

2022

# Trabajo U.D.5: Programación de Componentes de Sistemas ERP/CRM

SISTEMAS DE GESTIÓN EMPRESARIAL  
Tomás Mota Sánchez

## Índice

1.	¿Qué es Odoo? .....	2
2.	Módulos de Odoo:.....	2
2.1.	Módulos imprescindibles de Odoo:.....	2
	• MÓDULO DE COMPRA: PURCHASE.....	3
	• MÓDULO DE VENTAS: .....	3
	• MÓDULO DE CONTABILIDAD: .....	4
3.	Creación de un módulo en Odoo: .....	4
3.1.	Primer paso:.....	4
3.2.	Segundo paso: .....	6
3.3.	Tercer paso:.....	7
4.	Desarrollo y programación de nuestro modulo:.....	10
4.1.	Primer Paso -> Desarrollo de fichero __manifest__.py: .....	11
4.2.	Segundo Paso -> Desarrollo de fichero models.py y __init__.py:.....	12
4.3.	Tercer paso -> Comprobación de tablas en pgAdmin III: .....	15
4.4.	Cuarto paso -> Modificaciones al fichero models.py:.....	19
4.5.	Quinto paso -> Desarrollo de fichero views.xml: .....	23
4.6.	Sexto paso -> Desarrollo del fichero security.xml: .....	25
4.7.	Séptimo paso -> Reinicio de nuestro servidor local de Odoo y actualización de nuestro módulo: .....	30
5.	Prueba de nuestro módulo:.....	34
5.1.	Lista de Gestión de Muebles: .....	34
5.2.	Añadir un mueble a nuestra lista: .....	34
5.4.	Eliminar un mueble de nuestra lista: .....	39
5.5.	Exportación de nuestra lista a Excel:.....	41
6.	Síntesis personal de este proyecto: .....	42
7.	Webgrafía utilizada en este proyecto:.....	43

## 1. ¿Qué es Odoo?

Técnicamente Odoo es un software de ERP integrado, capaz de hacer todas las actividades internas de una empresa para optimización de los recursos y del tiempo, logrando así que el trabajo sea mucho más eficaz.

Dicho de otro modo, es el sistema por el cuál se puede gestionar, entre otras muchas cosas, la contabilidad, las cuentas de clientes, la gestión de compras y ventas, de almacenes... Y hasta los Recursos Humanos de una empresa.

Además, es un software de gestión libre, de manera que con pequeños cambios en el código se puede personalizar cualquier aspecto de este software de gestión para la adaptación total al negocio deseado.



## 2. Módulos de Odoo:

Todas las funcionalidades de Odoo se consiguen a través de sus módulos o aplicaciones. ¿De qué están formados estos módulos? Pues de todas las características que hacen que, con Odoo, puedas integrar en el Software todas las funcionalidades que agilizarán procesos internos y ayudaran a la organización.

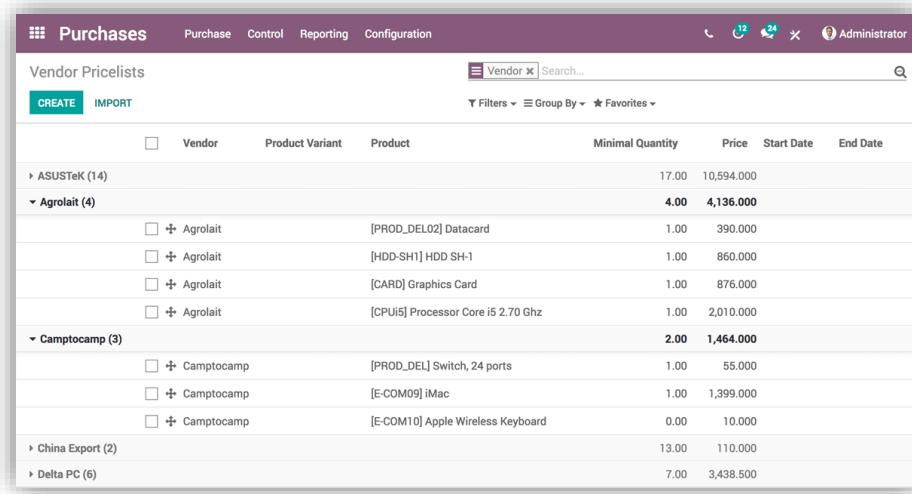


## 2.1. Módulos imprescindibles de Odoo:

En este apartado, vamos a ver los módulos de Odoo que se antojan imprescindibles para notar una mejoría en la eficacia de recursos y tiempos, y cuyas funcionalidades nos harán la vida empresarial interna, mucho más fácil.

- **MÓDULO DE COMPRA: PURCHASE**

Este módulo de Odoo está diseñado para la gestión de todas las compras de una empresa, así como la elaboración de informes y la comparación de ofertas por parte de tus proveedores.



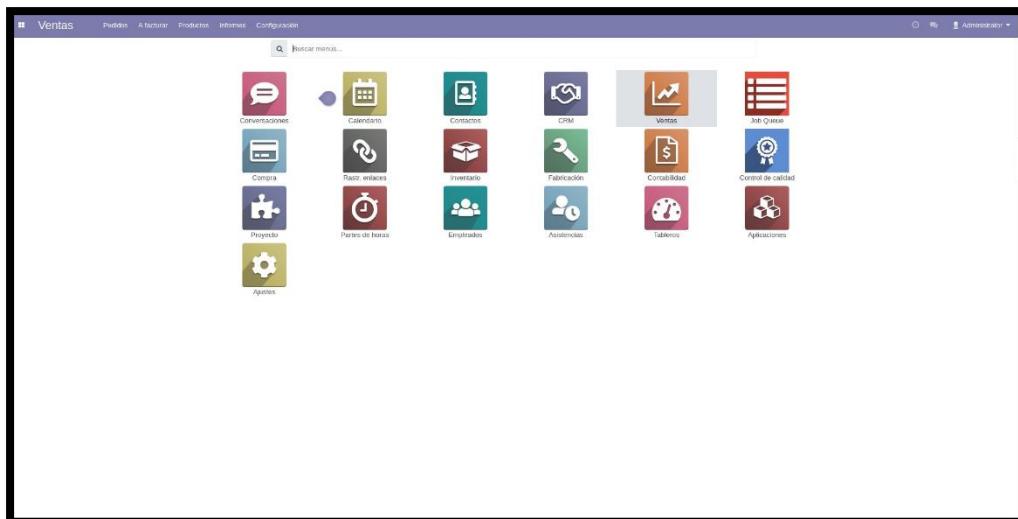
The screenshot shows the Odoo Purchase module interface. At the top, there are tabs for Purchase, Control, Reporting, and Configuration. The main area is titled "Vendor Pricelists" and shows a list of vendors with their respective minimal quantities and prices. The data includes:

Vendor	Product Variant	Product	Minimal Quantity	Price	Start Date	End Date
ASUSTeK (14)			17.00	10,594.000		
Agrolait (4)			4.00	4,136.000		
Agrolait	[PROD_DEL02] Datacard		1.00	390.000		
Agrolait	[HDD-SH1] HDD SH-1		1.00	860.000		
Agrolait	[CARD] Graphics Card		1.00	876.000		
Agrolait	[CPUi5] Processor Core i5 2.70 Ghz		1.00	2,010.000		
Campocamp (3)			2.00	1,464.000		
Campocamp	[PROD_DEL02] Switch, 24 ports		1.00	55.000		
Campocamp	[E-COM09] iMac		1.00	1,399.000		
Campocamp	[E-COM10] Apple Wireless Keyboard		0.00	10.000		
China Export (2)			13.00	110.000		
Delta PC (6)			7.00	3,438.500		

- **MÓDULO DE VENTAS:**

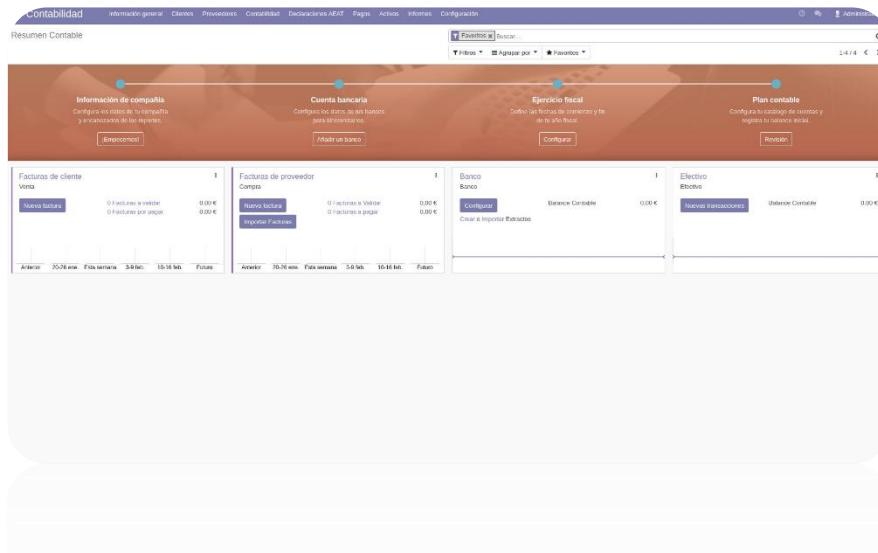
Con este módulo, podremos gestionar todas las ventas de la empresa, así como tomar decisiones acerca de estas y sus actividades adheridas.

Todo ello, a partir, como no podía ser de otra forma, de una interfaz moderna totalmente personalizable para su adaptación total al personal de ventas de cada tipo de empresa. Además, añade una interfaz Mobile para poder gestionar este módulo desde donde quieras.



- **MÓDULO DE CONTABILIDAD:**

Las características principales de este módulo, es la posibilidad de integración con un banco, de manera que podrás obtener de manera rápida y fluida los extractos bancarios sincronizados. ¿En qué se traduce? En un ahorro sustancial de tiempo en la conciliación banco-Contabilidad empresarial.



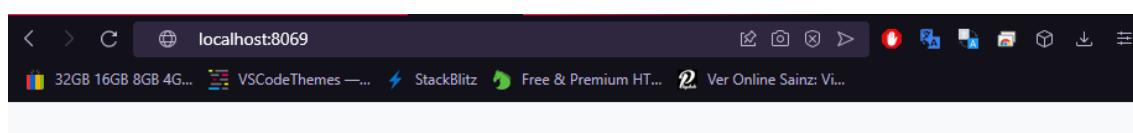
### 3. Creación de un módulo en Odoo:

A la hora de la creación de un módulo, antes tendremos que instalar Odoo para nuestro dispositivo. En mi caso instalare la versión de Odoo 13.0 y lo haré en el sistema operativo "Windows 11 Pro".

En este caso no explicare la instalación de Odoo ya que fue visto en anteriores trabajos de esta asignatura. Así que iremos directos a la ejecución del nuestro servidor local de Odoo.

#### 3.1. Primer paso:

Para esto buscaremos en nuestro navegador la siguiente dirección url.



Tras poner la dirección por primera vez, tendremos que crear nuestra base datos para la utilización de Odoo.

Para eso tendremos que llenar el formulario que nos pide Odoo para la creación de esta BBDD y del usuario que será administrador dentro de ella.

Los campos son obligatorios, además tendremos que dar una contraseña maestra para nuestra base de datos.

Los datos que nos pedirán para crear este usuario son: Email, contraseña, número de teléfono, lenguaje y país donde vive el usuario.

Tras esto, le daremos al botón de continuar para finalizar el proceso de la creación de nuestra base de datos.

Create Database

Master Password  
.....

Database Name webinardb

Email tomasmotasanchez14@gmail.com

Password ..... 

Phone number 682375195

Language Spanish / Español

Country Spain

Demo data

To enhance your experience, some data may be sent to Odoo online services. See our [Privacy Policy](#).

 Continue

Finalmente, nos llevará al sistema de logeo a nuestro sistema de Odoo. En el podemos ver que esta seleccionada la base de datos anteriormente creada "webinardb", aunque nos da la opción de elegir otra base de datos que tengamos creada.

Your logo

Base de datos  
webinardb 

Correo electrónico  
tomasmotasanchez14@gmail.com

Contraseña  
.....

Iniciar Sesión

Inicie sesión como superusuario

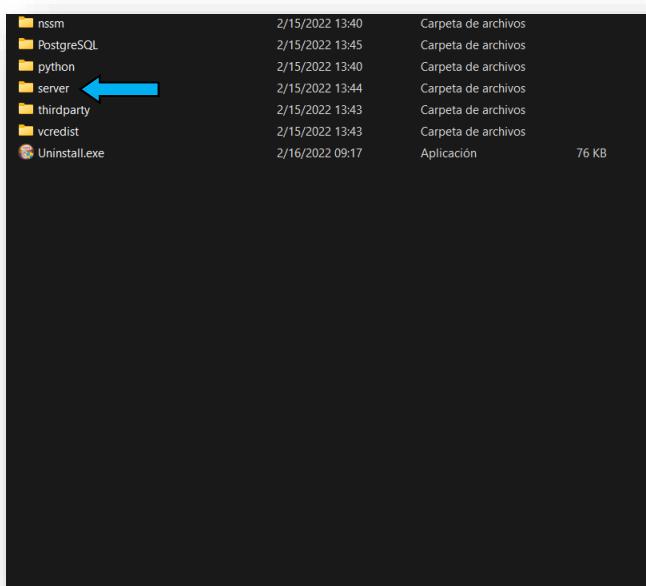
  


### 3.2. Segundo paso:

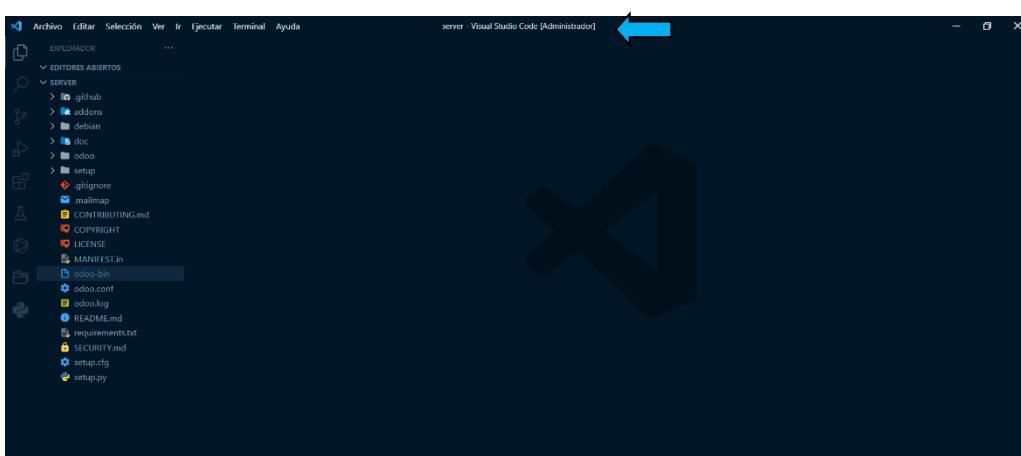
En este segundo paso tendremos que tocar la configuración de nuestro servidor de Odoo a través de Visual Studio Code.

Para eso tendremos que buscar la carpeta Odoo13.0 en nuestro explorador de archivos. En mi caso esta carpeta se encuentra en "C:\Program Files(x86) \Odoo13.0".

El siguiente paso tendremos será buscar la carpeta server, que se encuentra ubicada dentro de la carpeta anteriormente mencionada.

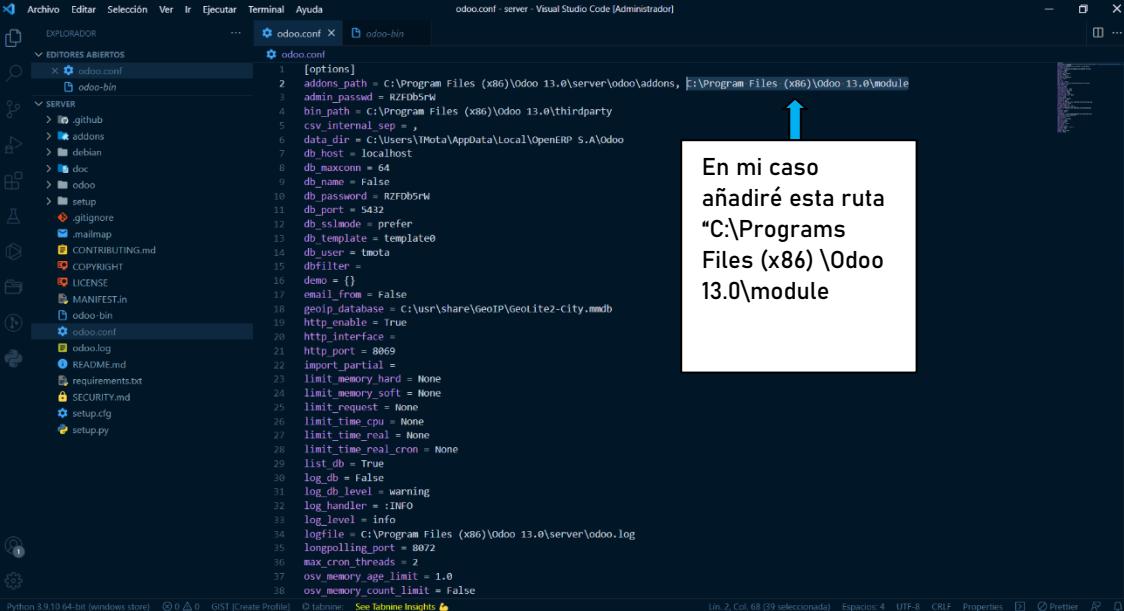


Tras esto, abriremos esta carpeta en Visual Studio Code pero en modo Administrador ya que necesitaremos estos permisos por si en algún futuro tenemos que cambiar parámetros de ficheros de esta carpeta.



