



DARKY'S GAMES

Gestor de videojuegos

Por Jordi Ros López



Índice

- 1. Objetivos del proyecto**
- 2. Módulo de gestión de videojuegos**
 - a. Creación del módulo videojuego
 - b. Creación de modelos
- 3. Creación de vistas**
- 4. Creación de acciones**
- 5. Creación de categorías**
- 6. Agradecimientos**



1. Objetivos del proyecto

La tienda online Darky's Games nos ha pedido que creamos un sistema para gestionar la compra de sus videojuegos y sus clientes.

Para ello nos ha indicado que un cliente puede comprar un videojuego, y a su vez un videojuego puede ser comprado por varios clientes. Aunque sí es cierto que un cliente puede regalar un juego a otro, a nosotros solo nos interesa almacenar la lista de juegos que tiene cada cliente de forma personal, por lo que no nos preocupamos por el hipotético caso de que alguien compre un juego para otra persona.

A la hora de efectuar la compra se guardará la fecha en la que se realizó la compra.

Se espera que de un videojuego se guarde su título, una descripción y el precio al que se vende.

También se nos ha pedido que de un cliente se guarde su nombre, el nickname que usa y un email.



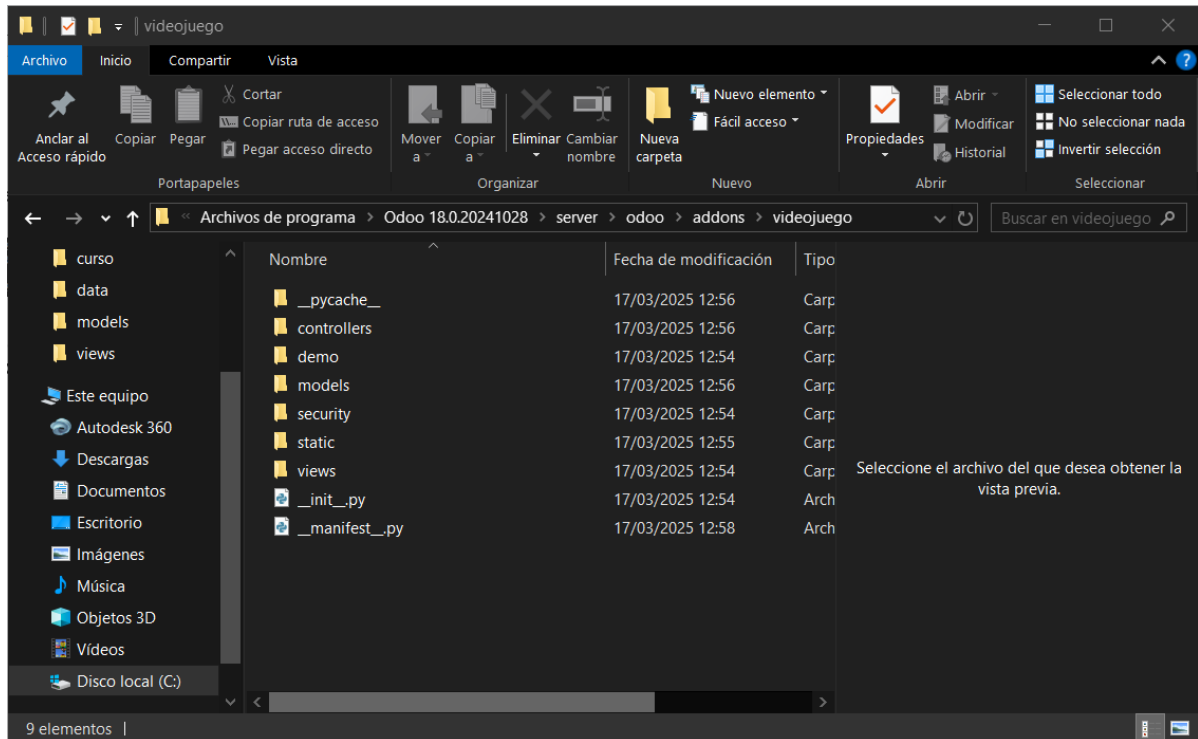
2. Módulo de gestión de videojuegos

a. Creación del módulo videojuego

Creamos la carpeta del módulo videojuego desde el terminal cmd con el siguiente comando:

```
c:\Program Files\Odoo 18.0.20241028\python>python.exe "c:\Program Files\Odoo 18.0.20241028\server\odoo-bin" scaffold Videojuego "c:\Program Files\Odoo 18.0.20241028\server\odoo\addons"  
c:\Program Files\Odoo 18.0.20241028\python>
```

Si nos dirigimos a la carpeta addons de Odoo veremos que se ha creado la carpeta de nuestro módulo con éxito.

