Accións Planificadas

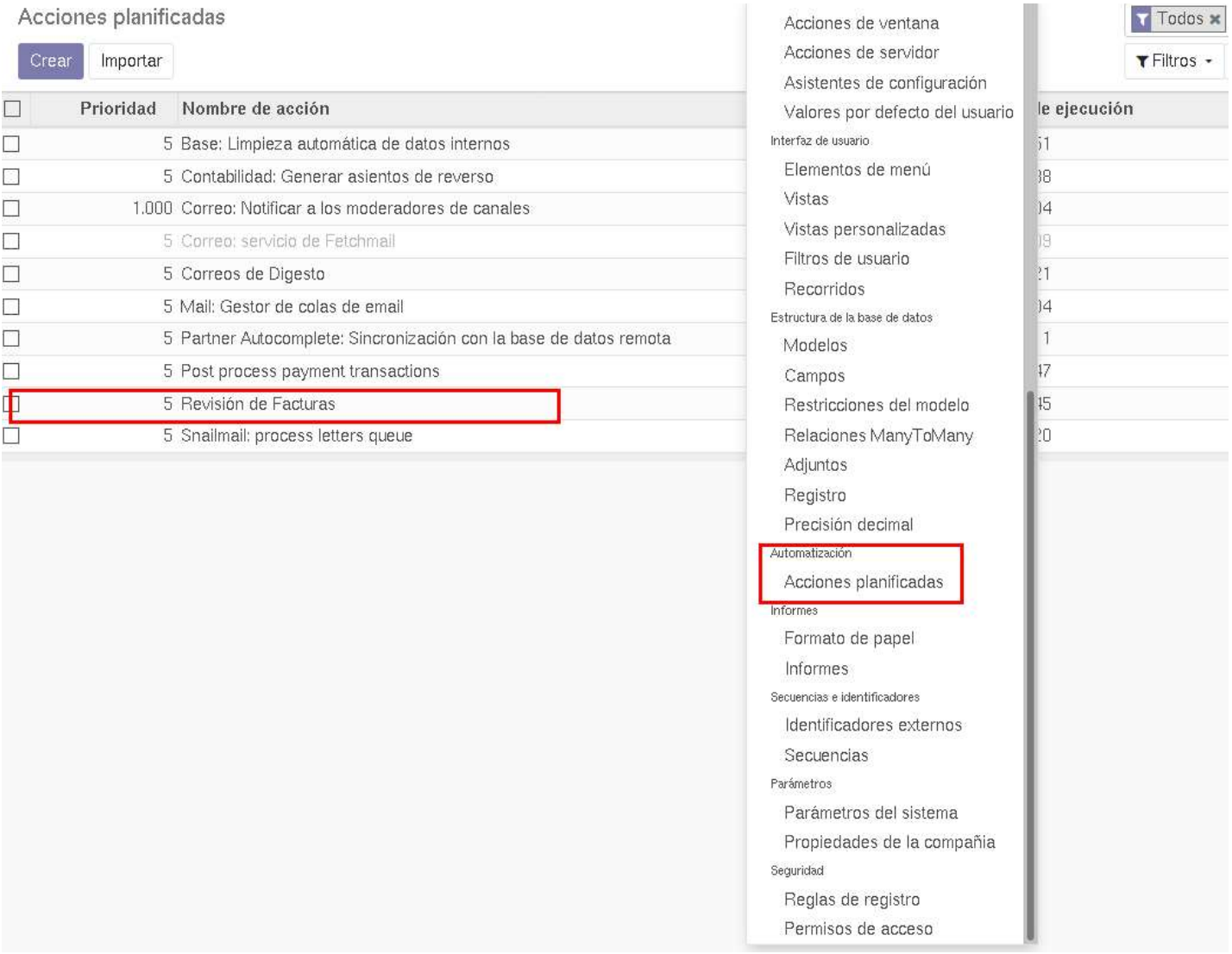
Accións Planificadas

Podemos executar accións planificadas dende Odoo cada certo tempo. Para elo temos que crear un rexistro na táboa **ir_cron**.