El fichero que utilizaremos será el fichero “odoo.conf”, el cuál tiene todos los parámetros de configuración de nuestro servidor de Odoo y el que será utilizado en nuestra base de datos a la hora de la creación del módulo.

En este fichero tocaremos el parámetro (`addons_path`), en este añadiremos la ruta de la carpeta donde se incluirá el módulo que crearemos próximamente.



```

Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR odoo-bin odoo.conf
odoo-bin
odoo.conf
[options]
1 addons_path = C:\Program Files (x86)\Odoo 13.0\server\odoo addons, E:\Program Files (x86)\Odoo 13.0\module
2 admin_passwd = RZFDb5rW
3 bin_path = C:\Program Files (x86)\Odoo 13.0\thirdparty
4 csv_internal_sep
5 data_dir = C:\Users\Hector\AppData\Local\OpenERP 5.0\Odoo
6 db_host = localhost
7 db_maxconn = 64
8 db_name = False
9 db_password = RZFDb5rW
10 db_port = 5432
11 db_stimode = prefer
12 db_template = template0
13 db_user = bmoa
14 dbfilter =
15 demo = []
16 email_from = False
17 geoip_database = c:/usr/share/GeoIP/GeoLite2-City.mmdb
18 http_enable = True
19 http_interface
20 http_port = 8069
21 import_partial =
22 limit_memory_hard = None
23 limit_memory_soft = None
24 limit_request = None
25 limit_time_cpu = None
26 limit_time_real = None
27 limit_time_real_cron = None
28 log_db = True
29 log_db_level = warning
30 log_handler = INFO
31 log_level = info
32 log_file = c:/Program Files (x86)\Odoo 13.0\server\odoo.log
33 longpolling_port = 8072
34 max_cron_threads = 2
35 osv_memory_age_limit = 1.0
36 osv_memory_count_limit = False

```

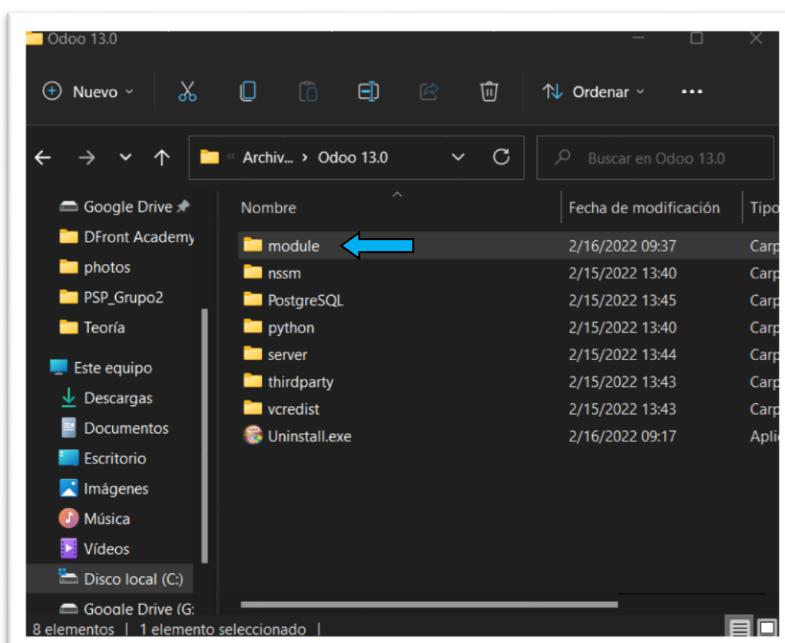
En mi caso  
añadiré esta ruta  
“C:\Programs  
Files (x86) \Odoo  
13.0\module”

### 3.3. Tercer paso:

Después de finalizar el segundo paso, nuestro objetivo será desarrollar nuestro modulo deseado para añadirlo a nuestro sistema personal de Odoo.

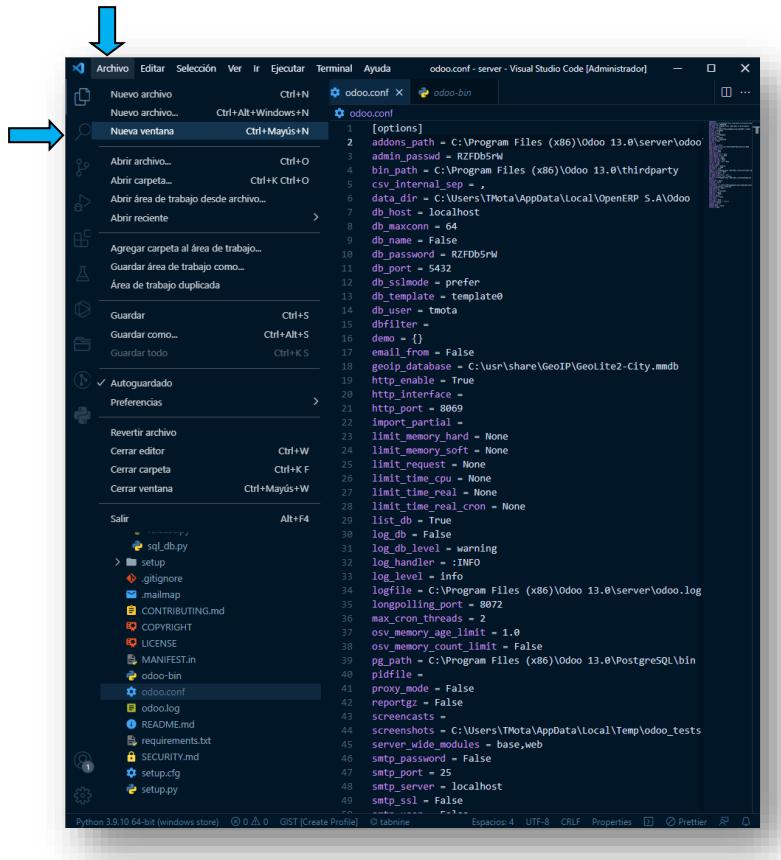
El paso previo para empezar a desarrollarlo será crear nuestra carpeta “module” en la ruta especificada anteriormente en el archivo de configuración de Odoo (odoo.conf).

Para eso nos iremos a la ruta C:\Programs Files (x86) \Odoo 13.0, y dentro de esta crearemos la carpeta deseada.



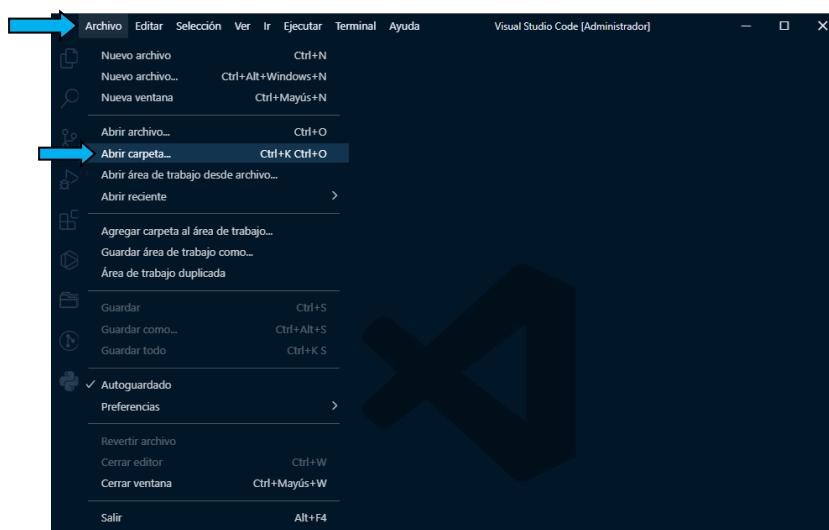
Tras finalizar la creación de nuestra carpeta, tendremos que volver a “Visual Studio Code”.

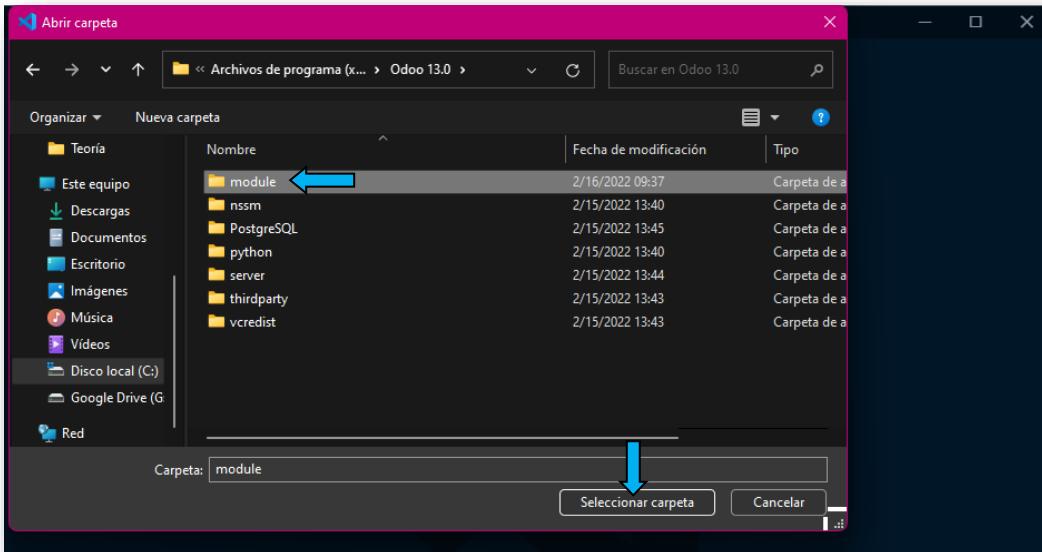
Dentro de Visual Studio Code tendremos que darle a la pestaña “Archivo” y seleccionaremos la opción “Nueva Ventana”.



Después de ejecutar la opción anterior, se nos abrirá una nueva pestaña en Visual Studio Code. En el seleccionaremos la pestaña “Archivo”, y después le daremos la opción “Abrir carpeta”.

En este caso se nos abrirá una nueva pestaña donde tendremos que elegir la carpeta donde se irá desarrollando nuestro modulo, en nuestro caso será “module”.

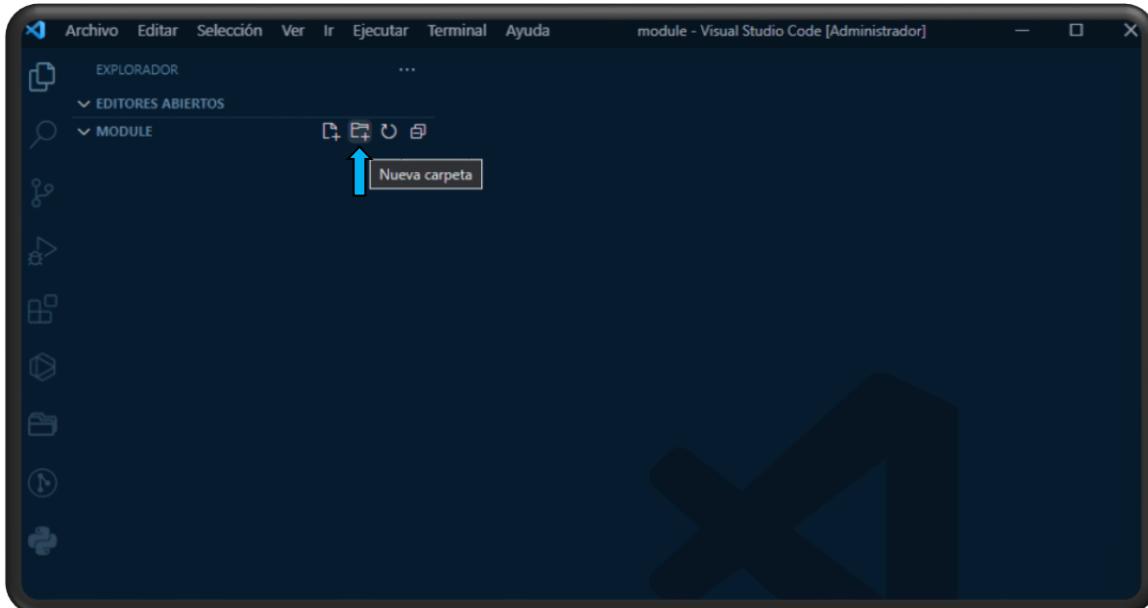


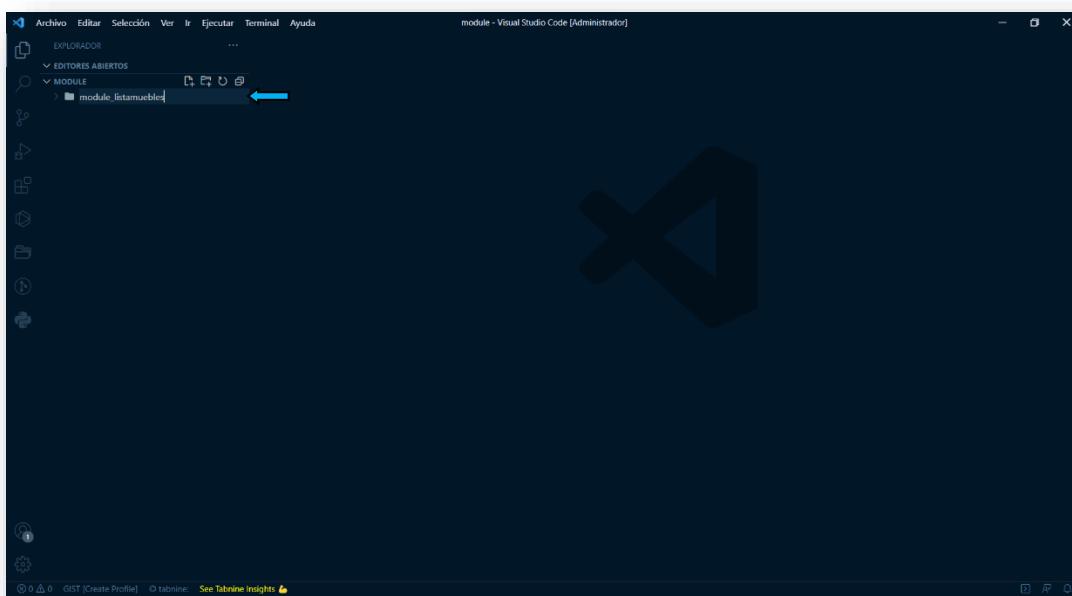


Ahora veremos que esta carpeta ha sido abierta por la aplicación donde iremos desarrollando y programando nuestro modulo.

En este caso crearemos otra carpeta más específica para gestionar el módulo que iremos desarrollando progresivamente.

Más tarde tendremos que darle un nuevo nombre a esta carpeta, en mi caso le llamaré “module\_listamuebles”.