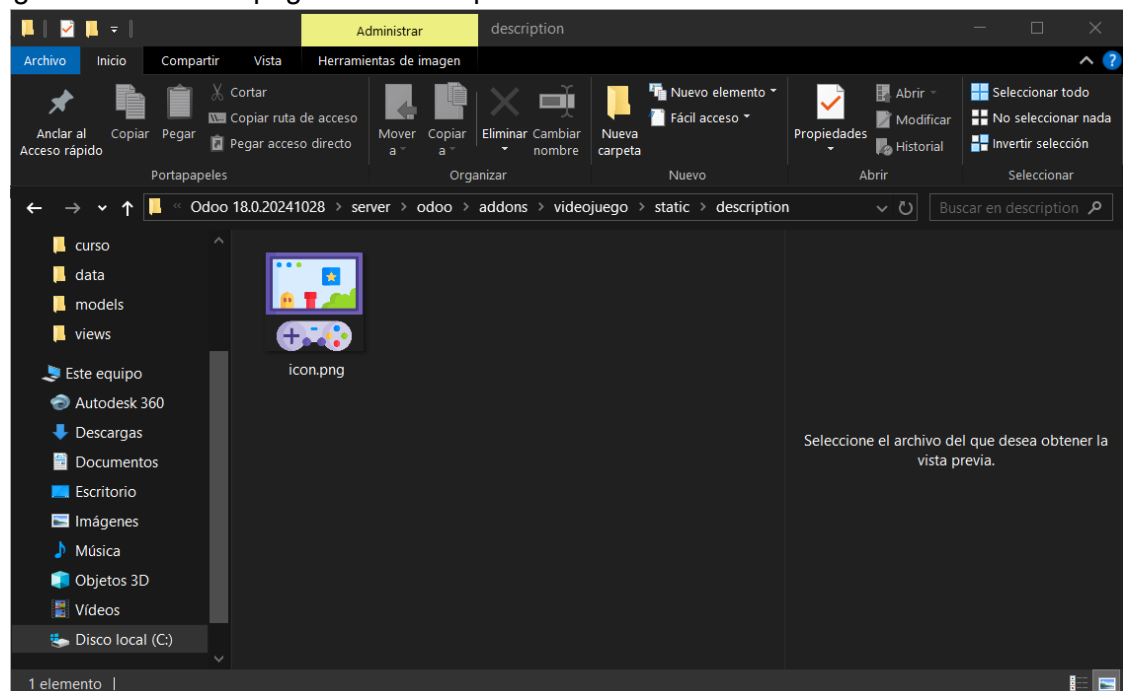


En el archivo `_manifest_` añadiremos estas dos líneas de código para convertir el módulo en una aplicación instalable, además de modificar algunos parámetros con la descripción o el nombre del autor (opcional).



```
1  # -*- coding: utf-8 -*-
2  {
3      'name': "Videojuego",
4      'summary': "Control de los videojuegos a los que se podran jugar.",
5      'description': """
6      Long description of module's purpose
7      """,
8      'author': "Jordi Ros López",
9      'website': "https://www.yourcompany.com",
10
11      # only loaded in demonstration mode
12      'demo': [
13          'demo/demo.xml',
14      ],
15      'installable': True,
16      'application': True,
17  }
```

En la carpeta del módulo añadiremos las carpetas “static/description” y dentro ponemos una imagen llamada “icon.png” de 100x100px como icono del módulo.



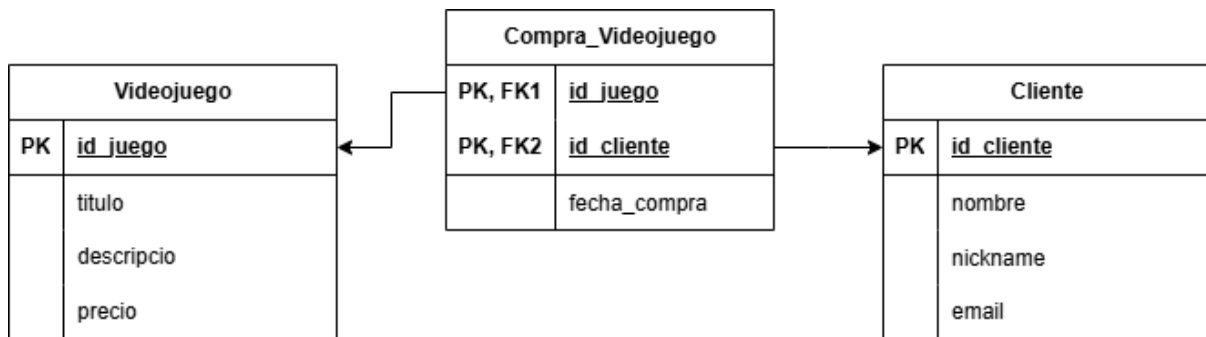
b. Creación de modelos

El módulo está compuesto por tres modelos: videojuego, cliente y compra_videojuego.

