



# The Round Table

## Indice

1. Resumen.....	3
2. Estructura del proyecto.....	4
3. Contenido de cada fichero.....	4
4. Implementación de dos modelos.....	5
5. Creación de las vistas, modelos y menús dentro de Odoo.....	5
6. Modificación de las vistas.....	7
7. Restricciones en los campos.....	8
8. pgAdmin.....	9

## Figuras

Figure 1: Estructura del módulo.....	4
Figure 2: Plugin Tree View.....	5
Figure 3: License Tree View.....	6
Figure 4: User Tree View.....	6
Figure 5: Plugin From View.....	6
Figure 6: New Plugin Form View.....	7
Figure 7: Plugin Form View XML.....	7
Figure 8: Plugin price constraint.....	8
Figure 9: Plugin price validation demo.....	8
Figure 10: Plugin monthly dependency on price.....	8
Figure 11: Plugin monthly dependency on price demo.....	8
Figure 12: Tablas creadas en la base de datos.....	9
Figure 13: Contenido de la tabla trt_plugin.....	9

# 1. Resumen

The Round Table es el software principal de LebaStudios, Software Solutions. Como su software principal, este requiere una gran cantidad de inversión en recursos humanos, técnicos y económicos. Con el objetivo de facilitar la gestión de este gran proyecto se ha planteado crear una base de módulo para Odoo que define las características básicas que se deben controlar y monitorear sobre este producto:

- Licencias
- Usuarios
- Plugins

Los tres ‘modelos’ mencionados son los elementos fundamentales del ciclo económico del software y, por ello, serán la base sobre la que comenzaremos a desarrollar nuestro módulo. Se definirán las relaciones pertinentes y las propiedades oportunas, se diseñarán las vistas básicas de la interfaz y por último, se programará las restricciones necesarias a cada modelo.

## 2. Estructura del proyecto

```
borja@6V62-7RD:~/Documents/dam-2/sxe/ud4/trt$ tree -I odoo
.
├── __init__.py
├── __manifest__.py
├── models
│   ├── __init__.py
│   ├── license.py
│   ├── plugin.py
│   └── user.py
├── presentacion-trabajo.odt
├── security
│   ├── ir.model.access.csv
│   └── trt_security.xml
└── views
    ├── license.xml
    ├── plugin.xml
    ├── root.xml
    └── user.xml
```

Figure 1: Estructura del módulo

## 3. Contenido de cada fichero

- `models/license.py`  
Contiene el modelo que representa las posibles licencias que cada usuario posee. Actualmente se contemplan tres tipos: Standard, Lifetime y Educational.
- `models/plugin.py`:  
Contiene el modelo que representa todos los plugins que estan disponibles para descarga en el servidor central. Entre los campos se encuentra el nombrdel plugin, costo y repositorio de git.
- `models/user.py`:  
Contiene el modelo que representa los usuarios actuales del software que tengan cuenta registrada. Esta cuenta registrada no es necesaria para las funcionalidades gratuitas pero si para las de pago como plugins no gratuitos o servicios de soporte.
- `views/license.xml`:  
Define las vistas del modelo que representa las licencias. (Tree, Search, Form)
- `views/plugin.xml`:  
Define las vistas del modelo que representa los plugins. (Tree, Search, Form)
- `views/root.xml`:

Define los elementos de interfaz elementales del modulo, por ejemplo, el menu padre del que heredan el resto de menus de los modelos.

- `views/user.xml`:  
Define las vistas del modelo que representa los usuarios. (Tree, Search, Form)

## 4. Implementación de dos modelos

Actualmente existen los tres modelos básicos que representan la información relacionada a los usuarios y su acceso a los servicios de The Round Table a través de licencias.

Una Licencia podrá tener asociada de 0 a más plugins y cada usuario podrá poseer de 1 a mas licencias. Nótese que como mínimo, cada usuario tendrá una licencia Standard con acceso básico a las funcionalidades gratuitas.

Cada licencia solo puede pertenecer a un usuario y son intransferibles.