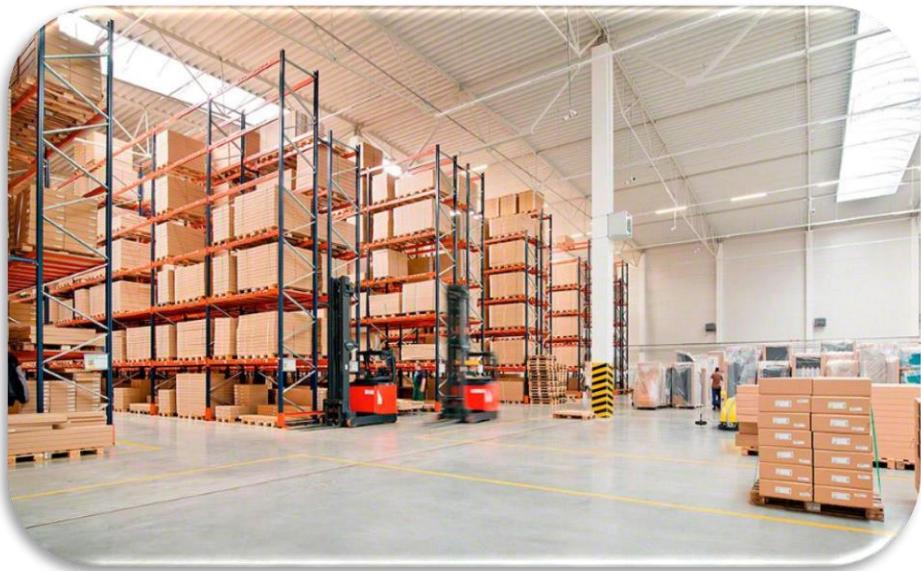


#### 4. Desarrollo y programación de nuestro modulo:

Al ver cómo es la creación de nuestro primer módulo, es la hora de ir ya desarrollando progresivamente la programación del uso de este.

En mi caso mi modulo será desarrollado para una lista de gestión de muebles, ya que en anteriores trabajos fuimos desarrollando una web para la venta de muebles.

Por eso, en mi caso le daré un enfoque más a la gestión de nuestro almacén de nuestra empresa.

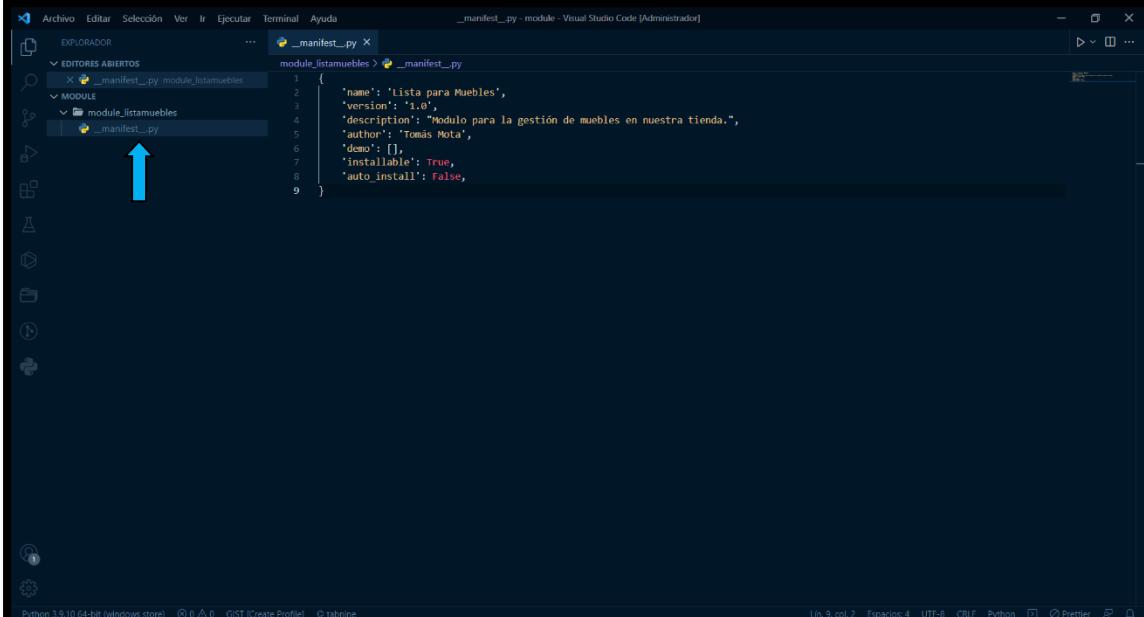


#### 4.1. Primer Paso -> Desarrollo de fichero \_\_manifest\_\_.py:

A la hora de empezar a desarrollar, tendremos que crear nuestro primer fichero dentro de la carpeta "module\_listamuebles" es \_\_manifest\_\_.py.

¿Qué hace este fichero? Este fichero lo que hará es darle información al usuario sobre el módulo. En él se podrá dar el nombre del módulo, la versión de este, el autor del que ha ido desarrollándolo y varios parámetros más.

En la siguiente imagen podremos ver como he ido desarrollando este fichero con los parámetros que he ido comentado:

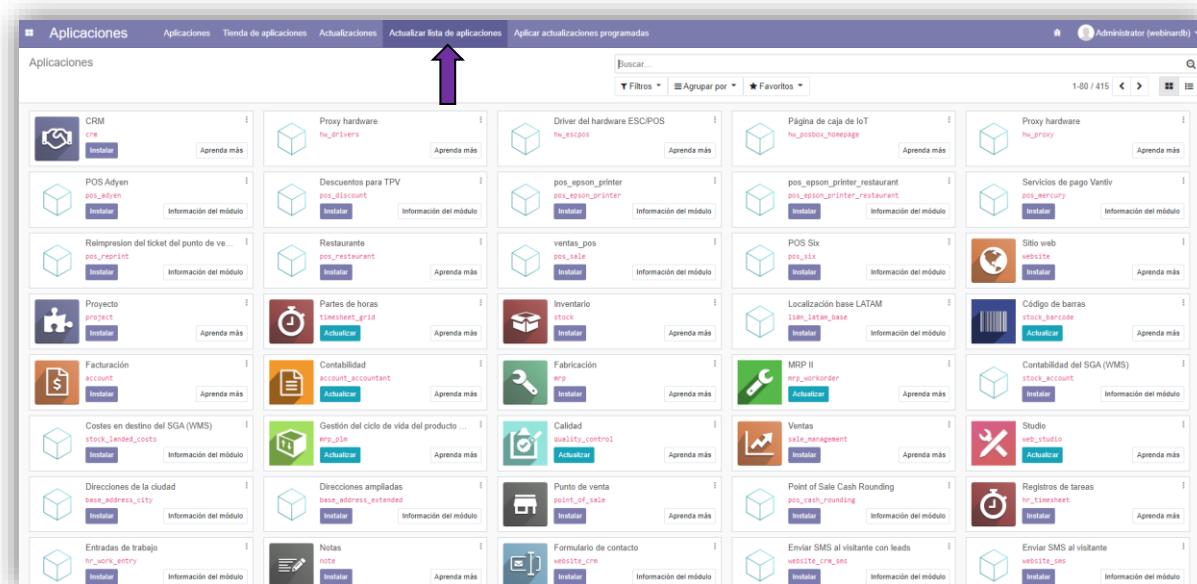


```

 1 {
 2     'name': 'Lista para Muebles',
 3     'version': '1.0',
 4     'description': "Modulo para la gestión de muebles en nuestra tienda.",
 5     'author': 'Tomás Mata',
 6     'demo': [],
 7     'installable': True,
 8     'auto_install': False,
 9 }

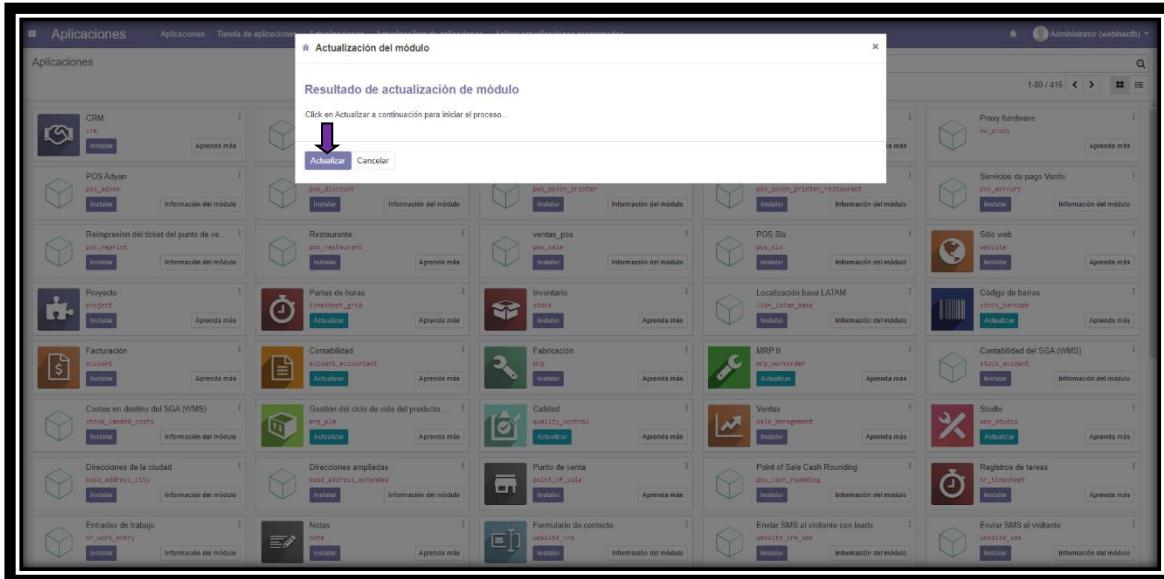
```

Después de finalizar el desarrollo de este, nos iremos a nuestro sistema de Odoo y le daremos a la opción que se encuentra en la barra de tareas "Actualizar lista de aplicaciones".



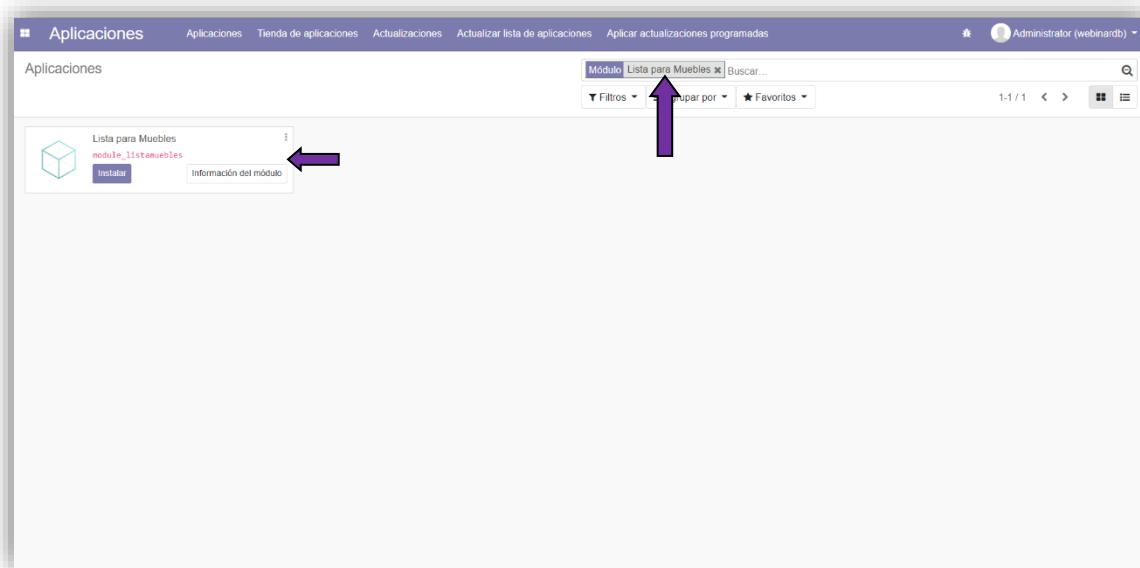
## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

Tras esto nos saldrá un mensaje donde nos dirá si queremos actualizar esta lista, en nuestro caso le daremos al botón Actualizar.



Al actualizarse la lista, iremos al buscador y escribiremos el nombre que le hemos dado a nuestro modulo en el fichero anteriormente desarrollado.

En la siguiente imagen podremos ver que el resultado es bueno, el módulo se encuentra ahora mismo en nuestro sistema de Odoo y se puede instalar.



**4.2. Segundo Paso -> Desarrollo de fichero models.py y \_\_init\_\_.py:**  
Ahora tendremos que crear nuestro segundo fichero dentro de la carpeta "module\_listamuebles" el cual es models.py.

TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

A screenshot of Visual Studio Code showing the file structure of a Python module named "module\_listamuebles". The tree view on the left shows the following structure:

- \_\_manifest\_\_.py
- models.py

The status bar at the bottom displays the path "module\_listamuebles" and the file name "models.py". A large blue arrow points upwards from the status bar towards the \_\_manifest\_\_.py file in the tree view.

En este fichero se dará el modelo del módulo desarrollado con código Python, generalmente siempre es llamado `models.py`. Este archivo representa la estructura de la base de datos de nuestro módulo.

En el `models.py` se escriben clases Python que representan las tablas y variables de clase que representan las columnas de las tablas.

En mi caso al desarrollar un módulo para la gestión de muebles, en esta imagen se puede ver el resultado de mi modelo.

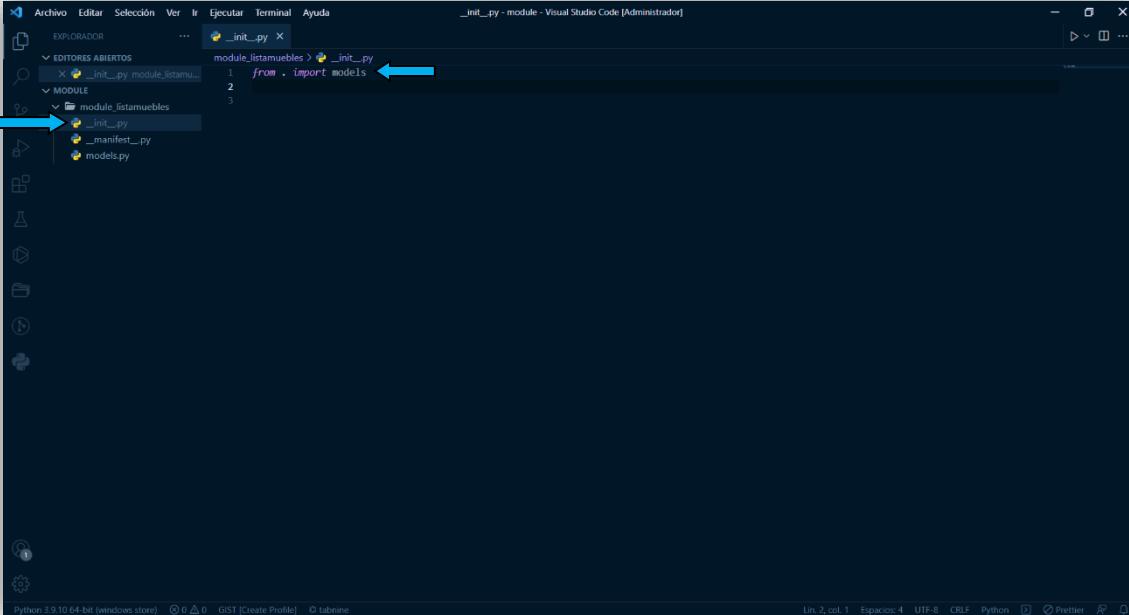
```
models.py 1 x
models.py module_is_1
module_listamuebles > models.py > modelspy ...
1   from odoo import models, fields
2
3   class ListaMuebles(models.Model):
4       _name="lista.muebles"
5       name = fields.Char("Nombre del mueble")
6       description = fields.Char("Descripción del mueble")
7       category = fields.Selection(selection=[("sofas","Sofás"),("silla","Sillas"),("mesas","Mesas"),("librerias","Librerías"),("camas","Camas")])
8       productstatus = fields.Selection(selection=[("nuevo","Nuevo"),("segundamano","Segunda Mano"),("reacondicionado","Reacondicionado")])
9       origincompany = fields.Char("Empresa de Origen")
10      saleprice = fields.Char("Precio de Venta")
11
12
```

Tras ver la funcionalidad del fichero `models.py`, nos tocará ver para la finalidad que tiene el fichero `__init__.py`.

El principal uso de `__init__.py` es inicializar paquetes de Python. La forma más fácil de darnos cuenta es ir hasta el directorio donde tenemos instalado python y fijarnos en las librerías y paquetes que trae por defecto Python.

El archivo `__init__.py` se utiliza para realizar configuraciones de importación. Algo que se utiliza mucho en este archivo es la importación de clases, funciones, etc, para que puedan ser utilizadas en otros paquetes.