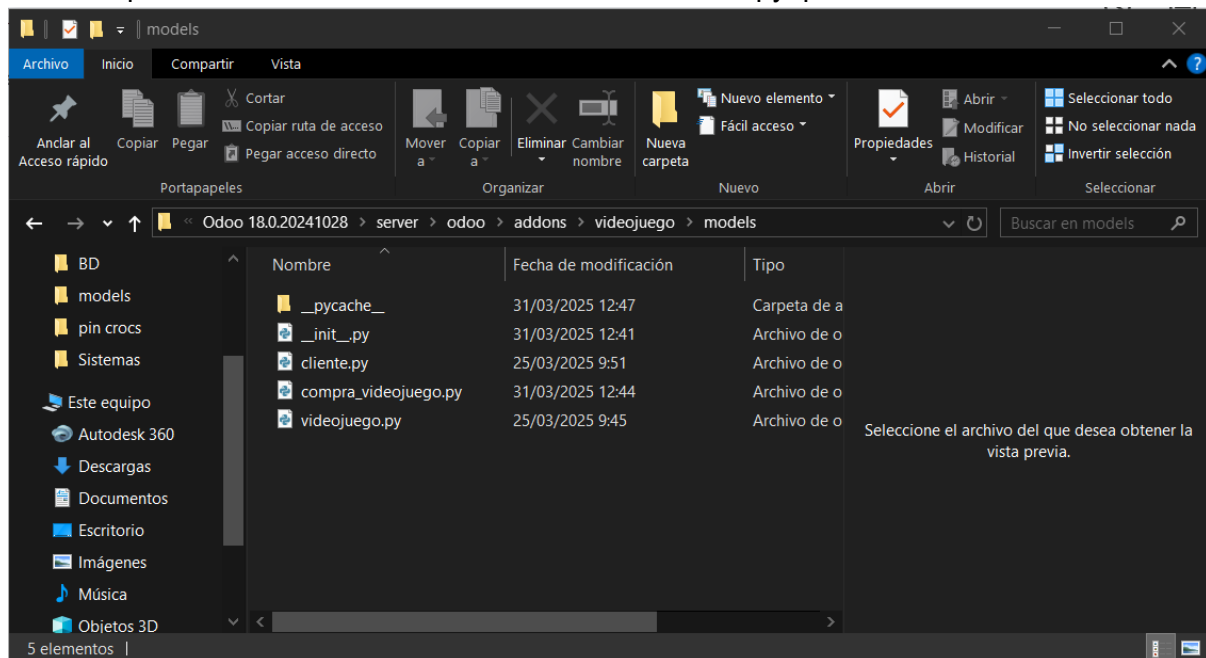
Un cliente puede comprar más de un videojuego y un videojuego a su vez puede ser comprado por más de una persona.



- Un videojuego está compuesto por una id como clave primaria, un título, una descripción y un precio.
- Un cliente está compuesto por una id como clave primaria, nombre, un nickname y un email.
- La clase compra_videojuego relaciona la id del videojuego y la id del cliente como claves primaria y guarda la fecha del día de la compra.



En la carpeta “models” del módulo creamos un archivo ‘.py’ para cada uno de los modelos:



Este es el código para crear el modelo videojuego:



```
# -*- coding: utf-8 -*-

from odoo import fields, models

class Videojuego(models.Model):
    _name = 'videojuego.videojuego'
    _description = 'Modelo de Videojuego'

    id_juego = fields.Integer(string="ID", required=True)
    titulo = fields.Char(string='Título', required=True)
    descripcion = fields.Text(string='Descripción')
    precio = fields.Float(string='Precio', required=True)

    _sql_constraints = [
        ('id_juego_unique', 'unique(id_juego)', 'La ID debe ser única.')
    ]
```

Este es el código del modelo cliente:

```
# -*- coding: utf-8 -*-

from odoo import fields, models

class Cliente(models.Model):
    _name = 'videojuego.cliente'
    _description = 'Modelo de Cliente'

    nombre = fields.Char(string="Nombre", required=True)
    nickname = fields.Char(string="Nickname", required=True)
    email = fields.Char(string="Correo Electrónico", required=True)

    _sql_constraints = [
        ('nombre_unique', 'unique(nombre)', 'El nombre debe ser único.')
    ]
```

Este es el código para el modelo compra_videojuego:

```
# -*- coding: utf-8 -*-

from odoo import fields, models

class Compra_Videojuego(models.Model):
    _name = 'videojuego.compra_videojuego'
    _description = 'Relación entre Videojuegos y Clientes'

    id_videojuego = fields.Many2one('videojuego.videojuego', string="ID_videojuego", required=True, ondelete="cascade")
    id_cliente = fields.Many2one('videojuego.cliente', string="ID_cliente", required=True, ondelete="cascade")
    fecha_compra = fields.Date(string="Fecha de Compra", required=True)
    titulo_videojuego = fields.Char(string="Título del Videojuego", related="id_videojuego.titulo", store=True)
    nombre_cliente = fields.Char(string="Nombre del Cliente", related="id_cliente.nombre", store=True)

    _sql_constraints = [
        ('compra_videojuego_unique', 'unique(id_videojuego, id_cliente)', 'Este cliente ya tiene este videojuego.')
    ]
```



En el documento “_init_.py” de la carpeta models importamos los modelos creados.

