid=39 order by gid`

	gid	uid
1	integer	integer
2	1	3
3	1	2
4	1	1
5	1	16
6	1	17
7	38	16
8	38	1
9	38	17

```

> res_partner_autocomplete_sync
> res_partner_bank
> res_partner_category

```

- O usuario escritura (no meu caso id=17) engadese no "Grupo Lectura" (38). Como vimo anteriormente ("Os usuarios do "Grupo Escritura" formarían parte do grupo "Grupo Lectura" `name="implied_ids" eval="[(4, ref('odoo_basico.grupo_lectura'))]"` ")
- Vemos tamén que os usuarios 16 e 17 están no grupo 1 porque como vimo anteriormente ("Os usuarios do "Grupo Lectura" formarían parte do grupo "Internal User" `name="implied_ids" eval="[(4, ref('base.group_user'))]"` ")

<http://www.odoo.yenthevg.com/creating-security-groups-odoo/>

Poderíanse aplicar os grupos de usuarios as opcións do menú.

### 3. Creación dos usuarios.

Creamos dous usuarios no modelo **res.users**

1. Con id. externo **usuario\_lectura** que estará no grupo\_lectura mediante a asignación n campo **groups\_id**
2. Con id. externo **usuario\_escritura** que estará no grupo\_escritura mediante a asignación no campo **groups\_id**

Asignamos tamén o timezone, idioma, login e a password en claro entre outros

```

<record model="res.users" id="usuario_lectura">
 <field name="name">Nome Lectura</field>
 <field name="visible">true</field>
 <field name="display_name">Nome Lectura Apellidos Lectura</field>
 <field name="email">Nome_Lectura@iesteis.com</field>
 <field name="active">true</field>
 <field name="company_id">1</field>
 <field name="lang">es_ES</field>
 <field name="tz">Europe/Madrid</field>
 <field name="login">lectura@odoo.com</field>
 <field name="password">odoo</field>
 <field name="groups_id" eval="[(6, 0, [ref('odoo_basico.grupo_lectura')])]" />
</record>
<!-- Se da erro por violar a constraint notification_type con valor null. -->
<!-- Temos que desinstalar e instalar o módulo para que tome por defecto o valor email -->

```

```

<record model="res.users" id="usuario_escritura">
 <field name="name">Nome Escritura</field>
 <field name="visible">true</field>
 <field name="display_name">Nome Escritura Apellidos Escritura</field>
 <field name="email">Nome_escritura@iesteis.com</field>
 <field name="active">true</field>
 <field name="company_id">1</field>
 <field name="lang">es_ES</field>
 <field name="tz">Europe/Madrid</field>
 <field name="login">escritura@odoo.com</field>
 <field name="password">odoo</field>
 <field name="groups_id" eval="[(6, 0, [ref('odoo_basico.grupo_escritura')])]" />
</record>

```

## 4. Asignar os permisos mediante grupos de usuarios aos modelos do noso módulo

Como vemos a primeira liña son as cabeceiras onde podemos ver os campos que estamos tratando o caracter separador é a coma (,)

```

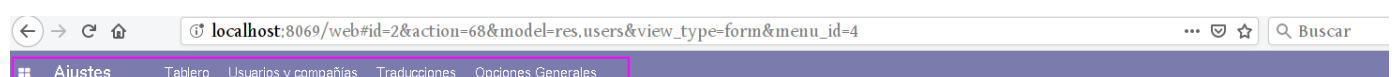
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
odoo_basico_informacion_lectura,odoo_basico.informacion.lectura,model_odoo_basico_informacion_lectura,odoo_basico.informacion.lectura,model_odoo_basico_informacion_lectura,odoo_basico.informacion.lectura,model_odoo_basico_informacion_lectura,odoo_basico.informacion.lectura,model_odoo_basico_informacion_lectura
odoo_basico_informacion_escritura,odoo_basico.informacion.escritura,model_odoo_basico_informacion_escritura,odoo_basico.informacion.escritura,model_odoo_basico_informacion_escritura,odoo_basico.informacion.escritura,model_odoo_basico_informacion_escritura,odoo_basico.informacion.escritura,model_odoo_basico_informacion_escritura

```

Vemos que para o noso modelo **odoo\_basico.informacion** xestionamos dous tipos de permisos:

1. Para o grupo de usuarios **odoo\_basico.grupo\_lectura** que terá perm\_read=perm\_write=0 perm\_create=0 e perm\_unlink(eliminar)=0. Co cal os usuarios asignados a **grupo\_lectura** só poderán visualizar.
2. Para o grupo de usuarios **odoo\_basico.grupo\_escritura** que terá perm\_read=perm\_write=1 perm\_create=1 e perm\_unlink(eliminar)=1. Co cal os usuarios asignados a **grupo\_escritura** terán todos os permisos.


Vemos na seguinte captura como configuramos o usuario Administrador a un grupo(lectura o escritura) para que poda acceder ao modelo de odoo\_basico.informacion.



Tablero / Usuarios / Administrator

Guardar Descartar

Enviar instrucciones de restablecimiento de la contraseña



Nombre

Administrator

Dirección de Email

odoo@odoo.com

✓ Activo

Permisos de acceso Preferencias

**Permisos de aplicaciones**

Xestión de odoo\_basico.  
Grupo:

Contabilidad y Finanzas

Administración

Grupo Lectura

Grupo Escritura

Ajustes

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