## 5. Creación de las vistas, modelos y menús dentro de Odoo

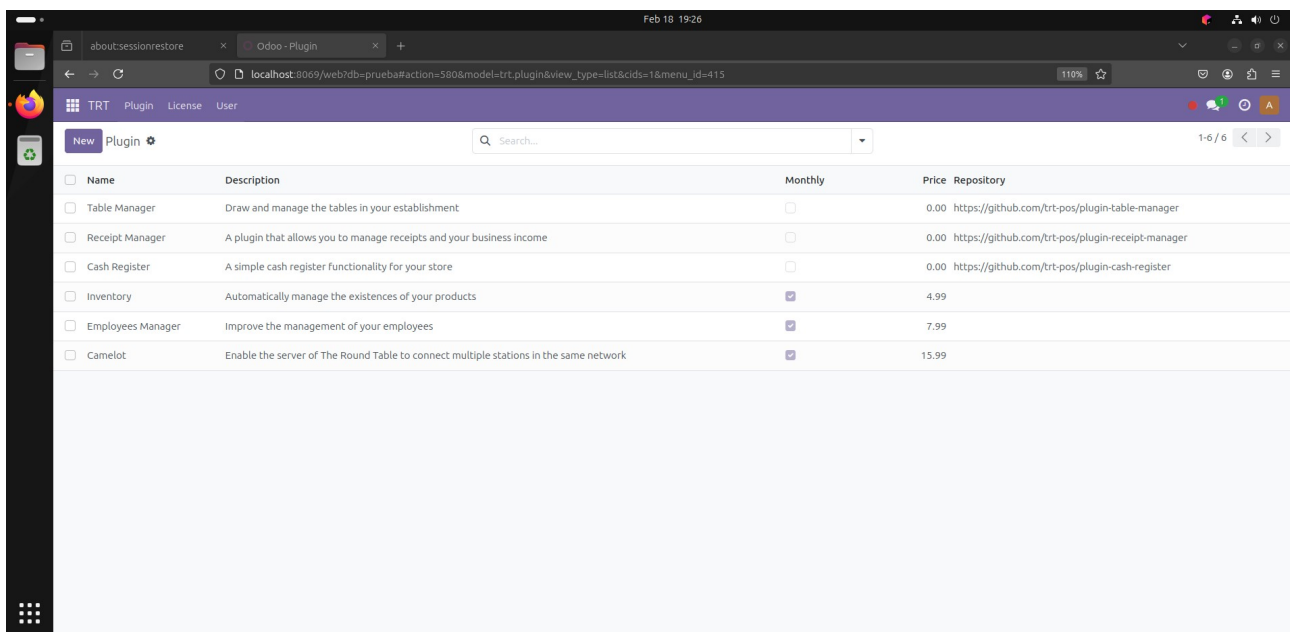


Figure 2: Plugin Tree View

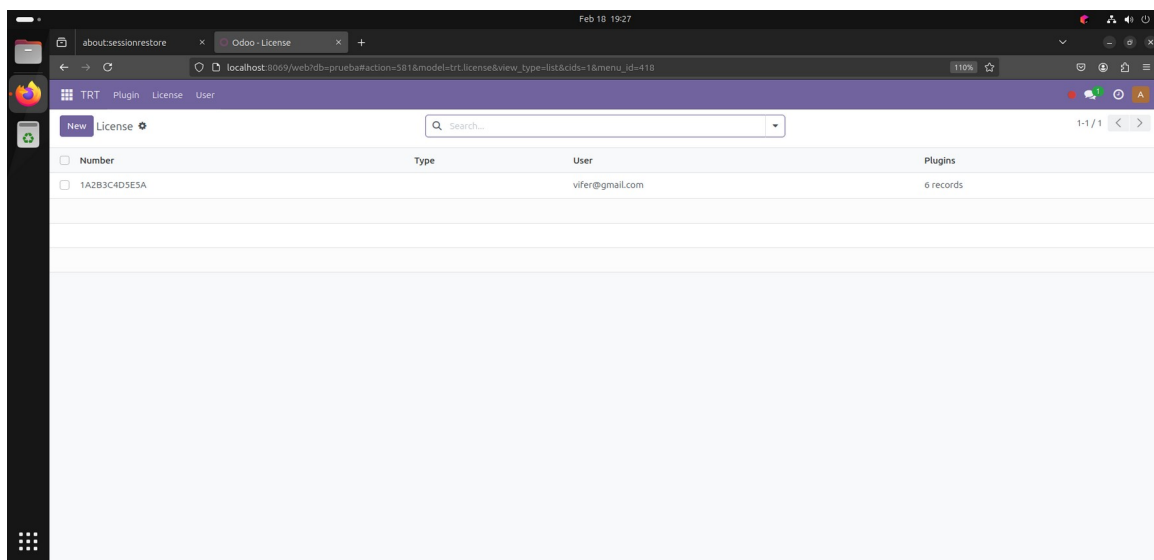


Figure 3: License Tree View

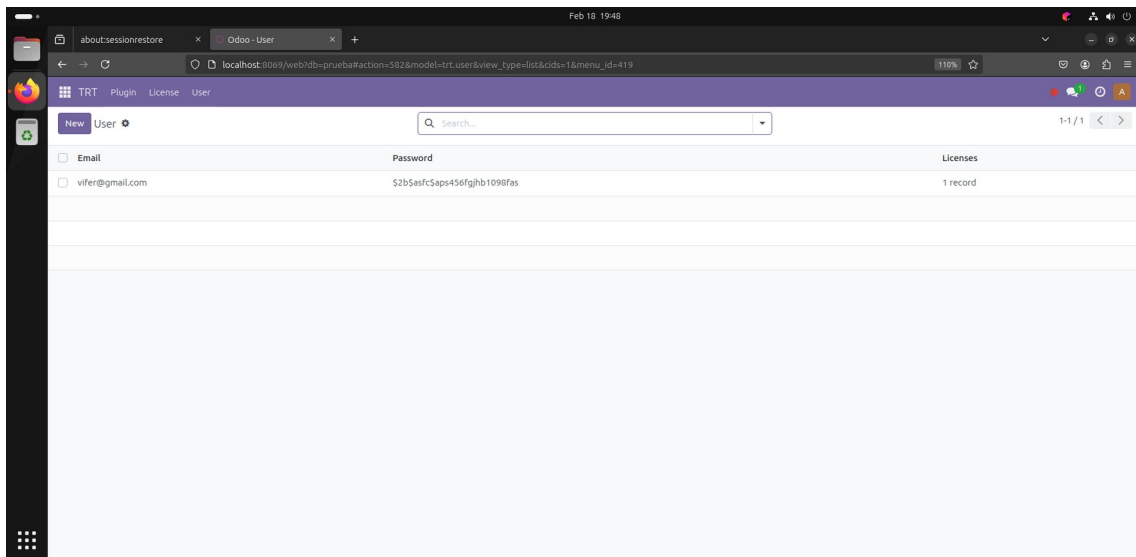


Figure 4: User Tree View

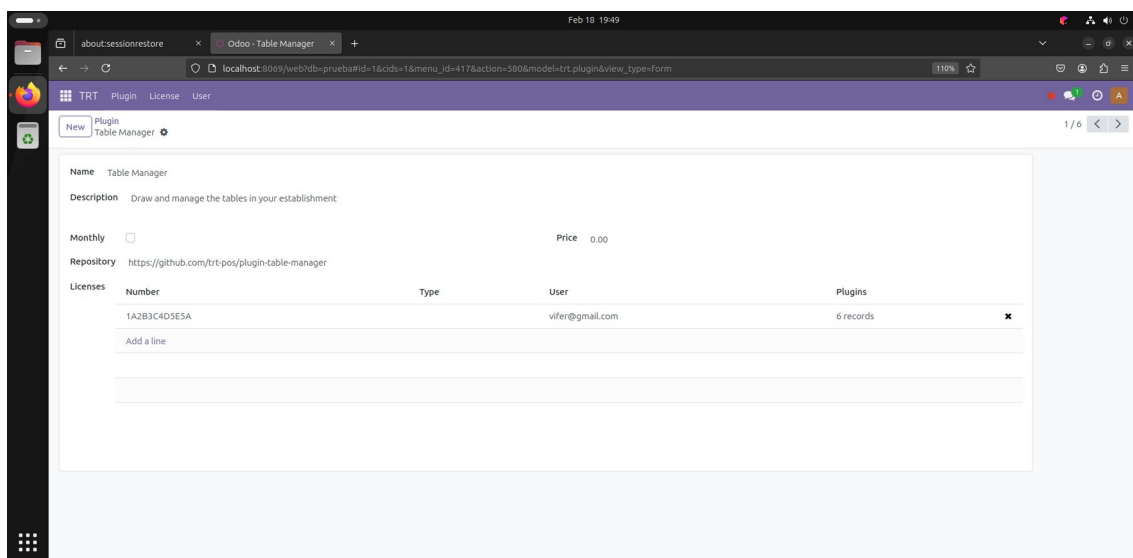


Figure 5: Plugin From View

## 6. Modificación de las vistas

Virta de formulario modificada:

The screenshot shows a web browser window with the address bar displaying 'localhost:8069/web#id=1&cids=1&menu\_id=72&action=88&model=trt.plugin&view\_type=form'. The browser has several tabs open: 'New Tab', 'Paquete de instaladores', and 'Odoo - Cash Register'. The Odoo interface shows a 'New' button and a 'Plugin Cash Register' header. Below this, there are two tabs: 'General' and 'Pricing'. The 'General' tab is selected, showing a form with three fields: 'Name' (Cash Register), 'Description' (Caja registradora por defecto), and 'Repository' (https://github.com/trt-pos/plugin-cash-register).