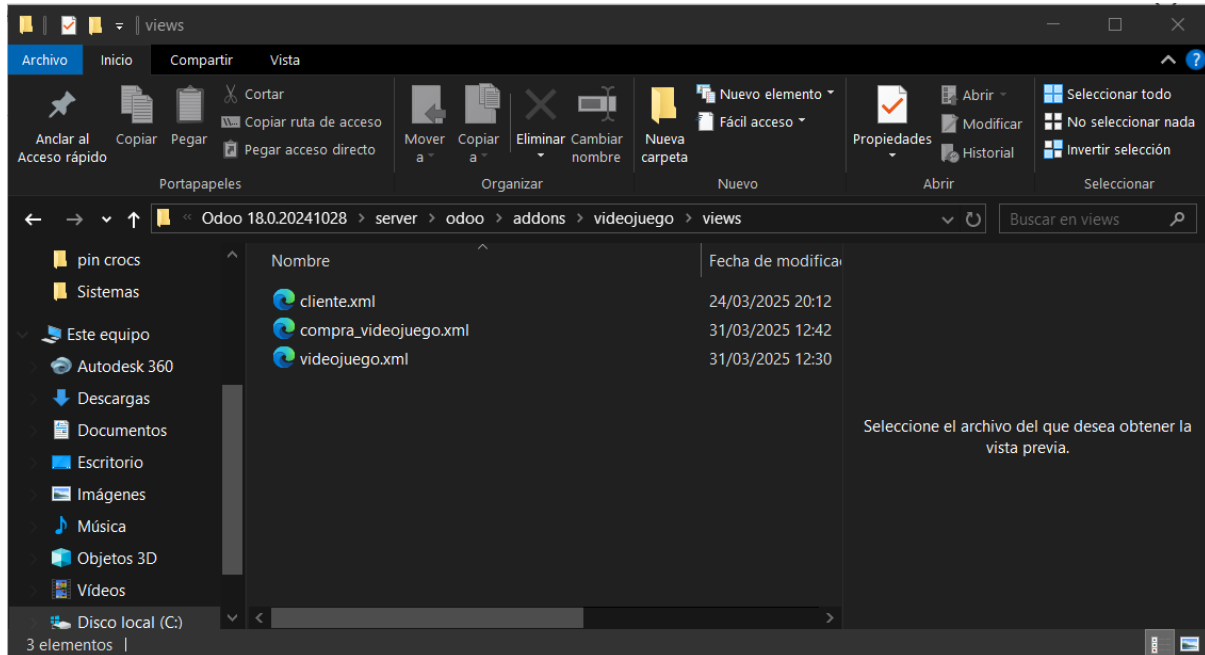
```
# -*- coding: utf-8 -*-  
  
from . import videojuego  
from . import cliente  
from . import compra_videojuego
```

Si revisamos en odoo podemos ver que se han cargado los modelos correctamente.



3. Creación de vistas

En la carpeta views crearemos un archivo “.xml” por cada modelo para generar sus respectivas vistas.



Este es el archivo para crear la vista árbol y la vista formulario del modelo videojuego:

```
<record id="view_videojuego_tree" model="ir.ui.view">
  <field name="name">videojuego.tree</field>
  <field name="model">videojuego.videojuego</field>
  <field name="arch" type="xml">
    <list string="Videojuegos">
      <field name="id_juego"/>
      <field name="titulo"/>
      <field name="descripcion"/>
      <field name="precio"/>
    </list>
  </field>
</record>
```

```
<record id="view_videojuego_form" model="ir.ui.view">
  <field name="name">videojuego.form</field>
  <field name="model">videojuego.videojuego</field>
  <field name="arch" type="xml">
    <form string="Videojuego">
      <sheet>
        <group>
          <field name="id_juego"/>
          <field name="titulo"/>
          <field name="descripcion"/>
          <field name="precio"/>
        </group>
      </sheet>
    </form>
  </field>
</record>
```

Vista árbol

Vista formulario

Este es el archivo para crear la vista árbol y la vista formulario del modelo cliente:



```
<record id="view_cliente_tree" model="ir.ui.view">
  <field name="name">cliente.tree</field>
  <field name="model">videojuego.cliente</field>
  <field name="arch" type="xml">
    <list string="Clientes">
      <field name="id_cliente"/>
      <field name="nombre"/>
      <field name="nickname"/>
      <field name="email"/>
    </list>
  </field>
</record>
```