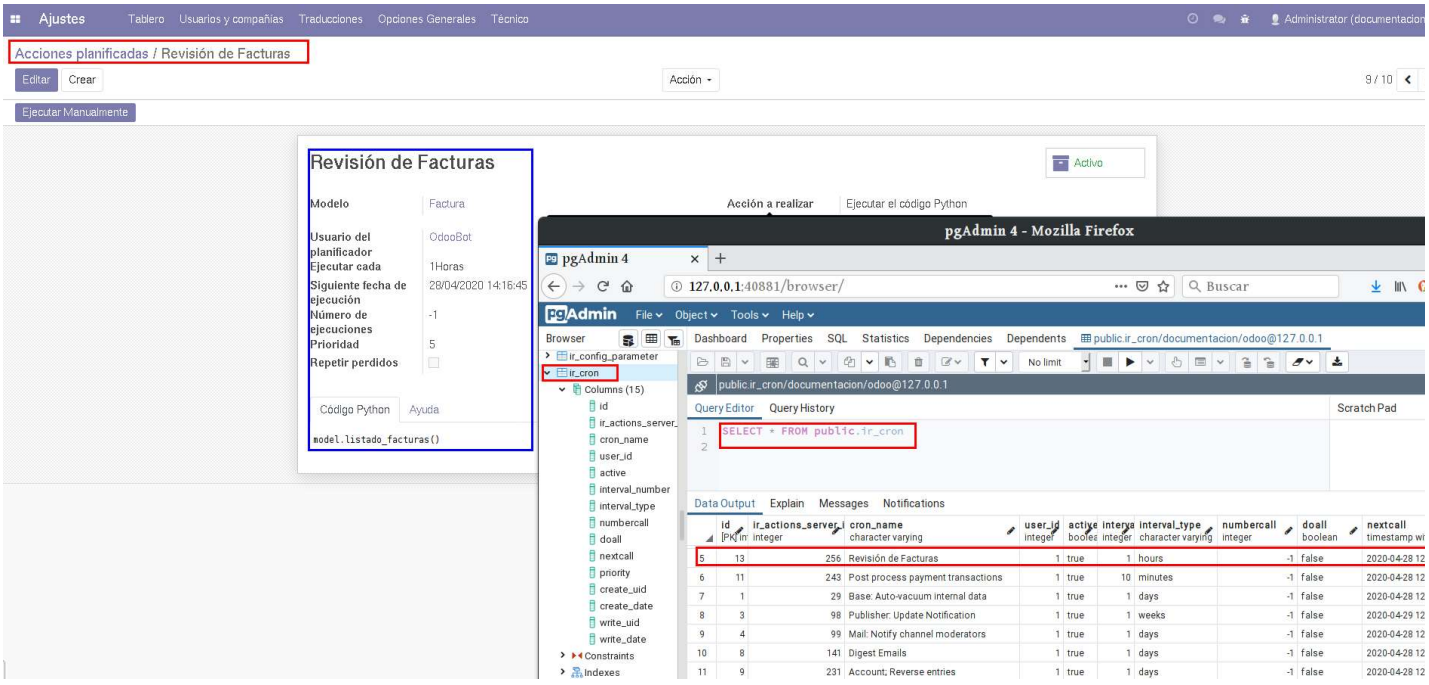
Podemos ver a definición dese rexistro no arquivo **accion_planificada.xml** no cartafo **accions_planificadas**. Dito arquivo ten que estar referenciado no apartado 'data' do arquiv "__manifest__.py"

```
<odoo>
    <data>
        <!--    https://odoo-development.readthedocs.io/en/latest/odoo/models/ir.c
        <record  model="ir.cron"  id="envio_facturas">
            <field name="name">Revisión de Facturas</field>
            <field name="active" eval="True"  />
            <field name="user_id" ref="base.user_root" />
            <field name="interval_number">1</field>
            <field name="interval_type">hours</field>
            <field name="numbercall">-1</field> <!--  An integer value specifying h
            <!--<field name="function">True</field>Odoo Version 12 -->
            <!--<field name="model_id" ref="model_account_invoice" />  Odoo Versio
            <field name="model_id" ref="model_account_move" />  <!-- Odoo Version 1
            <field name="state">code</field>
            <field name="code">model.listado_facturas()</field>
            <field name="doall" eval="False" /> <!--      A boolean value indicati
        </record>
    </data>
</odoo>
```