Figure 6: New Plugin Form View

XML de la vista modificada:

```
<record id="view_plugin_form_view" model="ir.ui.view">
  <field name="name">trt.plugin.form.view</field>
  <field name="model">trt.plugin</field>
  <field name="arch" type="xml">
    <form>
      <sheet>
        <notebook>
          <page string="General">
            <group>
              <field name="name"/>
              <field name="description" widget="text"/>
              <field name="repository" widget="url"/>
            </group>
          </page>
          <page string="Pricing">
            <group>
              <field name="is_monthly" widget="boolean_toggle"/>
              <field name="price" widget="monetary"/>
            </group>
          </page>
        </notebook>
      </sheet>
    </form>
  </field>
</record>
```

Figure 7: Plugin Form View XML

## 7. Restricciones en los campos

En cuanto a restricciones, se ha establecido que el precio de un plugin no pueda ser negativo:

```
@api.constrains('price')  ⬆ Borja Castellano
def _check_min_price(self):
    for record in self:
        if record.price < 0:
            raise ValidationError("The price must be equal or greater than 0")
```

Figure 8: Plugin price constraint

En esta imagen se puede apreciar la restricción en funcionamiento:

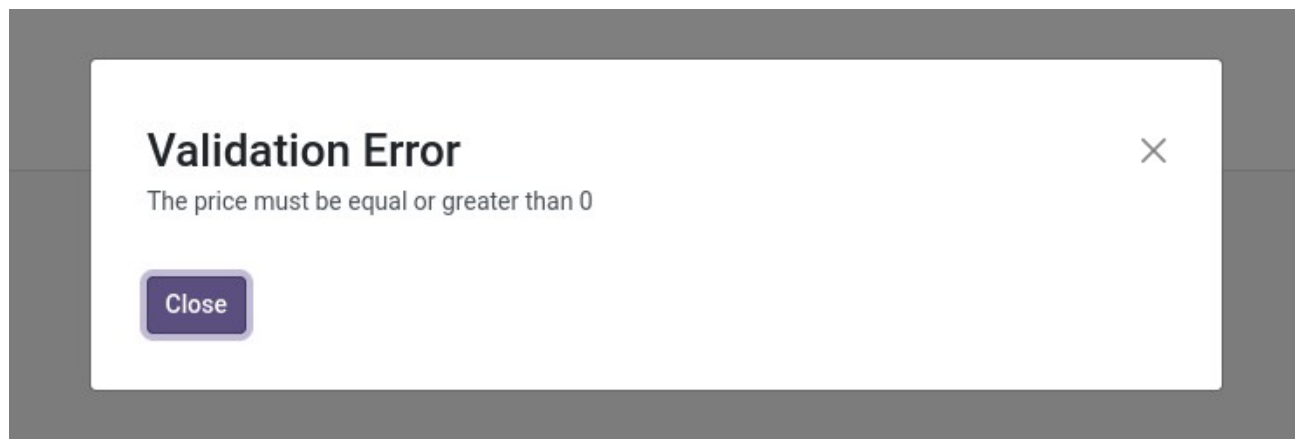


Figure 9: Plugin price validation demo

Junto a esto, también se ha decidido definir los plugins como productos de suscripción (monthly) siempre que tengan definido un precio:

```
@api.depends('price')  1 usage  ⬆ Borja Castellano
def _compute_is_monthly(self):
    for record in self:
        record.is_monthly = record.price > 0
```