Vista árbol

```
<record id="view_cliente_form" model="ir.ui.view">
  <field name="name">cliente.form</field>
  <field name="model">videojuego.cliente</field>
  <field name="arch" type="xml">
    <form string="Cliente">
      <sheet>
        <group>
          <field name="id_cliente"/>
          <field name="nombre"/>
          <field name="nickname"/>
          <field name="email"/>
        </group>
      </sheet>
    </form>
  </field>
</record>
```

Vista formulario

Este es el archivo para crear la vista árbol y la vista formulario del modelo compra:

```
<record id="view_compra_videojuego_tree" model="ir.ui.view">
  <field name="name">compra_videojuego.list</field>
  <field name="model">videojuego.compra_videojuego</field>
  <field name="arch" type="xml">
    <list string="Historial de Compras">
      <field name="titulo_videojuego"/>
      <field name="nombre_cliente"/>
      <field name="fecha_compra"/>
    </list>
  </field>
</record>
```

Vista árbol

```
<record id="view_compra_videojuego_form" model="ir.ui.view">
  <field name="name">compra_videojuego.form</field>
  <field name="model">videojuego.compra_videojuego</field>
  <field name="arch" type="xml">
    <form string="Registro de Compra">
      <sheet>
        <group>
          <field name="id_videojuego"/>
          <field name="id_cliente"/>
          <field name="fecha_compra"/>
        </group>
      </sheet>
    </form>
  </field>
</record>
```

Vista formulario

En el archivo “_manifest_.py” agregaremos este código para activar las vistas:

```
# always loaded
'data': [
    'views/videojuego.xml',
    'views/cliente.xml',
    'views/compra_videojuego.xml',
],
```

Finalmente podemos ver el menú de vistas del árbol y formulario para ver la lista de videojuegos y clientes, y los valores de cada videojuego y de cada cliente.



4. Creación de acciones

En cada archivo xml agregaremos estos códigos por cada vista generada

Este código se lo agregamos a la vista de videojuego:

```
<record id="videojuego_menu_action" model="ir.actions.act_window">
  <field name="name">Videojuegos</field>
  <field name="res_model">videojuego.videojuego</field>
  <field name="view_mode">list,form</field>
  <field name="domain">[]</field>
  <field name="help" type="html">
    <p class="oe_view_nocontent_create">Crea el primer videojuego.</p>
  </field>
</record>
<menuitem id="videojuego_menu" name="Videojuegos"/>
<menuitem id="videojuego_submenu" parent="videojuego_menu" name="Videojuegos" action="videojuego_menu_action"/>
```

Este código se lo agregamos a la vista de cliente:

```
<record id="cliente_menu_action" model="ir.actions.act_window">
  <field name="name">Clientes</field>
  <field name="res_model">videojuego.cliente</field>
  <field name="view_mode">list,form</field>
  <field name="domain">[]</field>
  <field name="help" type="html">
    <p class="oe_view_nocontent_create">Crear el primer cliente.</p>
  </field>
</record>
<menuitem id="cliente_menu" name="Clientes"/>
<menuitem id="cliente_submenu" parent="cliente_menu" name="Clientes" action="cliente_menu_action"/>
```

Este código se lo agregamos a la vista de compra_videojuego:

```
<record id="compra_videojuego_menu_action" model="ir.actions.act_window">
  <field name="name">Compras</field>
  <field name="res_model">videojuego.compra_videojuego</field>
  <field name="view_mode">list,form</field>
  <field name="domain">[]</field>
  <field name="help" type="html">
    <p class="oe_view_nocontent_create">Crea la primera compra.</p>
  </field>
</record>
<menuitem id="compra_videojuego_menu" name="Compras"/>
<menuitem id="compra_videojuego_submenu" parent="compra_videojuego_menu" name="Compras" action="compra_videojuego_menu_action"/>
```



Finalmente podemos hacer funcionar los modelos para almacenar videojuegos, clientes y realizar compras.

| <input type="checkbox"/> | ID | Título | Descripción | Precio |
|--------------------------|----|-----------------|--|--------|
| <input type="checkbox"/> | 0 | Portal 2 | ¡La «Iniciativa de Prueba Perpetua» se ha ampliado, permitiéndote ahora diseñar puzzles cooperativos para ti y tus amigos! | 9,75 |
| <input type="checkbox"/> | 1 | Bendy and ... | Bendy and the Ink Machine® is the first person puzzle action horror game that will forever ruin your childhood love of cartoons. | 18,49 |
| <input type="checkbox"/> | 2 | Bendy and ... | Bendy and the Dark Revival® is a first-person survival horror game set in the creepiest cartoon studio to ever exist. Discover the truth. Escape the studio. Above all, fear the Ink Demon...and survive. | 26,99 |
| <input type="checkbox"/> | 3 | Hello Neig... | Hello Neighbor es un juego de terror y sigilo donde debes meterte en la casa de tu vecino y averiguar los secretos que esconde en el sótano. Juega contra una IA avanzada que aprende de tus movimientos. ¿Te gusta escalar ventanas? Podrá una trampa para osos. ¿Tratas de escapar? Encontrará un atajo. | 28,99 |
| <input type="checkbox"/> | 4 | Little Night... | ¡Sumérgete en la enigmática atmósfera de Little Nightmares y enfréntate a los miedos de tu infancia! Ayuda a Seis a escapar de Las Fauces, un misterioso navío donde moran ánimas corrompidas en busca de su próxima comida... | 19,99 |
| <input type="checkbox"/> | 5 | Little Night... | Little Nightmares II es un juego de aventura y suspense en el que eres Mono, un joven atrapado en un mundo distorsionado por una malvada transmisión. Junto a su nueva amiga Six, Mono se dispone a descubrir de dónde viene esa transmisión. | 29,99 |
| <input type="checkbox"/> | 6 | Poppy Play... | You must stay alive in this horror/puzzle adventure. Try to survive the vengeful toys waiting for you in the abandoned toy factory. Use your GrabPack to hack electrical circuits or nab anything from afar. Explore the mysterious facility... and don't get caught. | 42,48 |
| <input type="checkbox"/> | 7 | Superliminal | La percepción es la realidad: este juego de rompecabezas en primera persona conseguirá dar la vuelta a tu forma de pensar. Escapa de un sueño surrealista resolviendo puzzles imposibles y usando la ambigüedad de la profundidad y la perspectiva. | 19,50 |

| <input type="checkbox"/> | ID | Nombre | Nickname | Correo Electrónico |
|--------------------------|----|--------------------------|-----------|----------------------|
| <input type="checkbox"/> | 0 | Jordi Ros López | Darky485 | jorgex1412@gmail.com |
| <input type="checkbox"/> | 2 | Juan Carlos Alonso | Femgeta | Juanito81@gmail.com |
| <input type="checkbox"/> | 3 | Fernando Alfonso Mumuera | Fernetero | Alfonso45@gmail.com |
| <input type="checkbox"/> | 1 | Noelia Pérez | Helloboys | NoePere@gmail.com |



Compras

localhost:8069/odoo/action-107

Fortnite.gg Minecraft Test autoescuela Tinkercad Pluto tv ¡Acepta el reto! Todos los favoritos

Compras Compras

Nuevo Compras 1-4 / 4

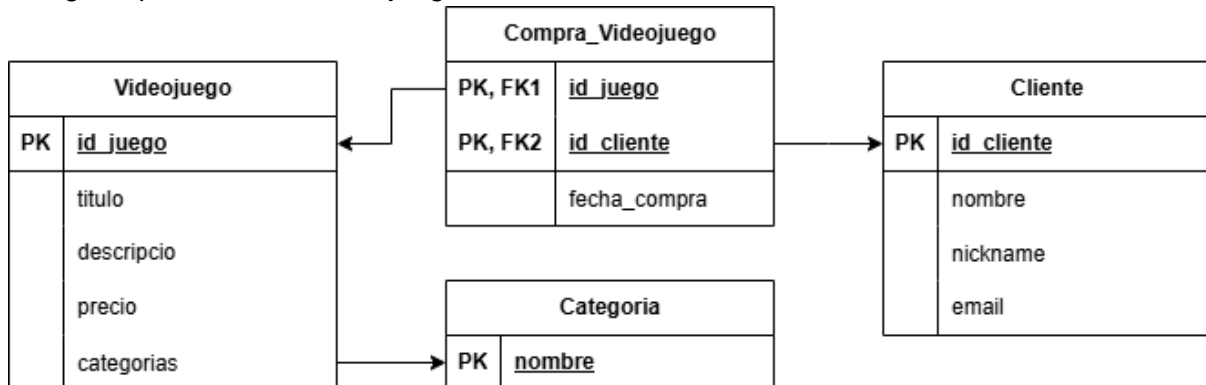
Buscar...

| <input type="checkbox"/> ID_videojuego | ID_cliente | Fecha de Compra |
|---|----------------------|-----------------|
| <input type="checkbox"/> videojuego.videojuego,6 | videojuego.cliente,1 | 31/03/2025 |
| <input type="checkbox"/> videojuego.videojuego,10 | videojuego.cliente,4 | 20/03/2025 |
| <input type="checkbox"/> videojuego.videojuego,8 | videojuego.cliente,3 | 11/03/2025 |
| <input type="checkbox"/> videojuego.videojuego,8 | videojuego.cliente,4 | 13/03/2025 |

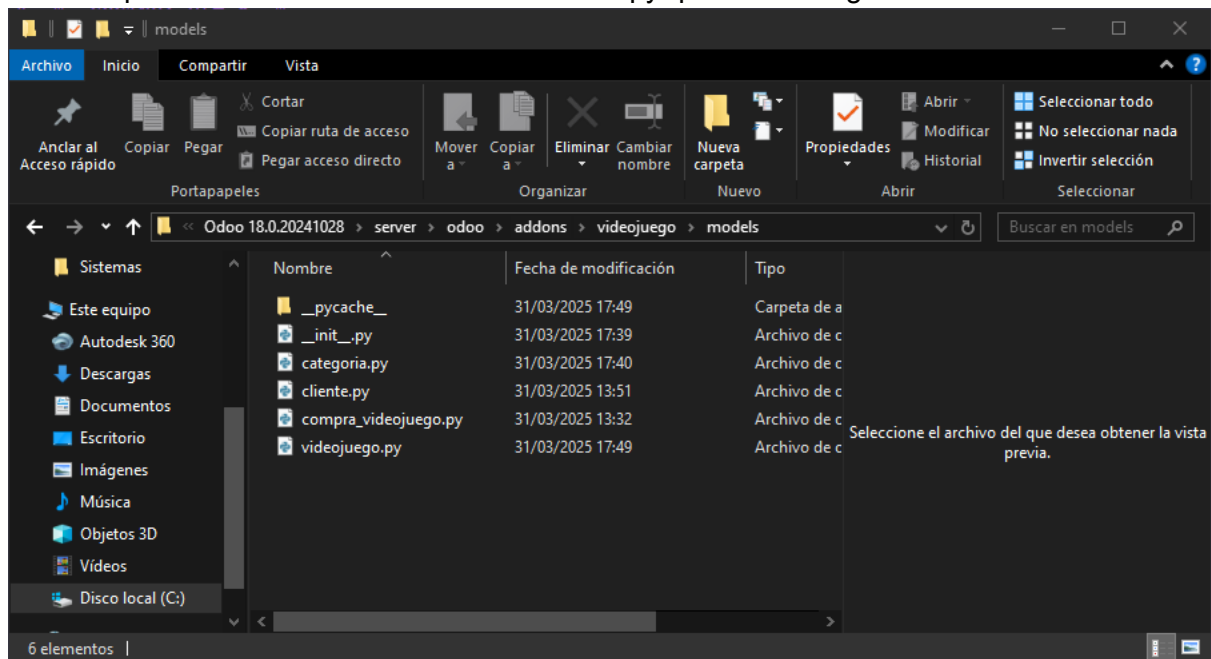


5. Creación de categorías

Como añadido he añadido en el modelo videojuego un atributo multiple llamado categoría, para definir qué tipo de juego es. Un videojuego puede tener varias categorías, y una categoría puede tener varios juegos.



En la carpeta de models añadimos un archivo “.py” para las categorías.



En este archivo añadimos este código para crear la clase categoría.



```
# -*- coding: utf-8 -*-

from odoo import fields, models

class Categoria(models.Model):
    _name = 'videojuego.categoria'
    _description = 'Modelo de Categoria'

    nombre = fields.Char(string='Nombre', required=True)

    _sql_constraints = [
        ('id_categoria_unique', 'unique(nombre)', 'La categoria debe ser única.')
    ]
```

Este es el código de categoría para las vistas de árbol y formulario.

```
<record id="view_categoria_tree" model="ir.ui.view">
  <field name="name">categoria.tree</field>
  <field name="model">videojuego.categoria</field>
  <field name="arch" type="xml">
    <list string="Categorias">
      <field name="nombre"/>
    </list>
  </field>
</record>
```

```
<record id="view_categoria_form" model="ir.ui.view">
  <field name="name">categoria.form</field>
  <field name="model">videojuego.categoria</field>
  <field name="arch" type="xml">
    <form string="Categorias">
      <sheet>
        <group>
          <field name="nombre"/>
        </group>
      </sheet>
    </form>
  </field>
</record>
```

Vista árbol

Vista formulario

Finalmente este es el código para generar el menú de categoría.

```
<menuitem id="categoria_menu" name="Categoria"/>
<menuitem id="categoria_submenu" parent="categoria_menu" name="Categoria" action="categoria_menu_action"/>
```

En el archivo “.py” de videojuego añadimos el atributo categoría de esta forma:

```
# -*- coding: utf-8 -*-

from odoo import fields, models

class Videojuego(models.Model):
    _name = 'videojuego.videojuego'
    _description = 'Modelo de Videojuego'

    id_juego = fields.Integer(string="ID", required=True)
    titulo = fields.Char(string='Título', required=True)
    descripcion = fields.Text(string='Descripción')
    categoria_id = fields.Many2many('videojuego.categoria', string="Categorias", required=True)
    precio = fields.Float(string='Precio (€)', required=True)

    _sql_constraints = [
        ('id_juego_unique', 'unique(id_juego)', 'La ID debe ser única.')
    ]
```



Ahora en la vista de árbol y formulario de videojuego añadimos el atributo categoría:

```
<record id="view_videojuego_tree" model="ir.ui.view">
  <field name="name">videojuego.tree</field>
  <field name="model">videojuego.videojuego</field>
  <field name="arch" type="xml">
    <list string="Videojuegos">
      <field name="id_juego"/>
      <field name="titulo"/>
      <field name="descripcion"/>
      <field name="categoria_id"/>
      <field name="precio"/>
    </list>
  </field>
</record>
<record id="view_videojuego_form" model="ir.ui.view">
  <field name="name">videojuego.form</field>
  <field name="model">videojuego.videojuego</field>
  <field name="arch" type="xml">
    <form string="Videojuego">
      <sheet>
        <group>
          <field name="id_juego"/>
          <field name="titulo"/>
          <field name="descripcion"/>
          <field name="categoria_id"/>
          <field name="precio"/>
        </group>
      </sheet>
    </form>
  </field>
</record>
```

Así se vería el menú de categorías con categorías ya creadas.



Screenshot of the Odoo 'Categoría' (Category) form. The browser address bar shows 'localhost:8069/odoo/action-111'. The form has a purple header with 'Categoría' and 'My Company'. Below the header is a search bar with 'Buscar...' and a 'Nuevo' button. The main content area is a list of categories with checkboxes:

- ☐ Nombre
- ☐ Aventura
- ☐ Acción
- ☐ RPG
- ☐ Estrategia
- ☐ Puzzles
- ☐ Plataformas
- ☐ Carreras
- ☐ Deportes
- ☐ Terror
- ☐ Simulación
- ☐ Sandbox

Y si ahora vemos el menú de videojuegos podemos ver que se ha añadido el atributo categorías.

Screenshot of the Odoo 'Videojuegos' (Video Games) form. The browser address bar shows 'localhost:8069/odoo/action-99'. The form has a purple header with 'Videojuegos' and 'My Company'. Below the header is a search bar with 'Buscar...' and a 'Nuevo' button. The main content area is a table of video games:

| <input type="checkbox"/> | ID | Título | Descripción | Categorías | Precio (€) |
|--------------------------|----|-----------------|--|-------------|------------|
| <input type="checkbox"/> | 0 | Portal 2 | ¡La «Iniciativa de Prueba Perpetua» se ha ampliado, permitiéndote ahora diseñar puzzles cooperativos para ti y tus amigos! | 4 registros | 9,75 |
| <input type="checkbox"/> | 1 | Bendy and ... | Bendy and the Ink Machine® is the first person puzzle action horror game that will forever ruin your childhood love of cartoons. | 4 registros | 18,49 |
| <input type="checkbox"/> | 2 | Bendy and ... | Bendy and the Dark Revival® is a first-person survival horror game set in the creepiest cartoon studio to ever exist. Discover the truth. Escape the studio. Above all, fear the Ink Demon...and survive. | 4 registros | 26,99 |
| <input type="checkbox"/> | 3 | Hello Neig... | Hello Neighbor es un juego de terror y sigilo donde debes meterte en la casa de tu vecino y averiguar los secretos que esconde en el sótano. Juega contra una IA avanzada que aprende de tus movimientos. ¿Te gusta escalar ventanas? Podrá una trampa para osos. ¿Tratas de escapar? Encontrará un atajo. | 5 registros | 28,99 |
| <input type="checkbox"/> | 4 | Little Night... | ¡Sumérgete en la enigmática atmósfera de Little Nightmares y enfréntate a los miedos de tu infancia! Ayuda a Seis a escapar de Las Fauces, un misterioso navío donde moran ánimas corrompidas en busca de su próxima comida... | 4 registros | 19,99 |
| <input type="checkbox"/> | 5 | Little Night... | Little Nightmares II es un juego de aventura y suspense en el que eres Mono, un joven atrapado en un mundo distorsionado por una malvada transmisión. Junto a su nueva amiga Six, Mono se dispone a descubrir de dónde viene esa transmisión. | 5 registros | 29,99 |
| <input type="checkbox"/> | 6 | Poppy Play... | You must stay alive in this horror/puzzle adventure. Try to survive the vengeful toys waiting for you in the abandoned toy factory. Use your GrabPack to hack electrical circuits or nab anything from afar. Explore the mysterious facility... and don't get caught. | 5 registros | 42,48 |
| <input type="checkbox"/> | 7 | Superliminal | La percepción es la realidad: este juego de rompecabezas en primera persona conseguirá dar la vuelta a tu forma de pensar. Escapa de un sueño surrealista resolviendo puzzles imposibles y usando la ambigüedad de la profundidad y la perspectiva. | 4 registros | 19,50 |



6. Agradecimientos

1. **Chat GPT**: buscar y encontrar la solución a los múltiples mensajes de error en los intentos fallidos de hacer funcional los modelos y vistas.