Despois de actualizar o noso módulo podemos ver como foi dado de alta:



Se vemos a acción planificada, podemos ver que temos un botón de "Ejecutar Manualmente"(qu podemos utilizar para facer as probas):

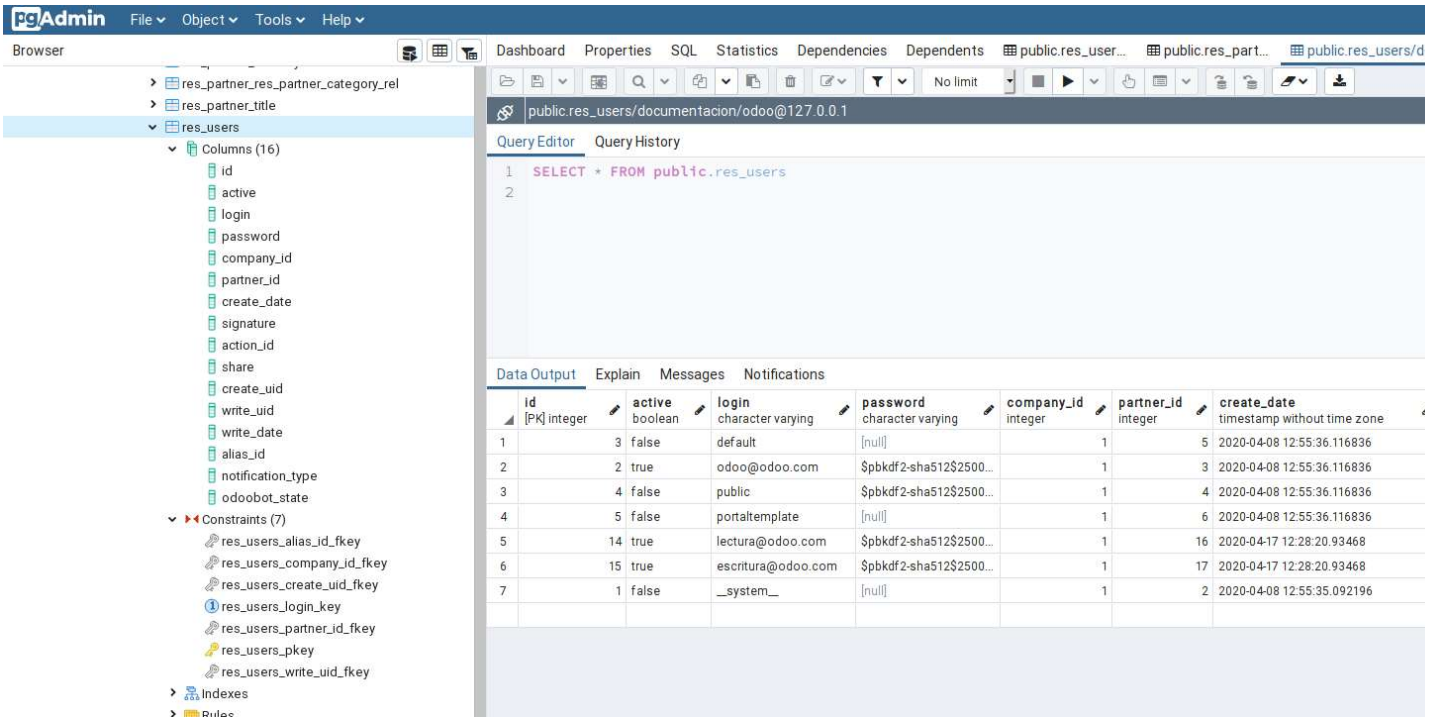


Cabe destacar nos campos que estamos definindo:

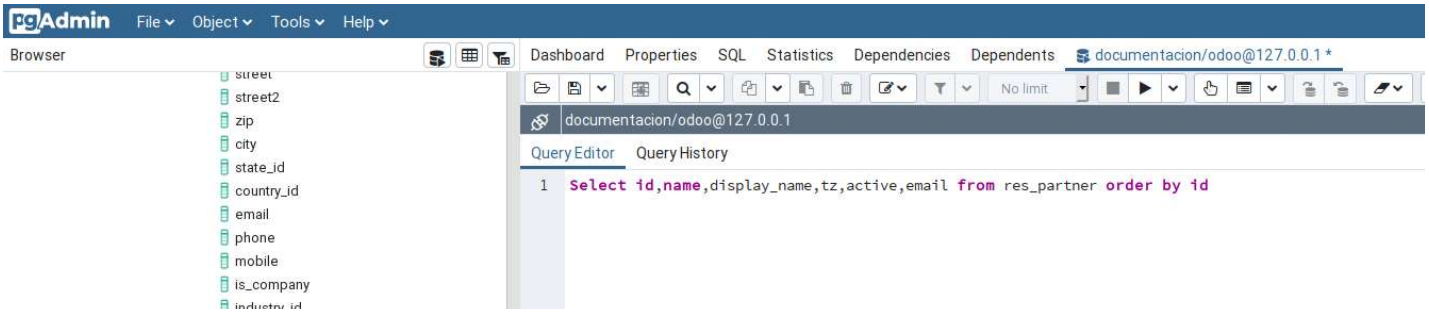
- **O usuario** que executará a acción planificada <field name="user_id" ref="base.user_root" /> definido a través do id externo(base.user_root).

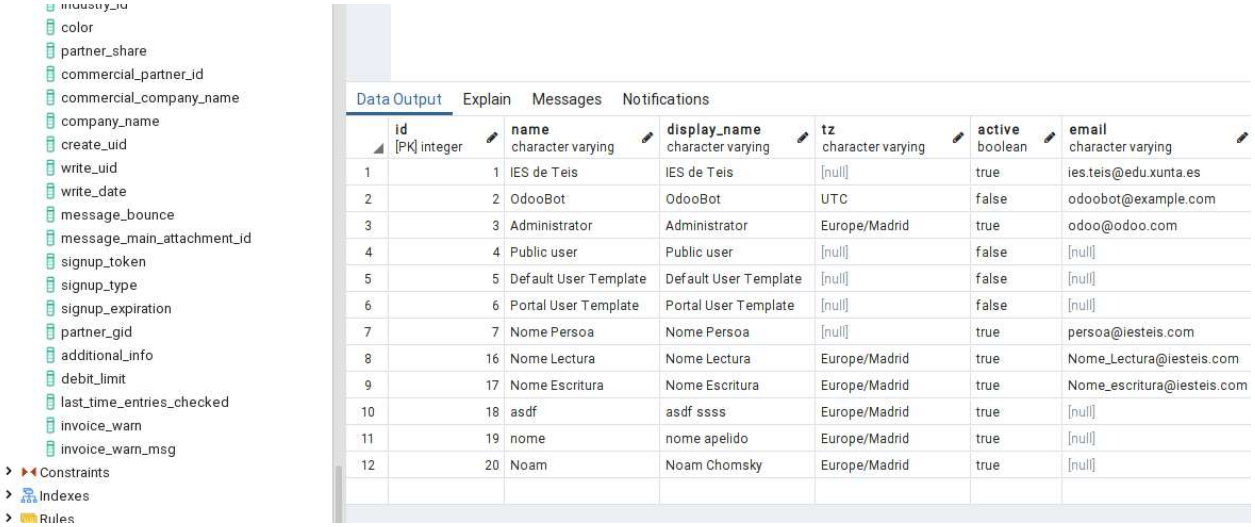
O cal nos permite saber:

Que é o usuario con id=1 da táboa res_users.