Figure 10: Plugin monthly dependency on price

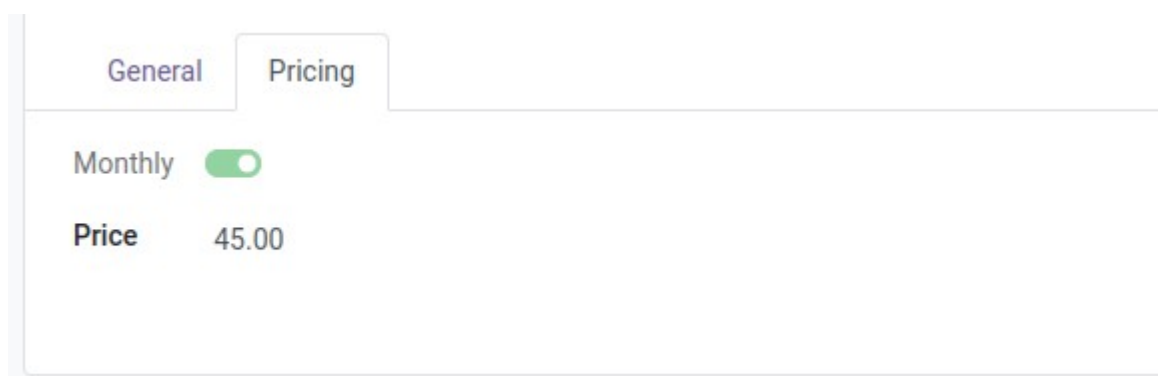


Figure 11: Plugin monthly dependency on price demo



## 8. pgAdmin

Tablas creadas por Odoo:

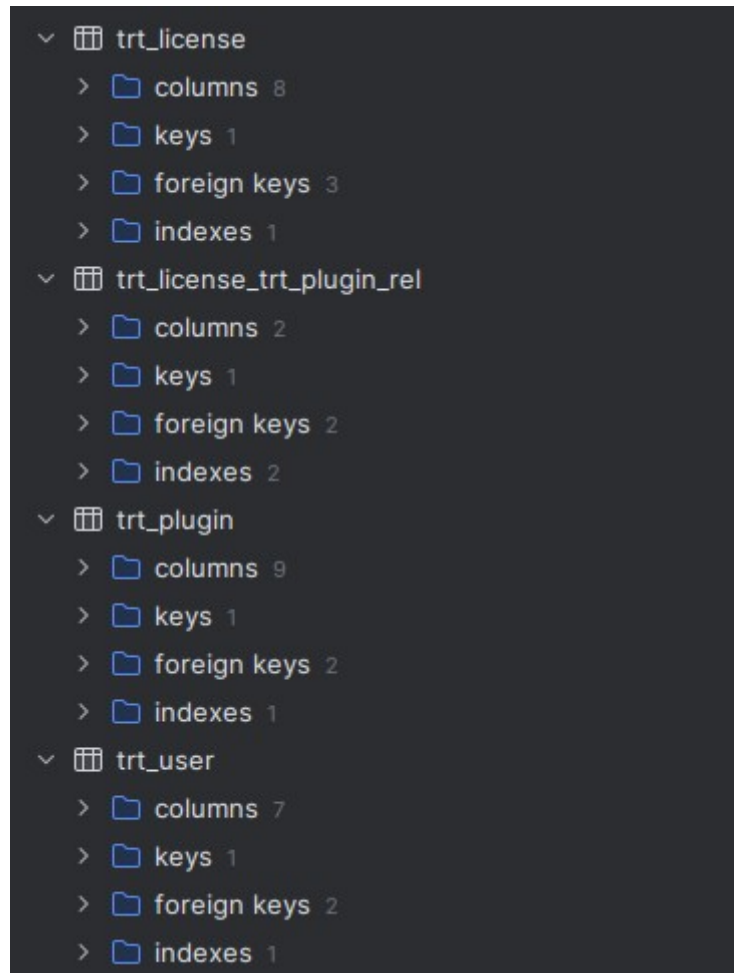


Figure 12: Tablas creadas en la base de datos

Vista del contenido de la tabla trt\_plugin:

	id	create_uid	write_uid	name	repository	description	create_date	write_date	price
1	1	2	2	Cash Register	<a href="https://github.com/tr...">https://github.com/tr...</a>	Caja registradora por...	2025-02-25 17:16:53.8...	2025-02-25 17:26:53.2...	0
2	2	2	2	Table Manager	<a href="https://github.com/tr...">https://github.com/tr...</a>	Manage the tables in ...	2025-02-25 17:27:45.3...	2025-02-25 17:27:45.3...	0
3	3	2	2	TRT Camelot	<a href="https://github.com/tr...">https://github.com/tr...</a>	Connect all the insta...	2025-02-25 17:29:21.0...	2025-02-25 17:29:31.4...	9.99

Figure 13: Contenido de la tabla trt\_plugin