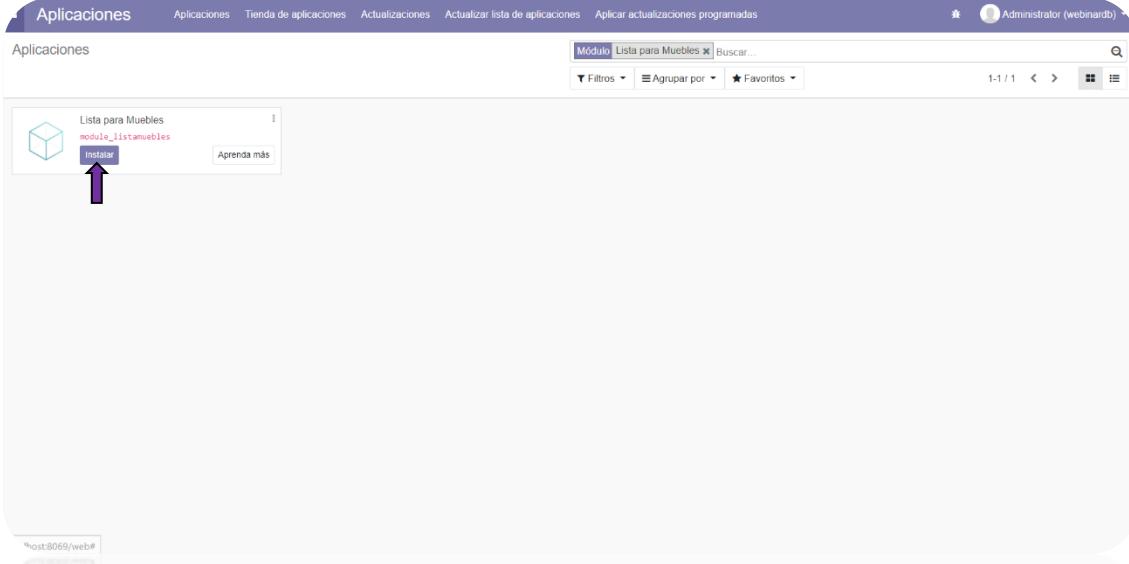
En la siguiente imagen podemos ver la importación que nos hará falta y ha sido utilizada para desarrollar este archivo.



```
__init__.py - module - Visual Studio Code [Administrador]
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR ... __init__.py X
EDTORES ABIERTOS module_listamuebles > __init__.py
MODULE
  module_listamuebles
    __init__.py
    __manifest__.py
    models.py
  __init__.py
  module_listamuebles

Python 3.9.10 64-bit (windows store) 0 0 GIST [Create Profile] 0 tabnine
Líns 2, col. 1 Espacio: 4 UTF-8 CRLF Python 0 0 Prettier 0 0
```

Después de ver las funcionalidades de estos dos ficheros, guardaremos lo desarrollado y volveremos a nuestro sistema local de Odoo para ya si instalar nuestro módulo.



Aplicaciones

Módulo Lista para Muebles Buscar...

Filtros Agrupar por Favoritos

host:8069/web#

The screenshot shows the Odoo application manager interface. In the top navigation bar, there are tabs for 'Aplicaciones', 'Aplicaciones', 'Tienda de aplicaciones', 'Actualizaciones', 'Actualizar lista de aplicaciones', and 'Aplicar actualizaciones programadas'. On the right, it says 'Administrator (webinar0)'. Below the tabs, there's a search bar with 'Módulo' and 'Lista para Muebles' entered, along with a 'Buscar...' button. Underneath the search bar are buttons for 'Filtros', 'Agrupar por', and '★ Favoritos'. A status bar at the bottom right shows '1-1 / 1' and icons for refresh, back, forward, and list.

**En esta imagen podemos ver como nuestro modulo ha sido instalado.**

**4.3. Tercer paso -> Comprobación de tablas en pgAdmin III:**  
**En este tercer paso veremos la herramienta pgAdmin III, ya que esta herramienta va ligada a Odoo.**

PgAdmin III es un programa de computadora que permite la interacción y manipulación de la información almacenada en un servidor de base de datos de PostgreSQL a través de una interfaz gráfica, fácil e intuitiva. También permite controlar aspectos tanto del servidor como de las bases de datos, ejecutar sentencias SQL, desarrollar bases de datos complejas, etc.



Tras esta breve explicación sobre esta herramienta, iniciaremos pgAdmin.

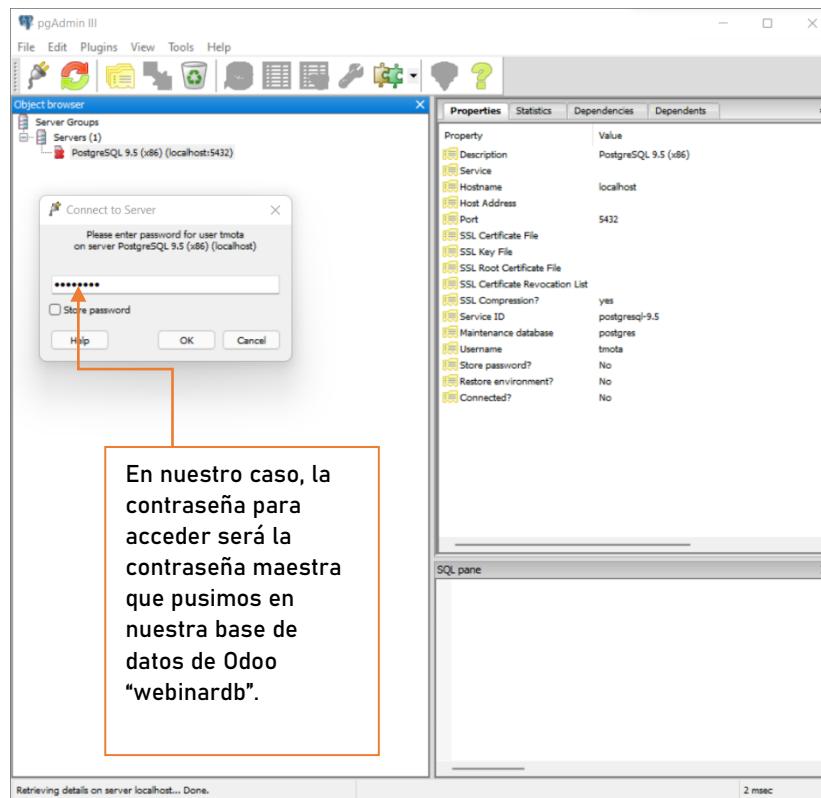
Al iniciar nos encontraremos que pgAdmin reconoce nuestro servidor local de Odoo.

The screenshot shows the pgAdmin III interface. At the top, there's a menu bar with 'File', 'Edit', 'Plugins', 'View', 'Tools', and 'Help'. Below the menu is a toolbar with various icons. The main window has a 'Server Groups' tree on the left, showing 'Servers (1)' with 'PostgreSQL 9.5 (x86) (localhost:5432)'. An orange arrow points to this server entry. To the right is a 'Properties' panel with several tabs: 'Properties', 'Statistics', 'Dependencies', and 'Dependents'. The 'Properties' tab is selected, displaying the following configuration details:

Property	Value
Description	PostgreSQL 9.5 (x86)
Service	
Hostname	localhost
Host Address	
Port	5432
SSL Certificate File	
SSL Key File	
SSL Root Certificate File	
SSL Certificate Revocation List	
SSL Compression?	yes
Service ID	postgreSQL-9.5
Maintenance database	postgres
Username	trnola
Store password?	No
Restore environment?	No
Connected?	No

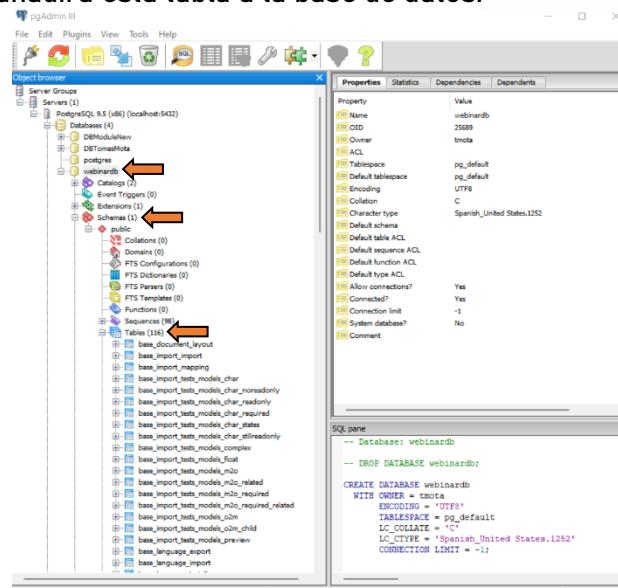
Para iniciar la conexión de la base de datos de pgAdmin con el sistema local de Odoo, tendremos que seleccionar el servidor.

Una vez seleccionado nos pedirá la contraseña de acceso que tiene nuestro sistema local de Odoo.



Tras acceder exitosamente a nuestro sistema local de Odoo, podemos ver la base de datos que gestionamos en Odoo. En mi caso tengo 4 bases de datos, pero la principal que usaremos es la anteriormente mencionada "webinardb".

En este punto buscaremos las tablas que tenemos en nuestro servidor local de Odoo y sobre todo en específico la tabla "lista\_muebles" ya que al instalar anteriormente nuestro modulo, se añadirá esta tabla a la base de datos.



## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

The screenshot shows the pgAdmin III interface. In the Object browser, under the 'ir\_filters' category, the 'lista\_muebles' table is selected. The Properties tab displays various table properties such as Name (lista\_muebles), OID (29406), Owner (tmota), Tablespace (pg\_default), and ACL. A callout box highlights the 'lista\_muebles' entry in the Object browser with the text: "Aquí podemos observar que la tabla 'lista\_muebles' se encuentra correctamente entre nuestras tablas instaladas." Below the table details, a SQL pane shows the table's creation command.

Property	Value
Name	lista_muebles
OID	29406
Owner	tmota
Tablespace	pg_default
ACL	
Of type	
Primary key	id
Rows (estimated)	0
Fill factor	
Rows (counted)	0
Inherits tables	No
Inherited tables count	0
Unlogged?	No
Has OIDs?	No
System table?	No
Comment	lista.muebles

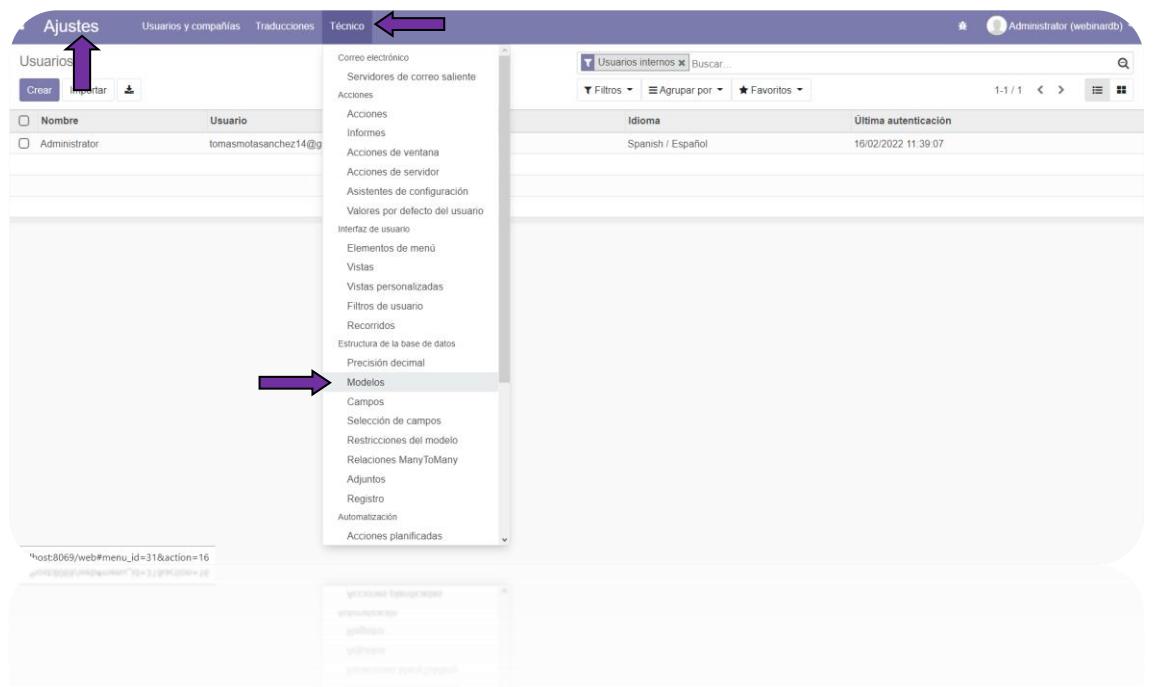
```
CREATE TABLE lista_muebles (
    id integer,
    name character varying(128),
    description character varying(1024),
    category character varying(128),
    productstatus character varying(128),
    origincompany character varying(128),
    saleprice numeric,
    create_uid integer,
    create_date timestamp without time zone,
    write_uid integer,
    write_date timestamp without time zone
);
```

Después de ver que el modelo de nuestro modulo se encuentra bien instalado en el sistema local de Odoo gracias a pgAdmin III.

Nos tocara ver si también se encuentra visible en Odoo, para eso tendremos que irnos a nuestro sistema local Odoo y darle a la opción de Ajustes.

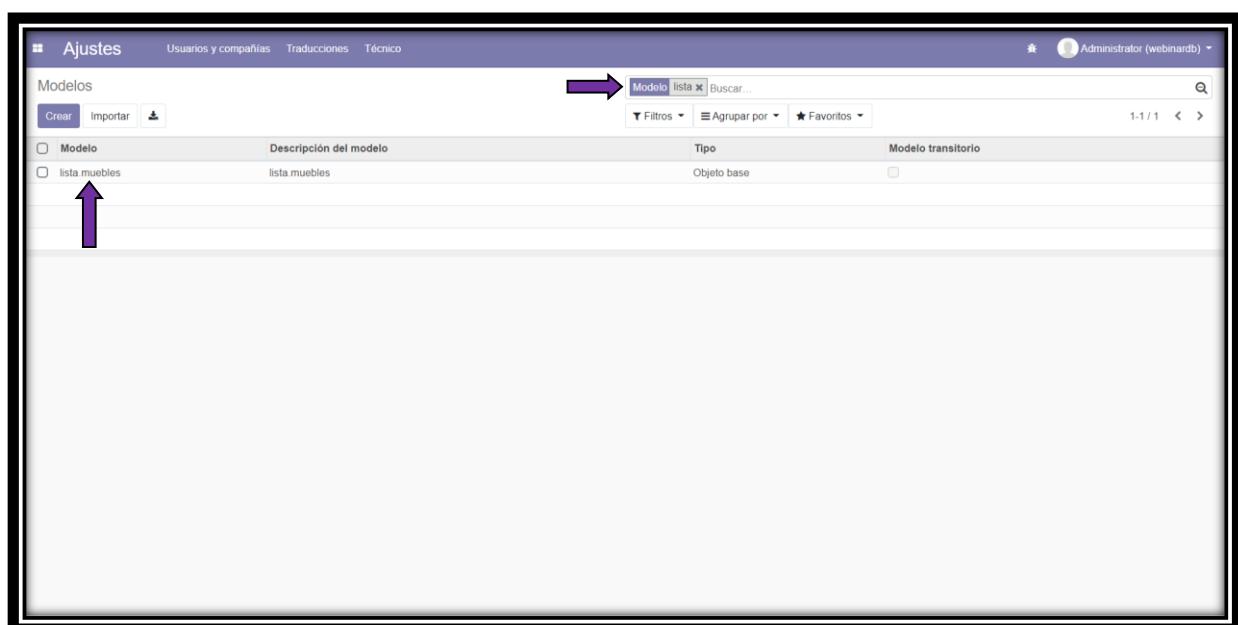
The screenshot shows the Odoo application settings page. A purple arrow points to the 'Ajustes' (Settings) button in the top navigation bar. The main content area lists the 'Lista para Muebles' module, which is installed. Another purple arrow points to the 'Lista para Muebles' entry in the list.

Tras llegar a nuestros ajustes le daremos al botón “Técnico” que se encuentra en nuestra barra de herramientas superior, el cual se abrirá una lista desplegable donde tendremos que seleccionar la opción “Modelos”.



Más tarde se cargará una lista de los modelos de los módulos que tenemos instalados, pero nosotros nos iremos al buscador y empezaremos a buscar el modelo deseado.

En la siguiente imagen podemos ver que el modelo ha sido encontrado con éxito, en mi caso mi modelo es “lista\_muebles”.



Si pinchamos dentro de este modelo podemos ver el mismo resultado que en pgAdmin III.

Eso significara que el anterior fichero "models.py" fue un éxito y al instalar nuestro módulo, se añadió correctamente a nuestro sistema local de Odoo.

Nombre de campo	Etiqueta de campo	Tipo de campo	Requerido	Sólo lectura	Indexado	Tipo
_last_update	Last Modified on	Fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
category	Category	Selección	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
create_date	Created on	Fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
create_uid	Created by	many2one	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
description	Descripción del mueble	Carácter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
display_name	Display Name	Carácter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
id	ID	entero	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
name	Nombre del mueble	Carácter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
origincompany	Empresa de Origen	Carácter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
productstatus	Productstatus	Selección	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
saleprice	Precio de Venta	Carácter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
write_date	Last Updated on	Fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
write_uid	Last Updated by	many2one	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base

#### 4.4. Cuarto paso -> Modificaciones al fichero models.py:

Este paso es opcional, pero en mi caso quiero que podáis ver que si volvemos a desarrollar o cambiar nuestro fichero "models.py" de nuestro módulo y lo actualizamos en el sistema local de Odoo, los modelos serán modificados o se añadirán nuevos campos en esta.

Lo primero que haremos será modificar mis campos del fichero "models.py", en mi caso he cambiado algunos nombres de mis campos y los he traducido al español.

Aquí en esta imagen podemos ver cómo han sido modificados:

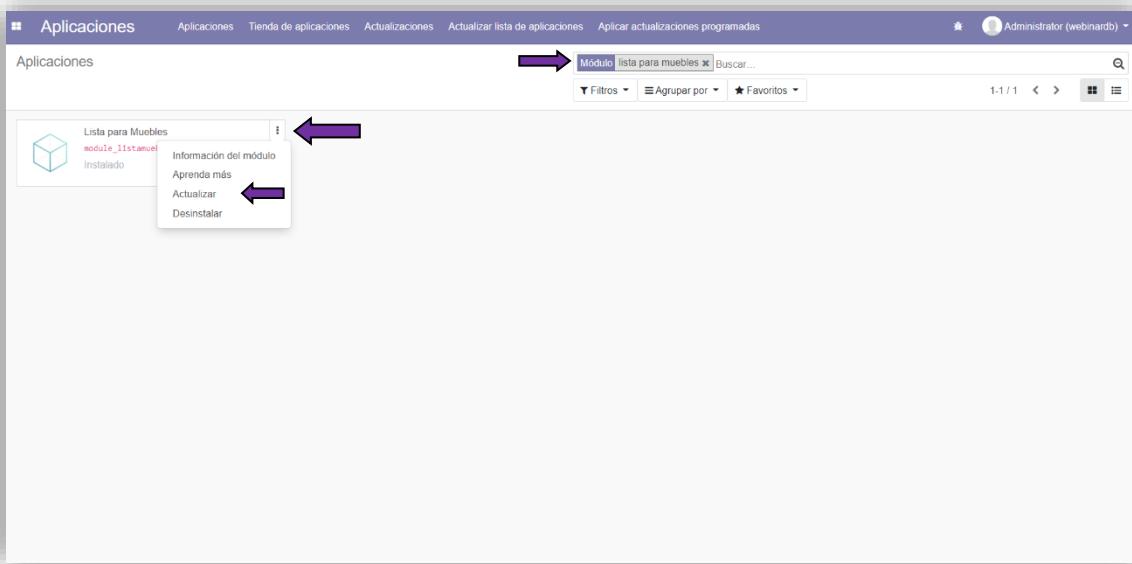
```

  Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
  EXPLORADOR ... models.py x module_listamuebles > models.py ...
  EDITORES ABIERTOS module_listamuebles > models.py ...
  MODULE module_listamuebles
    > __pycache__
    > __init__.py
    > manifest.py
    models.py
  models.py - module_listamuebles - Visual Studio Code [Administrador]
  module_listamuebles > models.py ...
  1   from odoo import models, fields
  2
  3   class ListaMuebles(models.Model):
  4       _name='lista.mueble'
  5       nombre = fields.Char('Nombre del mueble')
  6       descripción = fields.Char("Descripción del mueble")
  7       categoría = fields.Selection(selection=[("sofás", "Sofás"), ("sillas", "Sillas"), ("mesas", "Mesas"), ("librerías", "Librerías"), ("camas", "Camas")])
  8       productstatus = fields.Selection(selection=[("nuevo", "Nuevo"), ("segundamano", "Segunda Mano"), ("recondicionado", "Reacondicionado")])
  9       empresasorigen = fields.Char("Empresa de Origen")
 10      precioventa = fields.Char("Precio de Venta")

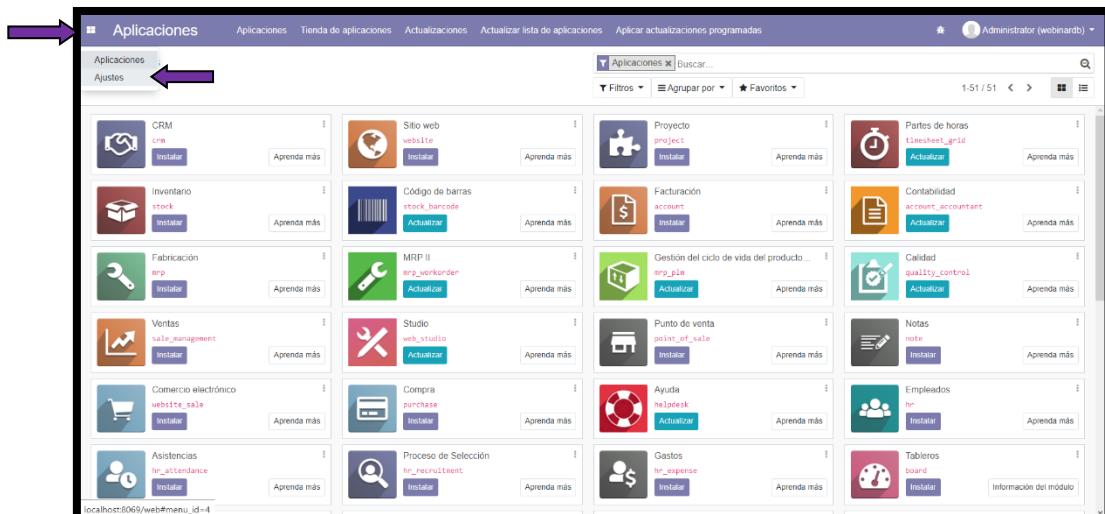
```

En el siguiente paso nos iremos a nuestro sistema de Odoo y en nuestro módulo "Lista para muebles" le daremos a la opción "Actualizar" que se encuentra en esta lista desplegable. Más tarde nos saldrá una ventana emergente diciendo si queremos actualizar nuestro módulo, en nuestro caso le daremos al botón "Actualizar".

## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

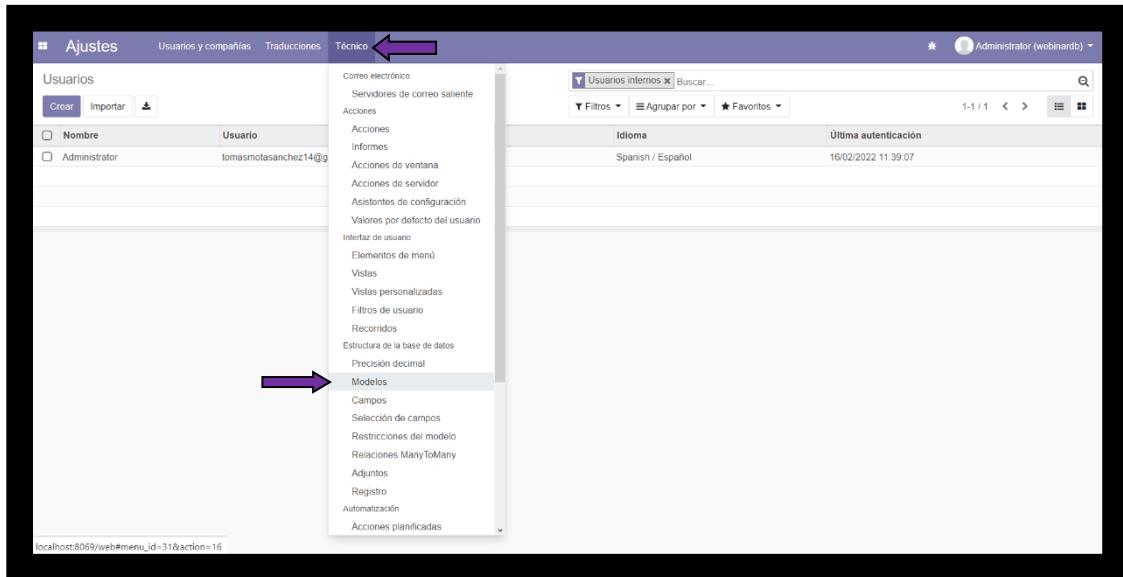


Tras actualizarse nuestro módulo, lo que haremos será irnos a nuestro sistema local Odoo y darle a la opción de Ajustes.



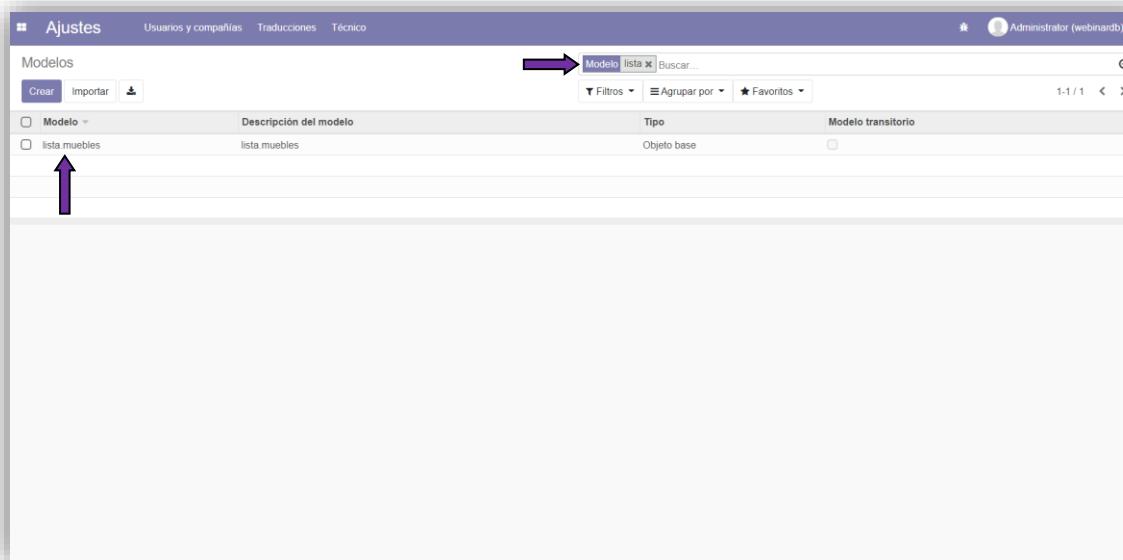
## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

Tras llegar a nuestros ajustes le daremos al botón “Técnico” que se encuentra en nuestra barra de herramientas superior, el cual se abrirá una lista desplegable donde tendremos que seleccionar la opción “Modelos”.



Más tarde se cargará una lista de los modelos de los módulos que tenemos instalados, pero nosotros nos iremos al buscador y empezaremos a buscar el modelo deseado.

En la siguiente imagen podemos ver que el modelo se sigue encontrando con éxito, en mi caso mi modelo es “lista\_muebles”.



Si pinchamos dentro de este modelo podemos ver que los campos que fueron cambiaron en el fichero "models.py" también han sido cambiado dentro de Odoo.

En este caso podemos ver una imagen donde se puede ver que los cambios han sido realizados con éxito:

## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

Modelos / lista.muebles

Nombre de campo	Etiqueta de campo	Tipo de campo	Requerido	Sólo lectura	Indexed	Tipo
__last_update	Last Modified on	Fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
categoria	Categoría	Selección	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
create_date	Created on	Fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
create_uid	Created by	many2one	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
descripcion	Descripción del mueble	Carácter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
display_name	Display Name	Carácter	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
empresaorigen	Empresa de Origen	Carácter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
id	ID	entero	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
nombre	Nombre del mueble	Carácter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
recioventa	Precio de Venta	Carácter	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
productstatus	Productstatus	Selección	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Campo base
write_date	Last Updated on	Fecha y hora	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base
write_uid	Last Updated by	many2one	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	Campo base

También podemos ver que en la aplicación pgAdmin III, los cambios han sido buenos y los nombres de los campos han sido modificados correctamente.

pgAdmin III

Object browser

Properties

Name	lista_muebles
OID	29406
Owner	tmota
Tablespace	pg_default
ACL	
Of type	
Primary key	id
Rows (estimated)	0
Fill factor	
Rows (counted)	0
Inherits tables	No
Inherited tables count	0
Unlogged?	No
Has OIDs?	No
System table?	No
Comment	lista.muebles

SQL pane

```
-- Table: public.lista_muebles
-- DROP TABLE public.lista_muebles;
CREATE TABLE public.lista_muebles
(
    id integer NOT NULL DEFAULT nextval('lista_muebles_id_seq'),
    productstatus character varying, -- Productstatus
    create_uid integer, -- Created by
    create_date timestamp without time zone, -- Created on
    write_uid integer, -- Last Updated by
    write_date timestamp without time zone, -- Last Updated on
    nombre character varying -- Nombre del mueble
)
```

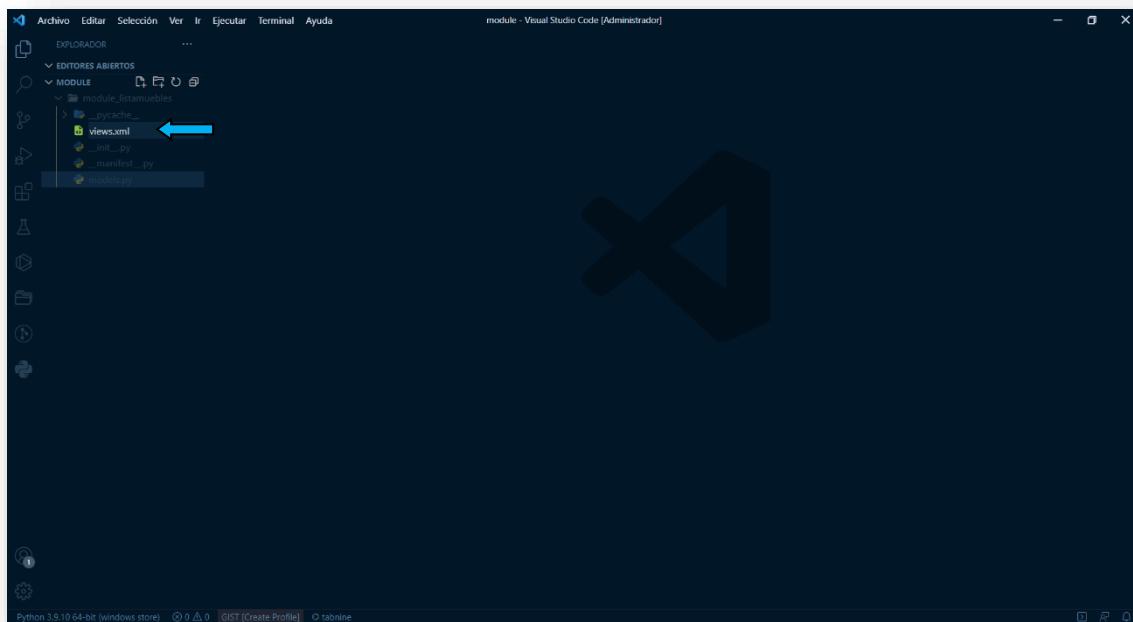
#### 4.5. Quinto paso → Desarrollo de fichero views.xml:

En este paso veremos el cómo crear la interfaz gráfica para que los usuarios interactúen con la aplicación de tareas pendientes.

En anteriores paso ya hemos hecho la capa de modelo, y ahora necesitamos la capa de vista para la interfaz de nuestro módulo.

Cada componente de la interfaz del módulo se almacena en un registro de base de datos, al igual que los registros de otros módulos. Estos se añaden elementos de interfaz de usuario a la base de datos cargando los datos correspondientes de archivos XML.

Lo primero que haremos será irnos a Visual Studio Code y en el crear un fichero nuevo llamado “views.xml”.



Después iremos desarrollando nuestra vista dependiendo del sentido y relación a la idea que tenemos con nuestro módulo. En este fichero tendremos que ir llamando al nombre de nuestra carpeta donde se encuentra el módulo, nombre del modelo, campos que se encuentra en la tabla ...

```

<?xml version="1.0" encoding="UTF-8"?>
<odoo>
    <record id="view_tree_todo" model="ir.ui.view">
        <field name="name">view.tree.todo</field>
        <field name="model">module_listamuebles.listamuebles</field>
        <field name="arch" type="xml">
            <treetree>
                <field name="nombre"/>
                <field name="descripcion"/>
                <field name="categoria"/>
                <field name="productstatus"/>
                <field name="empresaorigin"/>
                <field name="precoventa"/>
            </treetree>
        </field>
    </record>

    <record id="action_view_todo" model="ir.actions.act_window">
        <field name="name">Lista de gestión de Muebles</field>
        <field name="res_model">listamuebles.listamuebles</field>
        <field name="view_mode">tree,form</field>
    </record>

    <menuitem id="menu_todo" parent="--" action="module_listamuebles.action_view_todo"/>
</odoo>

```

Ahora podemos comenzar editando el archivo `__manifest__.py` para declarar estos nuevos archivos de datos, en nuestro caso llamaremos al archivo anteriormente desarrollado.

```

manifest_py - module - Visual Studio Code [Administrador]
Archivo Editar Selección Ver Ir Ejecutar Terminal Ayuda
EXPLORADOR ... __manifest__.py
EDITORES ABIERTOS X __manifest__.py module_listamuebles > __manifest__.py
MODULE
    > __pycache__
        __init__.py
        manifest.py
    models.py
    views.xml
1  {
2      'name': 'Lista para Muebles',
3      'version': '1.0',
4      'summary': 'Aplicación para gestionar nuestra lista de muebles',
5      'description': 'Modulo para la gestión de muebles en nuestra tienda.',
6      'author': 'Tomás Mota',
7      'website': 'https://github.com/TomasMota107tab-repositories',
8      'data': [
9          'views.xml'
10     ],
11     'demo': [],
12     'installable': True,
13     'auto_install': False,
14 }

```

Para finalizar este punto, nos iremos a nuestro sistema local de Odoo y si recargamos nuestra página y le damos a la opción **Vistas**, nos aparece la vista que hemos desarrollado en nuestro “views.py”.

Nombre de la vista	Tipo de vista	Modelo	ID externo	Vista heredada
view.tree_todo	Árbol	lista.muebles	module_listamuebles.view_tree_todo	

#### 4.6. Sexto paso -> Desarrollo del fichero security.xml:

En este paso veremos el desarrollo del fichero “security.xml” para eso nos tendremos que ir a nuestro sistema local de gestión de Odoo, y en los ajustes de este pinchar sobre nuestro usuario Administrador.

Nombre	Usuario	Idioma	Última autenticación
Administrador	tomasmotasanchez14@gmail.com	Spanish / Español	16/02/2022 11:39:07

Se abrirá una tarjeta nueva donde nos aparecerá los datos de nuestro usuario Administrador.

En este caso nos tendremos que fijar en el apartado “Administración” en el cuál veremos que pone Ajustes, pero en este punto no nos dejara pinchar en esta opción.

**Administrator**  
tomasmotasanchez14@gmail.com

Contacto relacionada: Administrator

Permisos de acceso: Preferencias

Acción: 1 / 1

**Administración**

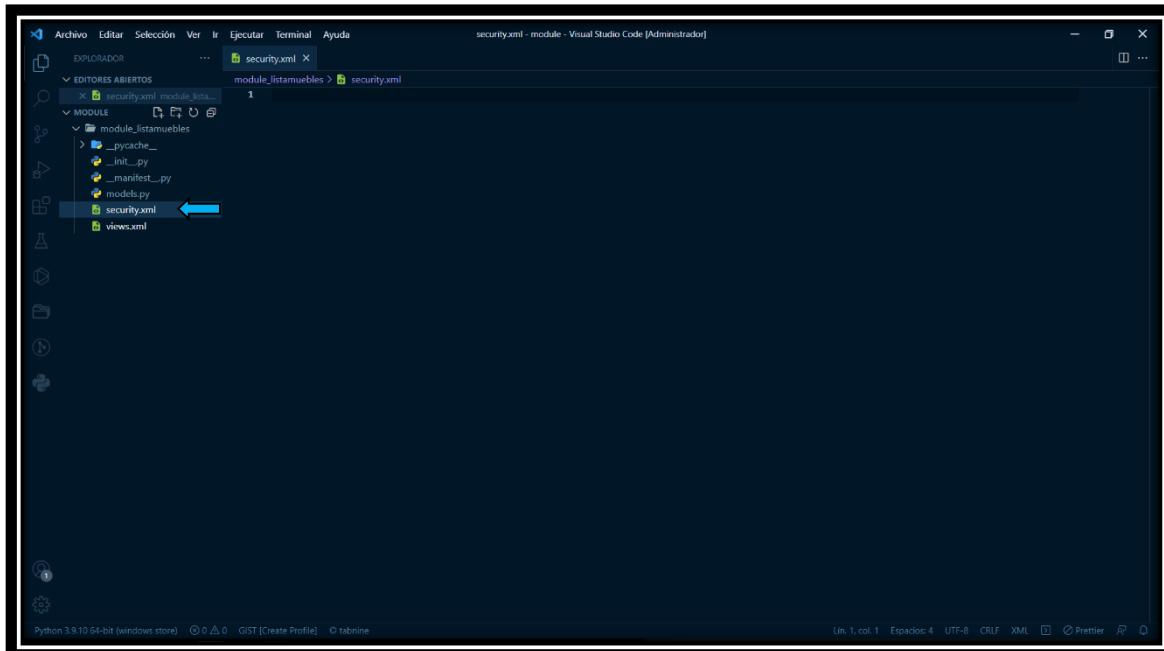
Administración: Ajustes

**Permisos extra**

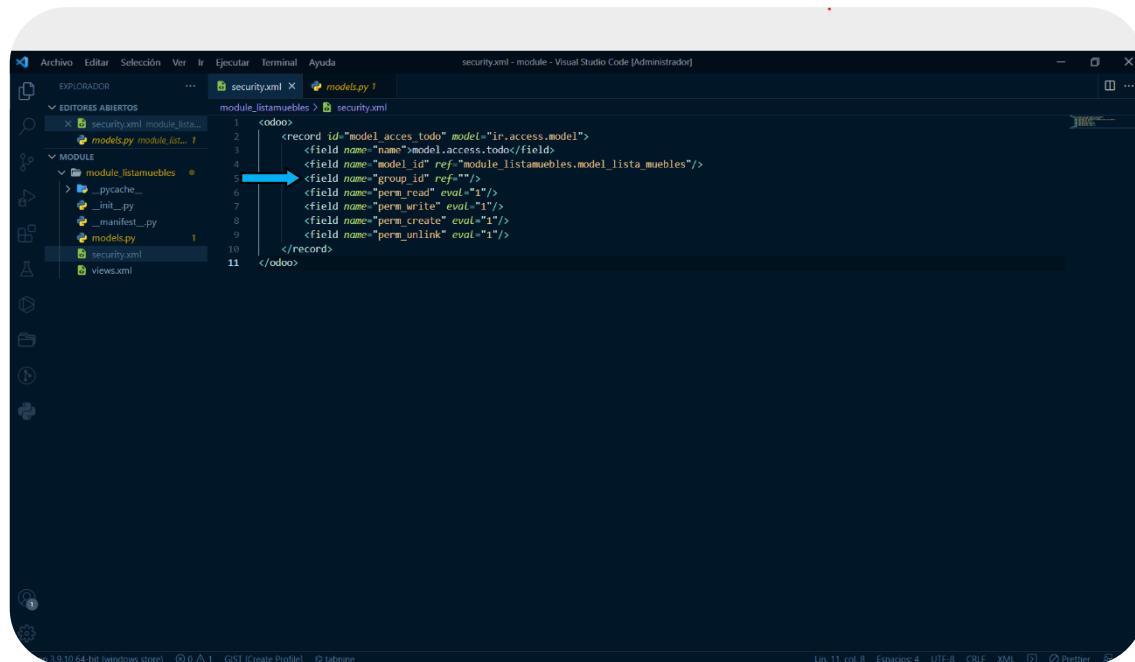
- Creación de contactos
- Multicompañía
- Multimoneda
- Otro

## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

Ahora nos tocará irnos a Visual Studio Code y crear el fichero “security.xml”.



En este fichero iremos desarrollando los parámetros de seguridad, menos un parámetro el cual es “group\_id” ya que tendremos que buscarlo en nuestros identificadores externos de nuestro sistema local de Odoo.



Tras parar el desarrollo final de nuestro fichero “security.xml”, tendremos que volver a los Ajustes de nuestro sistema local de Odoo.

The screenshot shows the Odoo web interface with the following details:

- Top Navigation Bar:** Shows 'Ajustes' (selected), 'Usuarios y compañías', 'Traducciones', and 'Técnico'.
- Breadcrumb:** 'Aplicaciones / Ajustes' (highlighted by a purple arrow).
- Content Area:** A table listing user information. One row is selected, showing:
 

Nombre	Usuario	Idioma	Última autenticación
Administrator	tomasmotasanchez14@gmail.com	Spanish / Español	16/02/2022 11:39:07
- Page Footer:** 'localhost:8069/web#menu\_id=4'

Dentro de los Ajustes, le daremos al botón “Técnico” que se encuentra en nuestra barra de herramientas superior, el cual se abrirá una lista desplegable donde tendremos que seleccionar la opción “Identificadores externos”.

The screenshot shows the Odoo web interface with the following details:

- Top Navigation Bar:** Shows 'Ajustes' (selected), 'Usuarios y compañías', 'Traducciones', and 'Técnico' (highlighted by a purple arrow).
- Content Area:** A sidebar menu under 'Ajustes' with various technical options. One item is highlighted with a purple arrow:
  - Identificadores externos
- Right Panel:** Shows a list of users with one row selected:
 

Idioma	Última autenticación
Spanish / Español	16/02/2022 11:39:07
- Page Footer:** 'localhost:8069/web#menu\_id=35&action=19'

Ahora que estamos en el apartado de “Identificadores externos”, tendremos que agrupar estos identificadores por objetos, ya que tendremos que acortar la búsqueda para encontrar la referencia exacta para el parámetro que nos hace falta para terminar de desarrollar el fichero que estamos viendo en este punto.

## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

ID	Nombre mostrado	Módulo	ID de registro
1	Id. completo		3
2	base.action_open_website	Menú de inicio	37
3	base.action_theme_store	Tienda de temas	9
4	base.action_third_party	Aplicaciones de terceros	15
5	base.act_ir_actions_todo_form	Asistentes de configuración	69
6	base.act_menu_create	Crear menú	23
7	base.act_view_currency_rates	Tasas monetarias	50
8	base.action_attachment	Adjuntos	66
9	base.action_base_document_layout_configurator	Configurar el diseño de tu documento	67
10	base.action_country	Países	68
11	base.action_country_group	Grupo de países	71
12	base.action_country_state	Provincias	70
13	base.action_currency_all_form	Monedas	4
14	base.action_currency_form	Monedas	14
15	base.action_decimal_precision_form	Precisión decimal	20
16	base.action_ir_mail_server_list	Servidores de correo saliente	19
17	base.action_model_constraint	Restricciones del modelo	17
18	base.action_model_data	Identificadores externos	18
19	base.action_model_fields	Campos	16
20	base.action_model_fields_selection	Selección de campos	
21	base.action_model_model	Modelos	
localhost:8069/web#			

Tras agrupar la búsqueda por Objeto, encontraremos un apartado que será por grupos. En este veremos una cosa que nos resulta familiar que es la de Administración/Ajustes que lo vimos en nuestro perfil Administrador.

Al ver esto ya sabemos que esto era el parámetro que nos faltaba para terminar de desarrollar nuestro fichero "security.xml".

ID	Nombre mostrado	Módulo	ID de registro
1	res.company (1)		31.375
2	res.country (250)		957.036
3	res.country.group (2)		14.535
4	res.country.state (1383)		55
5	res.currency (170)		2
6	res.groups (10)	Administración / Permisos de acceso	4
7	base.group_erp_manager	Permisos extra / Multicompañía	5
8	base.group_multi_company	Permisos extra / Multimonedas	6
9	base.group_multi_currency	Permisos extra / Características técnicas	7
10	base.group_no_one	Permisos extra / Creación de contactos	8
11	base.group_partner_manager	Tipos de Usuario / Portal	10
12	base.group_portal	Acceso a direcciones privadas	9
13	base.group_private_addresses	Tipos de Usuario / Público	3
14	base.group_public	Administración / Ajustes	1
15	base.group_system	Tipos de Usuario / Usuarios	
16	base.group_user	Administración / Ajustes	
17	res.lang (85)		3.655
18	res.partner (9)		30
19	res.partner.industry (21)		231

Nombre	Correo electrónico	Grupos	Permisos de acceso	Reglas de red
Administrator	tomasmot Sanchez14@gmail.com	6	87	8

En esta imagen podemos ver para que sirve este identificador, este será el identificador de grupo que nos faltaba.

Módulo	base	Nombre mostrado	Administración / Ajustes
Identificador externo	group_system	Nombre del modelo	res.groups
No actualizable		ID de registro	3
Fecha de actualización	16/02/2022 11:38:37	Registro	Administración / Ajustes
Fecha inicial	16/02/2022 11:38:37		

Ahora si volvemos a Visual Studio Code. En este ya podremos añadir este parámetro que nos faltaba y podemos finalizar el desarrollo de esta.

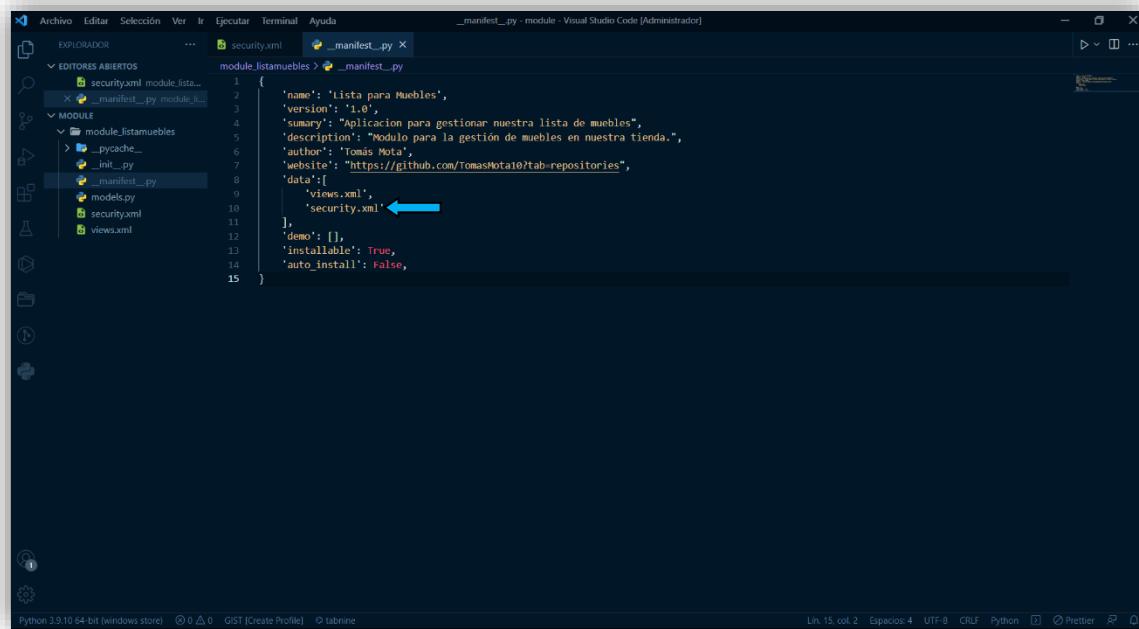
```

<odo>
  <record id="model_acces_todo" model="ir.model.access">
    <field name="name">model.acces.todo</field>
    <field name="model_id" ref="module_listamuebles.module_lista_muebles"/>
    <field name="group_id" ref="base.group_system"/> ←
    <field name="perm_read" eval="1"/>
    <field name="perm_write" eval="1"/>
    <field name="perm_create" eval="1"/>
    <field name="perm_unlink" eval="1"/>
  </record>
</odo>

```

Para finalizar comenzaremos editando el archivo `__manifest__.py` para declarar este nuevo archivo de datos, en nuestro caso llamaremos al archivo anteriormente desarrollado.

## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM



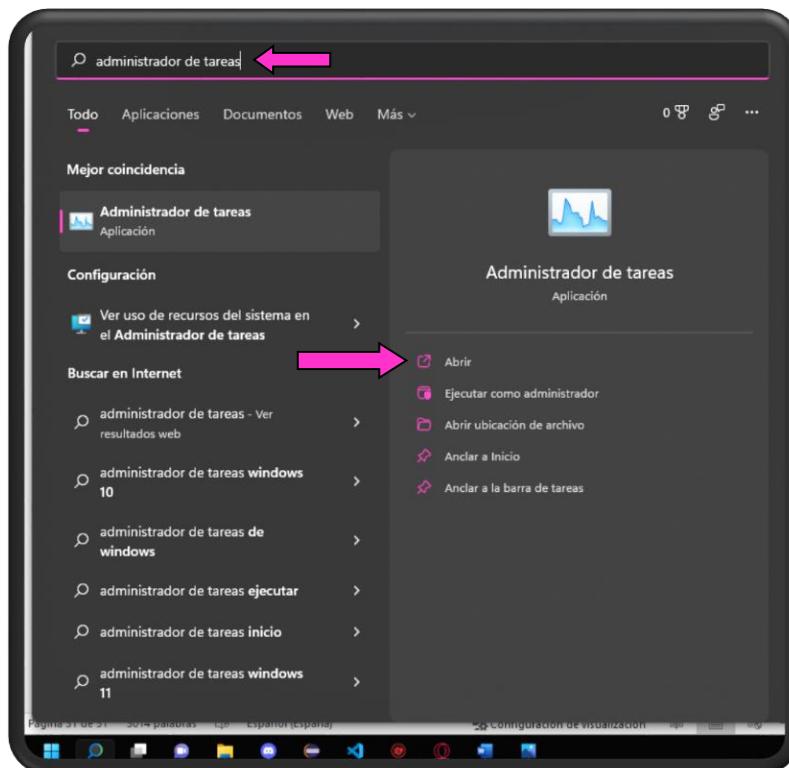
```
1 {
2     'name': 'Lista para Muebles',
3     'version': '1.0',
4     'summary': 'Aplicación para gestionar nuestra lista de muebles',
5     'description': 'Módulo para la gestión de muebles en nuestra tienda.',
6     'author': 'Tomas Mota',
7     'website': "https://github.com/TomasMota10?tab=repositories",
8     'data': [
9         'views.xml',
10        'security.xml' ←
11    ],
12    'demo': [],
13    'installable': True,
14    'auto_install': False,
15 }
```

Ya hemos finalizado último paso del desarrollo del módulo, en los próximos pasos veremos el funcionamiento de nuestro módulo y los usos que puede tener.

### 4.7. Séptimo paso -> Reinicio de nuestro servidor local de Odoo y actualización de nuestro módulo:

Al finalizar ya nuestro desarrollo de programación de nuestro módulo de Odoo, tendremos que reiniciar nuestro servidor local de Odoo.

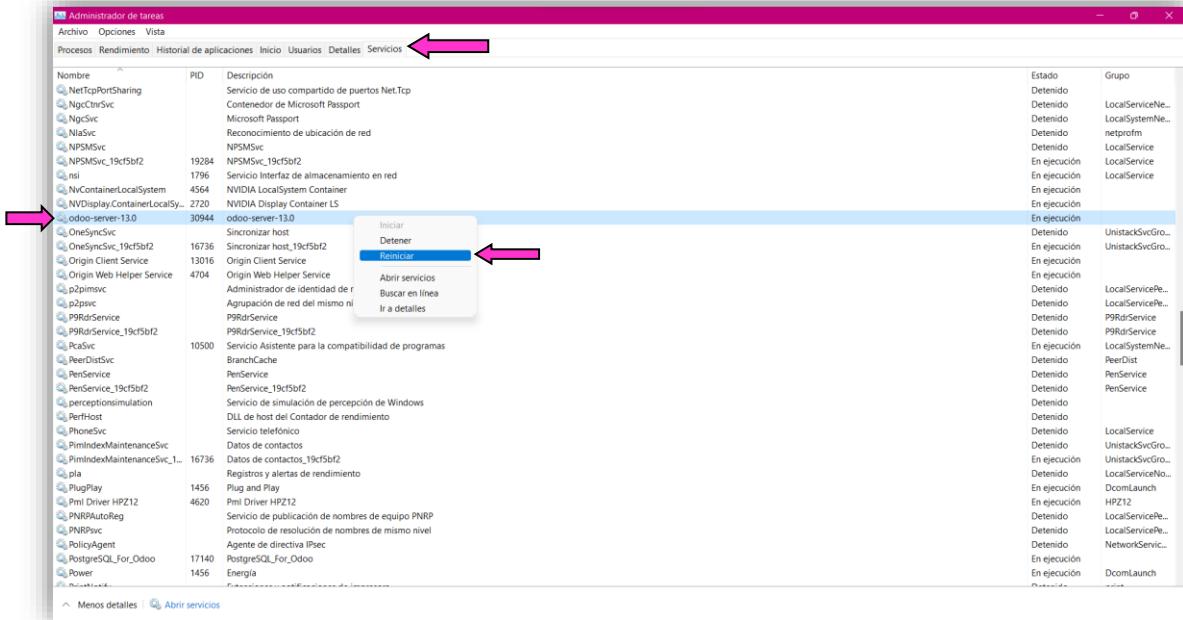
Para eso tendremos que irnos a nuestro sistema de Windows 11, buscar "Administrador de tareas" y le daremos a la opción "Abrir".



## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

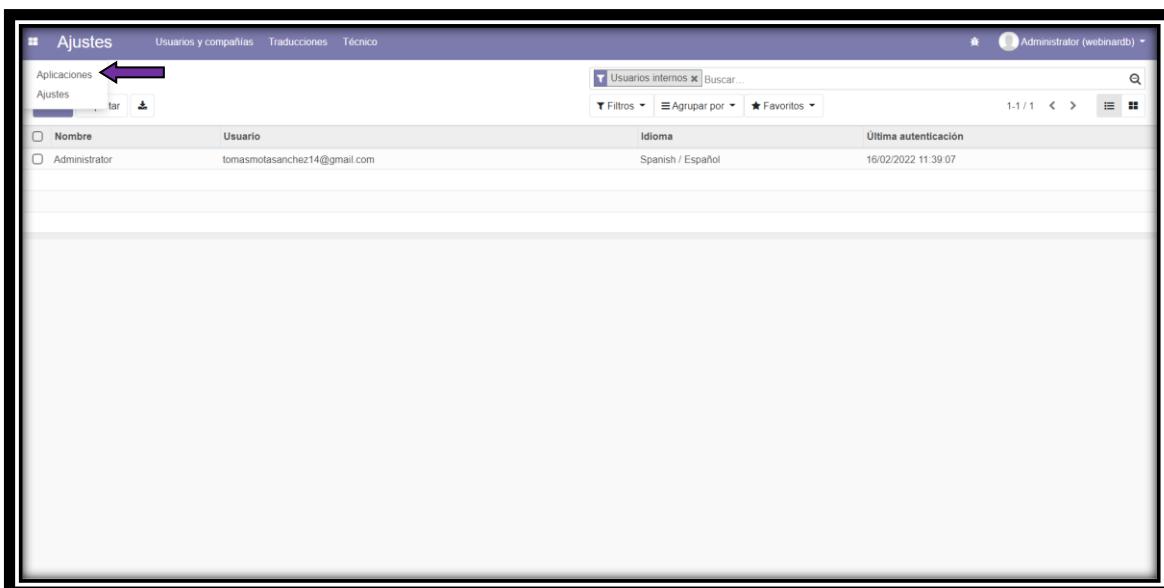
Una vez ya en la ventana de administrador de tareas, tendremos que irnos a la pestaña “Servicios” y dentro de esta ventana buscaremos nuestro servidor de “odoo-server-13.0”.

Cuando encontramos este servicio le daremos a click izquierdo y la opción “Reiniciar”.



Tras reiniciar el servidor y ver que se ha iniciado, nos iremos a nuestro sistema local de Odoo.

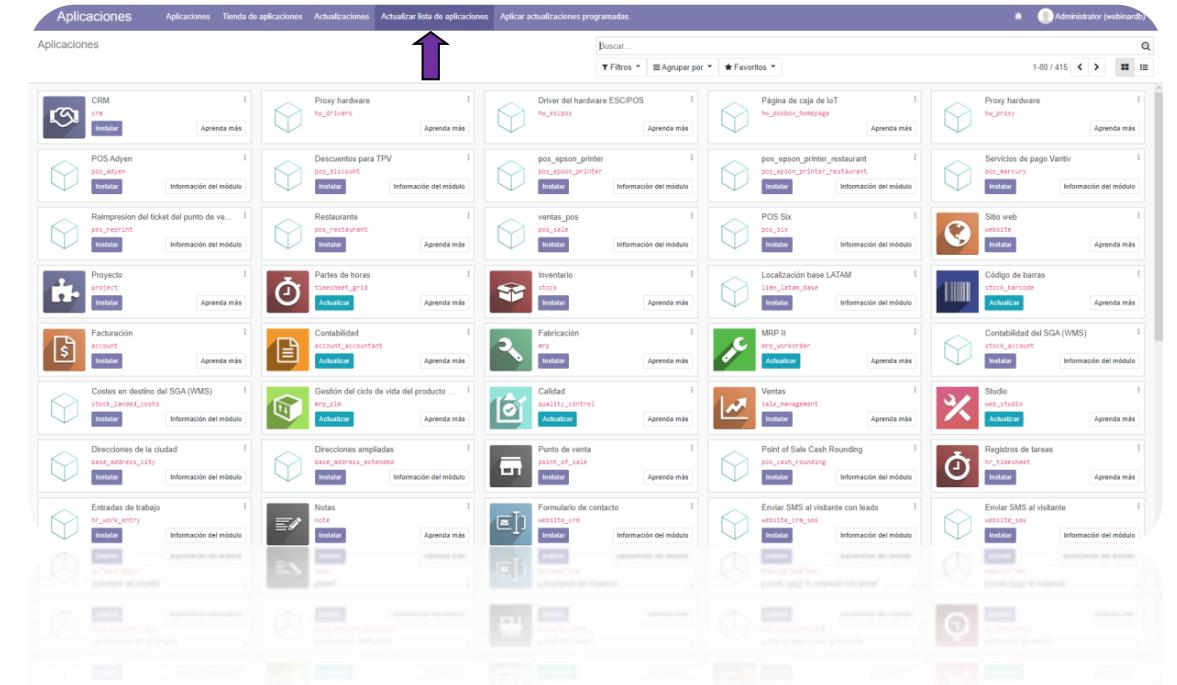
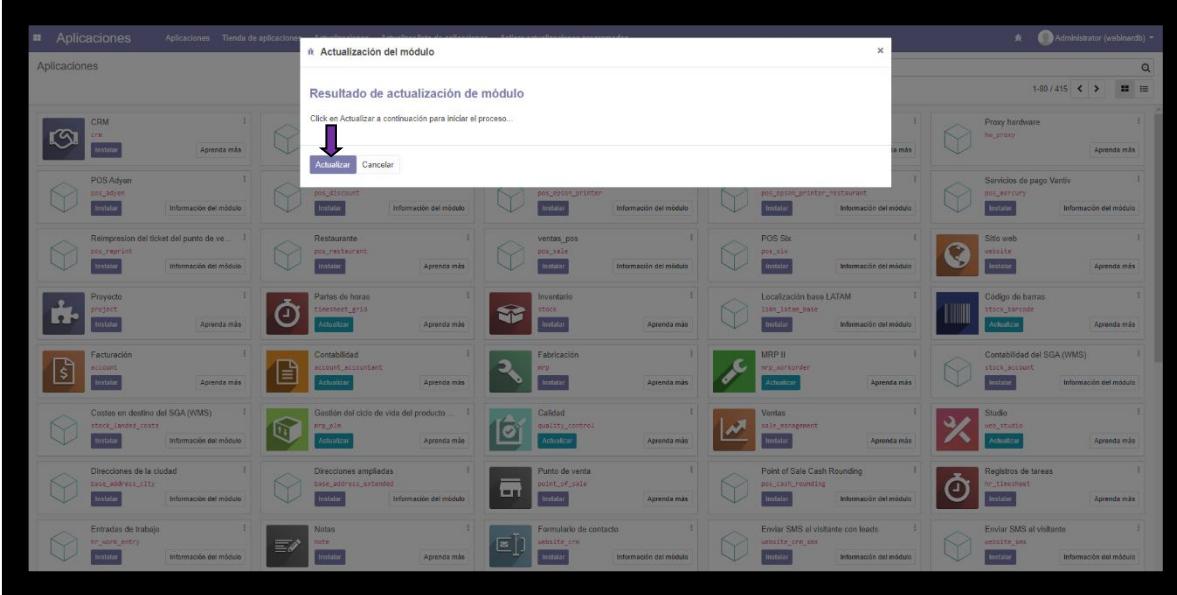
Dentro del sistemas le daremos a la opción “Aplicaciones”, la cuál nos llevara a la tienda de aplicaciones de nuestro sistema de Odoo.



## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

Dentro de la tienda de aplicaciones, le daremos a la opción que se encuentra en la barra de tareas “Actualizar lista de aplicaciones”.

Esto lo que provoca es que salga un mensaje donde nos dirá si queremos actualizar esta lista, en nuestro caso le daremos al botón Actualizar.

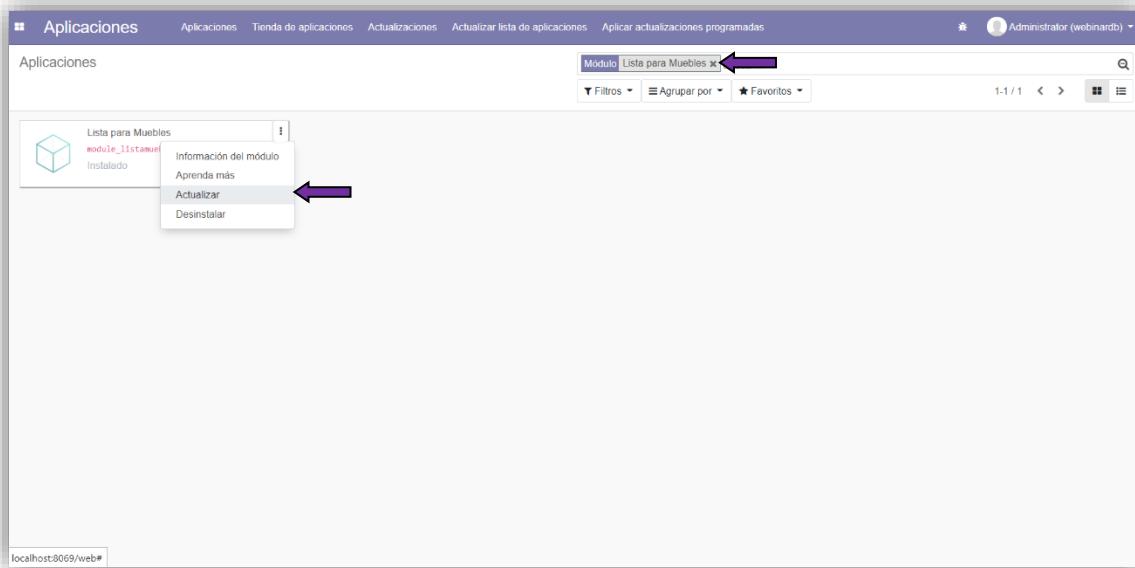



The screenshots show the Odoo application store interface. In the top navigation bar, there is a blue button labeled "Actualizar lista de aplicaciones". A purple arrow points from the text above to this button. Below the navigation bar, there is a search bar and some filters. The main area displays a grid of application modules, each with a thumbnail, name, and status (e.g., "Instalar" or "Aprende más"). A second purple arrow points from the text below to the "Actualizar" button in a modal dialog box titled "Actualización del módulo". The dialog box contains the message "Resultado de actualización de módulo" and the instruction "Click en Actualizar a continuación para iniciar el proceso...". It has two buttons: "Actualizar" (highlighted with a purple arrow) and "Cancelar".

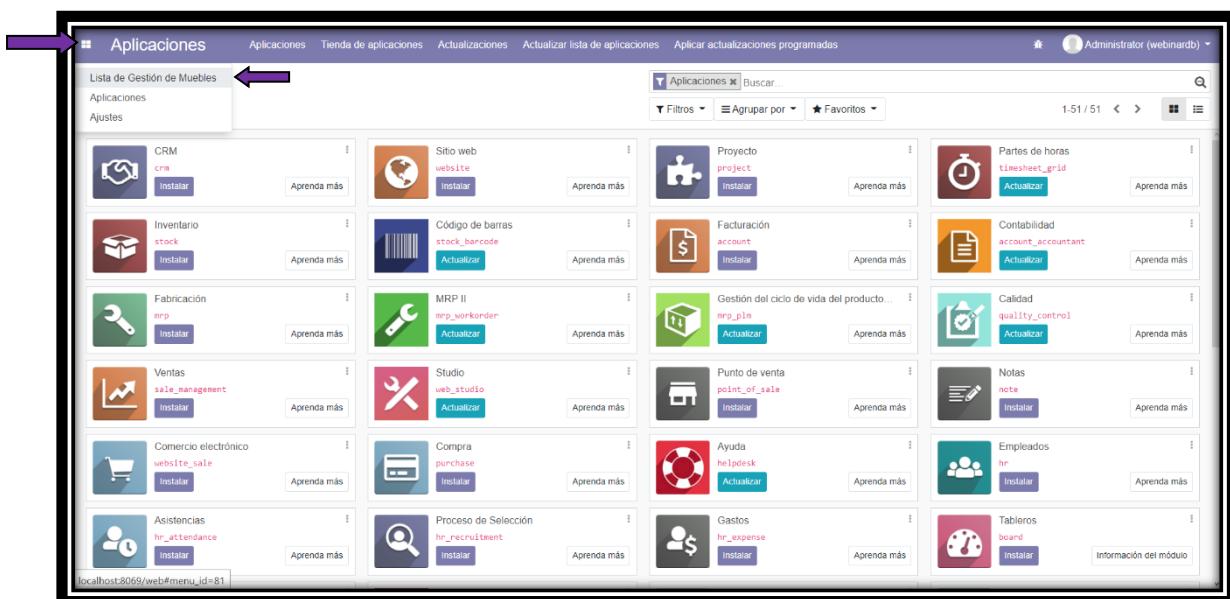
Al actualizarse la lista, iremos al buscador y volveremos a escribir el nombre que le hemos dado a nuestro modulo.

En la siguiente imagen podremos ver que el resultado es bueno, el módulo se encuentra ahora mismo en nuestro sistema de Odoo y lo que tendremos que hacer es actualizar nuestro propio módulo.

## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM



Como punto final, al actualizar nuestro módulo volveremos a la tienda de aplicaciones y si le damos en el botón que esta a la izquierda nos saldrá en la lista desplegable nuestro módulo de manera correcta.



## 5. Prueba de nuestro módulo:

Después de finalizar el desarrollo y la implementación de nuestro módulo en nuestro sistema local de Odoo, veremos la funcionalidad y rendimiento de este.



### 5.1. Lista de Gestión de Muebles:

En la entrada de nuestro módulo, lo primero que veremos es la muestra se un listado para la gestión de muebles.

El cuál se podrá crear un producto, editar o eliminar un producto.

En la siguiente imagen se podrá ver el listado de la gestión de muebles:

Nombre del mueble	Descripción del mueble	Categoría	Productstatus	Empresa de Origen	Precio de Venta

### 5.2. Añadir un mueble a nuestra lista:

Para añadir un registro simplemente pulsamos en el botón de Crear y se nos mostrará el formulario para añadir el nuevo producto.

## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

Lista de Gestión de Muebles

Lista de Gestión de Muebles

Crear Importar

Nombre del mueble Descripción del mueble Categoría Productstatus Empresa de Origen Precio de Venta

Lista de Gestión de Muebles

Lista de Gestión de Muebles / Nuevo

Guardar Descartar

Nombre del mueble Descripción del mueble

Categoría Productstatus

Empresa de Origen Precio de Venta

Poco a poco se vera en las siguientes imágenes como hemos ido rellenando cada dato que nos pide en el formulario:

Lista de Gestión de Muebles

Lista de Gestión de Muebles / Nuevo

Guardar Descartar

Nombre del mueble: Nacon CH-300 Silla Gaming

Categoría: Sofás, Sillas, Mesas, Librerías, Camas

Descripción del mueble

Productstatus

Precio de Venta

## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

En esta imagen podremos ver como vamos rellenando la empresa de donde pertenece este producto y la descripción de este:

The screenshot shows a web-based application titled "Lista de Gestión de Muebles". A purple arrow points to the "Empresa de Origen" dropdown menu, which contains "PC Componentes S.L.". Another purple arrow points to the "Descripción del mueble" input field, which contains "Silla Gaming para juegos".

En esta foto podemos ver como en el ProductStatus se abrirá un desplegable donde podremos seleccionar el estado del producto. Una vez finalizado el formulario le daremos al botón de Guardar:

The screenshot shows the same application with the "ProductStatus" dropdown open. The options "Nuevo", "Segunda Mano", and "Reacondicionado" are visible, with "Nuevo" highlighted by a blue selection bar. A purple arrow points to the "Guardar" button at the top left of the form.

Y en la última foto podemos ver como el producto ha sido añadido con éxito a nuestra lista:

The screenshot shows a table view of the furniture list. The last row represents the newly added item: "Nacon CH-300 Silla Gaming" (Category: Sillas, Origin: PC Componentes S.L.), "Silla Gaming para juegos" (Description), and "Reacondicionado" (Status). The price is listed as "89,95 €".

Nombre del mueble	Categoría	Descripción del mueble	Precio de Venta
Nacon CH-300 Silla Gaming	Sillas	Silla Gaming para juegos	89,95 €
Empresa de Origen		Productstatus	
PC Componentes S.L.		Reacondicionado	

Aquí se puede ver como el producto se encuentra en la lista, en mi caso añadí un producto más:

Nombre del mueble	Descripción del mueble	Categoría	Productstatus	Empresa de Origen	Precio de Venta
Nacon CH-300 Silla Gaming	Silla Gaming para juegos.	Sillas	Reacondicionado	PC Componentes S.L	89,99 €
Rinconera con cama de tela DAYTONA	Sofá Reversible.	Sofás	Nuevo	IKEA	949,00 €

También podemos ver que estos dos productos han sido añadidos correctamente en pgAdmin III:

<b>[PK]</b> id	productstatus	create_uid	create_date	write_uid	write_date	nombre	descripcion	categoria	empresaorigen	precioventa
1	reacondicionado	2	2022-02-17 17:20:47.167501	2	2022-02-17 17:20:47.167501	Nacon CH-300 Silla Gaming	Silla Gaming para juegos.	sillas	PC Componentes S.L	89,99 €
2	nuevo	2	2022-02-17 17:23:10.483847	2	2022-02-17 17:23:10.483847	Rinconera con cama de tela DAYTONA	Sofá Reversible.	sofas	IKEA	949,00 €

### 5.3. Editar un mueble en nuestra lista:

Para editar un mueble simplemente debemos pinchar en una tarea de la lista y nos aparecerá lo siguiente:

Nombre del mueble	Nacon CH-300 Silla Gaming	Descripción del mueble	Silla Gaming para juegos.
Categoría	Sillas	Productstatus	Reacondicionado
Empresa de Origen	PC Componentes S.L.	Precio de Venta	89.99 €

Tras esto, se nos abre la ventana del formulario en mi caso cambiare el nombre del mueble y el precio de venta de este.

Una vez cambiado estos parámetros, le daremos a guardar:

Nombre del mueble	SetGaming CH-300 Silla Gaming	Descripción del mueble	Silla Gaming para juegos.
Categoría	Sillas	Productstatus	Reacondicionado
Empresa de Origen	PC Componentes S.L.	Precio de Venta	109.99 €

Y en la última foto podemos ver como el producto ha sido editado con éxito a nuestra lista:

## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

The screenshot shows a CRM interface titled "Lista de Gestión de Muebles". It displays a single product record:

Nombre del mueble	SetGaming CH-300 Silla Gaming	Descripción del mueble	Silla Gaming para juegos.
Categoría	Sillas	Productstatus	Reacondicionado
Empresa de Origen	PC Componentes S.L.	Precio de Venta	109.99 €

Aquí se puede ver como el producto se encuentra totalmente cambiado en la lista:

The screenshot shows a list of furniture items in the CRM:

Nombre del mueble	Descripción del mueble	Categoría	Productstatus	Empresa de Origen	Precio de Venta
SetGaming CH-300 Silla Gaming	Silla Gaming para juegos.	Sillas	Reacondicionado	PC Componentes S.L.	109,99 €
Rinconera con cama de tela DAYTONA	Sofá Reversible.	Sofás	Nuevo	IKEA	949,00 €

También podemos ver que este producto ha sido actualizado correctamente en pgAdmin III:

The screenshot shows the "Edit Data" window in pgAdmin III displaying the "public.lista\_muebles" table. The table has columns: id, productstatus, nombre, descripción, categoría, empresaorigen, and precioventa. Two rows are shown:

1	reacondicionado	SetGaming CH-300 Silla Gaming	Silla Gaming para juegos.	silla	PC Componentes S.L.	109,99 €
2	nuevo	Rinconera con cama de tela DAYTONA	Sofá Reversible.	sofas	IKEA	949,00 €

### 5.4. Eliminar un mueble de nuestra lista:

Para eliminar un mueble simplemente debemos seleccionar un objeto, se nos mostrará en la parte superior del CRM un botón llamado acción y le daremos al botón "Suprimir".

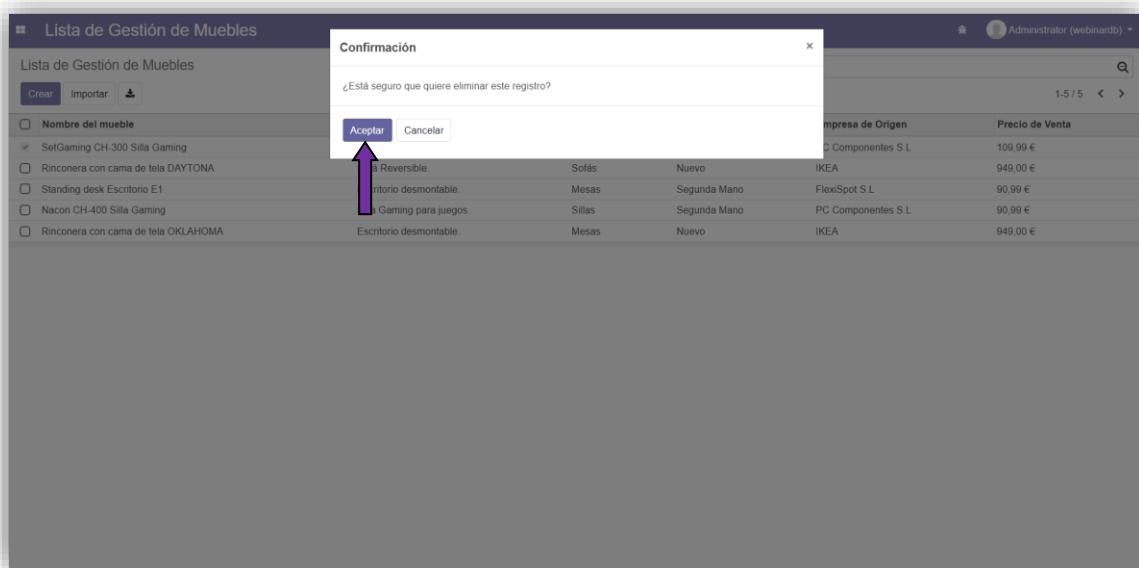
The screenshot shows the CRM interface with a context menu open over a selected furniture item. The menu includes options like "Exportar" and "Suprimir". A purple arrow points from the text above to this menu.

The list of furniture items includes the SetGaming CH-300 Silla Gaming chair, which was previously updated.

## TRABAJO U.D.5: PROGRAMACIÓN DE COMPONENTES DE SISTEMAS ERP/CRM

Al pulsar la opción de suprimir, nos mostrará el siguiente cuadro de diálogo.

En este caso le daremos al botón “Aceptar” que eliminará y nos llevará de vuelta a la lista, pero con dicho registro ya eliminado:



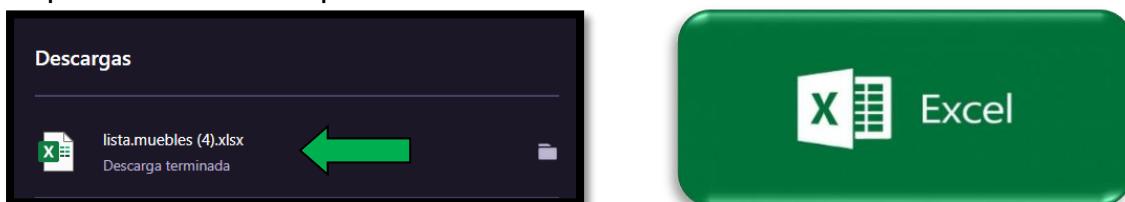
También podemos ver que este producto ha sido eliminado correctamente en pgAdmin III:

## 5.5. Exportación de nuestra lista a Excel:

Para exportar nuestra lista de muebles en formato Excel, tendremos que dirigirnos a nuestra lista y pinchamos en la siguiente opción de descarga:

Nombre del mueble	Descripción del mueble	Categoría	Productstatus	Empresa de Origen	Precio de Venta
Rinconera con cama de tela DAYTONA	Sofá Reversible.	Sofás	Nuevo	IKEA	949,00 €
Standing desk Escritorio E1	Escritorio desmontable.	Mesas	Segunda Mano	FlexiSpot S.L	90,99 €
Nacon CH-400 Silla Gaming	Silla Gaming para juegos.	Sillas	Segunda Mano	PC Componentes S.L	90,99 €
Rinconera con cama de tela OKLAHOMA	Escritorio desmontable.	Mesas	Nuevo	IKEA	949,00 €

Tras darle a este botón, lo que hará es descargar un archivo Excel en la esquina superior derecha de la pantalla:



Si pinchamos en el fichero que hemos descargado, se abrirá Excel y en el podremos ver nuestra lista de muebles correctamente:

Nombre del mueble	Descripción del mueble	Categoría	Productstatus	Empresa de Origen	Precio de Venta
Rinconera con cama de tela DAYTONA	Sofá Reversible.	Sofás	Nuevo	IKEA	949,00 €
Standing desk Escritorio E1	Escritorio desmontable.	Mesas	Segunda Mano	FlexiSpot S.L	90,99 €
Nacon CH-400 Silla Gaming	Silla Gaming para juegos.	Sillas	Segunda Mano	PC Componentes S.L	90,99 €
Rinconera con cama de tela OKLAHOMA	Escritorio desmontable.	Mesas	Nuevo	IKEA	949,00 €

## 6. Síntesis personal de este proyecto:

Tomás Mota Sánchez: En este proyecto, hemos podido ver bastante sobre la creación de un módulo para la gestión de una web comercial que ODDO nos da, también la importancia sobre el desarrollo de este.

A lo primero empezamos con la idea de creación de nuestro módulo, la cual es bastante simple y fácil de comprender.

Tras esto, empezamos ya con la creación y desarrollo de nuestro modulo, añadiendo tanto la información sobre el módulo, creación del modelo de este y el desarrollo de la personalización de la vista de nuestro módulo. En el siguiente paso, añadimos un sistema de seguridad para que nuestro módulo se pueda utilizar correctamente.



También pude aprender a gestionar la herramienta pgAdmin III y el uso que se le da con la base de datos que creamos en nuestro sistema local de Odoo.

Una vez implementado ya estas cosas, se empezó a ver como podemos actualizar nuestro módulo mientras se iba desarrollando su programación a la par.

También me pude adentrar en el apartado de los ajustes del sistema local de Odoo, el cuál pude ver los apartados de modelos e identificadores externos.

A la hora de finalizar, contemple las funcionalidades que puedo hacer con mi módulo como puede ser el añadir, editar y eliminar un mueble. Además de también poder descargar nuestra lista de muebles en formato Excel.

Mi conclusión es que me ha parecido un proyecto muy interesante y funcional, el cual me ha ayudado a la programación de un módulo en Odoo para nuestra empresa u otra empresa de forma fácil y menos cansada.



7. Webgrafía utilizada en este proyecto:

---

<https://comunitea.com/los-modulos-de-odoo-que-te-sorprenderan-en-2020/>

[https://www.youtube.com/watch?v=v-Tcp\\_kun7s&t=1023s&ab\\_channel=EscuelaFULLSTACK](https://www.youtube.com/watch?v=v-Tcp_kun7s&t=1023s&ab_channel=EscuelaFULLSTACK)

<https://www.postgresql.org/ftp/pgadmin/pgadmin4/v6.5/windows/>

[http://siabuc.ucol.mx/sitio/Descargas/Documentos/S9\\_PgAdminIII.pdf](http://siabuc.ucol.mx/sitio/Descargas/Documentos/S9_PgAdminIII.pdf)

<https://pythondiario.com/2013/06/initpy-en-python-que-es-y-como-funciona.html>

[https://odoodevelopment.readthedocs.io/en/latest/dev/docs/\\_manifest\\_.py.html](https://odoodevelopment.readthedocs.io/en/latest/dev/docs/_manifest_.py.html)

<https://www.ibm.com/docs/es/was-nd/9.0.5?topic=wsscv5a-ws-securityxml-file-default-configuration-websphere-application-server-network-deployment>

<https://emperove.gitbooks.io/fundamentos-de-desarrollo-odoo-10/content/capitulo-6.html>