Vemos que é o usuario **__system__** que está relacionado coa táboa res_partner mediant partner_id = 2. Recordemos que res_users e res_partners teñen unha relación de Herdanza pc Delegación.





	id [PK] integer	name character varying	display_name character varying	tz character varying	active boolean	email character varying
1	1	IES de Teis	IES de Teis	[null]	true	ies.teis@edu.xunta.es
2	2	OdooBot	OdooBot	UTC	false	odoobot@example.com
3	3	Administrator	Administrator	Europe/Madrid	true	odoo@odoo.com
4	4	Public user	Public user	[null]	false	[null]
5	5	Default User Template	Default User Template	[null]	false	[null]
6	6	Portal User Template	Portal User Template	[null]	false	[null]
7	7	Nome Persoa	Nome Persoa	[null]	true	persoa@iesteis.com
8	16	Nome Lectura	Nome Lectura	Europe/Madrid	true	Nome_Lectura@iesteis.com
9	17	Nome Escritura	Nome Escritura	Europe/Madrid	true	Nome_escritura@iesteis.com
10	18	asdf	asdf ssss	Europe/Madrid	true	[null]
11	19	nome	nome apelido	Europe/Madrid	true	[null]
12	20	Noam	Noam Chomsky	Europe/Madrid	true	[null]

Agora podemos ver que o rexistro de res_partner con id=**2**, ten como name=**OdooBot**, com tz=**UTC** e como email=**odoobot@example.com**

- **O intervalo e tipo**, no noso caso cada "1 hora". <field name="interval_number">1</field <field name="interval_type">hours</field>
- **O modelo**, no noso caso:
  - Versión 12 de Odoo **account_invoice** (usaremos a herdanza). Identificase tamén través do id externo. <field name="model_id" ref="model_account_invoice" />
  - Versión 13 de Odoo **account_move** (usaremos a herdanza). Identificase tamén través do id externo. <field name="model_id" ref="model_account_move" />
- **O método a executar** que será **listado_facturas** <fiel name="code">model.listado_facturas()</field>

Vemos a definición do método **listado_facturas** que temos definid no arquivo **accion_planificada.py**

```
class accion_planificada (models.Model):
    #_inherit = "account.invoice" # Odoo Version 12
    _inherit = "account.move" # Odoo Version 13

    @api.model
    def listado_facturas(self):
        usuario_que_executa_o_metodo_que_e_o_definido_no_xml = self.env.user  # usu
        self.env.user.tz = self.env['res.partner'].search([('id', '=', 3)])[0].tz
        agora = self.env['odoo_basico.informacion'].convirte_data_hora_de_utc_a_tir
        self.env.user.tz = 'UTC' # deixamolo como estaba con tz=UTC
        #facturas_ids = self.search([('state', '=', 'open')]) Version Odoo 12
        facturas_ids = self.search([('state', '=', 'posted')])
```

```
        if facturas_ids:
            listado = ""
            for rexistro in facturas_ids:
                #listado = listado + "<br/>" + str(rexistro.number) + "-> " + str(r
                listado = listado + "<br/>" + str(rexistro.name) + "-> " + str(rex
            # mail_de     Odoo pon o email que configuramos en gmail para facer o e
            mail_reply_to = usuario_que_executa_o_metodo_que_e_o_definido_no_xml.pa
            mail_para = self.env['res.partner'].search([('id', '=', 3)])[0].email
            mail_valores = {
                    'subject': "Listaxe de facturas neste momento %s" % agora,
                    'author_id': usuario_que_executa_o_metodo_que_e_o_definido_no_x
                    'email_from': mail_reply_to,
                    'email_to': mail_para,
                    'message_type': 'email',
                    'body_html': "Neste momento %s existen as seguintes facturas: %
                }
            mail_id = self.env['mail.mail'].create(mail_valores)
            mail_id.send()
```

Vemos que herdamos por Extensión o modelo **account.move** (na versión 13 de Odoo) e nest caso engadimos un novo método "**listado_facturas**" para xerar e enviar un email.

O primeiro é determinar a data e hora actual para engadir no email. Mediant fields.Datetime.now() temos o datetime. Pero en hora UTC e queremos convertila ao timezone d usuario administrador.

Isto faremolo chamando ao método "convirte_data_hora_de_utc_a_timezone_do_usuario" qu temos definido no módulo información.

```
agora = self.env['odoo_basico.informacion'].convirte_data_hora_de_utc_a
```

Neste momento temos un problema xa que se recordamos o código que temos no módul "informacion" :

```
    def convirte_data_hora_de_utc_a_timezone_do_usuario(self,data_hora_utc_object)
        usuario_timezone = pytz.timezone(self.env.user.tz or 'UTC')  # obter a zona
        return pytz.UTC.localize(data_hora_utc_object).astimezone(usuario_timezone)
        # para usar  pytz temos que facer  import pytz
```

Podemos ver que para determinar o tz usa o obxeto 'self' (self.env.user.tz) e como xa sabemo python pasa automáticamente 'self' dende onde fixemos a chamada. No noso caso self.env.user far

referencia ao usuario que está executando a accion_planificada que non é o usuario administrado senon que é como xa vimos na definición de **accion_planificada.xml** o usuario **__system__** qu ten como tz=**UTC**.

Por tanto, o que facemos e modificar temporalmente o valor de self.env.user.tz e **accion_planificada.py** para que cando chamemos "convirte_data_hora_de_utc_a_timezone_do_usuario" teña o valor do usuario administrador. Is conseguimolo mediante:

```
self.env.user.tz = self.env['res.partner'].search([('id', '=', 3)])[0].tz  # como
```

Despois de facer a conversión volvemos deixar o obxeto self co seu valor orixinal:

```
self.env.user.tz = 'UTC' # deixamolo como estaba con tz=UTC
```

A continuación seleccionamos todas as facturas que teñen state= posted. Como estamos e **accion_planificada.py** e este modelo herda de **account.move** (na versión 13 de Odoo) o obxet 'self' fai referencia aos valores das facturas.

```
        #facturas_ids = self.search([('state', '=', 'open')]) Version Odoo 12
        facturas_ids = self.search([('state', '=', 'posted')])
```

Se existe algunha factura inicializamos unha variable de texto (**listado**) para ir engadindo mensaxe que queremos enviar no body do email.

Mediante un bucle imonos movendo por todas as facturas e acumulando na variable de text **listado** :

- Unha factura por liña, mediante a etiqueta HTML **br**
- Por exemplo seleccionamos para cada factura: Número de factura, Nome do cliente Importe total da factura
- Como vimos no apartado de "Envio de Email" xogamos con 3 enderezos de email.
  - From: será a conta de gmail que temos configurado en Odoo. Non temos que indicalo xa o fai Odoo.
  - Reply_to: No noso caso será o email do usuario que executa a acción planificad

(__system__):

```
mail_reply_to = usuario_que_executa_o_metodo_que_e_o_definido_no_xml
```

- To: O email do usuario administrador. (Se nos fixamos nas capturas anteriores figur odoo@odoo.com, pero modifiquei a configuración do email do usuario administrador unha conta miña onde poder recibir os emails)

- E finalmente como xa vimos no apartado de "Envio de Email" cargamos os valore necesarios para o envio do email na variable mail_valores, creamos o email e enviamol Neste caso non temos que utilizar sudo porque o usuario que envia o email é **__system__** ten os permisos necesarios para facelo. Caso distinto de cando enviamos o email dende módulo informacion xa que nese caso o usuario era o administrador e necesitamos utiliza **sudo** para que tivese os permisos necesarios.

Vemos o email recibido: