

TUTORIAL

Creación de Módulo Personalizado en Odoo

Para gestión de tu negocio de impresión 3D

Samuel Hernández Guerrero · DAM2 · Odoo 17+

0. Introducción

Este tutorial te guiará paso a paso para crear un módulo personalizado en Odoo que te permita gestionar los pedidos de clientes y el stock de filamento de tu granja de impresoras 3D.

No necesitas instalar nada extra para programar el módulo — solo acceso al servidor donde está instalado Odoo y un editor de texto. Los ejemplos usan Python básico con la capa ORM de Odoo.

¿Qué vamos a construir?

- Un modelo de Pedido de Impresión (print.order) donde el cliente solicita piezas.
- Un modelo de Filamento (print.filament) para controlar el stock de material.
- Vistas de lista, formulario y búsqueda para cada modelo.
- Una barra de estado en los pedidos (Borrador → En Cola → Imprimiendo → Entregado).
- Lógica Python para descontar automáticamente filamento al iniciar una impresión.

Aplicaciones de Odoo a instalar

Antes de crear el módulo, asegúrate de tener instaladas estas apps desde el menú Apps de Odoo:

App	Para qué la necesitas
Ventas (sale)	Gestión de presupuestos y pedidos de clientes. Imprescindible.
Inventario (stock)	Control de stock de filamento, entradas y salidas de material.
Contactos (contacts)	Base de datos de clientes y proveedores de filamento.
Facturación (account)	Opcional: para generar facturas al entregar los pedidos.
Sitio Web (website)	Opcional: si quieres que los clientes hagan pedidos online.

1. Estructura de un Módulo Odoo

Un módulo en Odoo es simplemente una carpeta con una estructura concreta de archivos. Odoo la reconoce automáticamente si está en el directorio de addons del servidor.

1.1 Carpeta y archivos mínimos

Crea esta carpeta dentro del directorio de addons de tu Odoo (normalmente /odoo/addons/ o /opt/odoo/addons/):

```
print3d/
├── __init__.py
└── __manifest__.py
```

```
|-- models/
|  |-- __init__.py
|  |-- print_order.py
|  |-- print_filament.py
|-- views/
|  |-- print_order_views.xml
|  |-- print_filament_views.xml
|-- security/
    |-- ir.model.access.csv
```

💡 ¿Qué es cada archivo?

`__init__.py` le dice a Python que esta carpeta es un paquete. `__manifest__.py` describe el módulo (nombre, versión, dependencias). `models/` contiene las clases Python con la lógica. `views/` contiene los XML con las pantallas. `security/` controla quién puede ver/editar qué.

1.2 El archivo `__manifest__.py`

Este es el archivo de configuración del módulo. Cópialo y adapta el nombre:

```
{  
    'name': 'Gestión Impresión 3D',  
    'version': '1.0',  
    'summary': 'Pedidos y stock de filamento para granja de impresoras 3D',  
    'author': 'Samuel Hernandez',  
    'depends': ['base', 'stock', 'sale'],  
    'data': [  
        'security/ir.model.access.csv',  
        'views/print_filament_views.xml',  
        'views/print_order_views.xml',  
    ],  
    'installable': True,  
    'application': True,  
}
```

⚠️ Importante

El campo 'depends' es clave. Al poner 'stock' y 'sale', tu módulo puede usar modelos de esas apps (productos, clientes, etc.) sin reinventarlos desde cero.

2. Los Modelos Python (La Lógica del Negocio)

Un modelo en Odoo es una clase Python que hereda de `models.Model`. Cada clase se convierte automáticamente en una tabla en la base de datos PostgreSQL. No tienes que escribir SQL.

2.1 Modelo de Filamento

Crea el archivo `models/print_filament.py` con este contenido:

```
from odoo import models, fields, api

class PrintFilament(models.Model):
    _name = 'print.filament'          # nombre técnico del modelo
    _description = 'Stock de Filamento 3D'

    name = fields.Char(
        string='Nombre',           # etiqueta que ve el usuario
        required=True              # campo obligatorio
    )
    material = fields.Selection([
        ('pla', 'PLA'),
        ('petg', 'PETG'),
        ('abs', 'ABS'),
        ('tpu', 'TPU (flexible)'),
        ('resin', 'Resina'),
    ], string='Material', required=True)
    color = fields.Char(string='Color')
    stock_kg = fields.Float(
        string='Stock (kg)',
        digits=(10, 3),            # 3 decimales
        default=0.0
    )
    stock_minimo_kg = fields.Float(
        string='Stock mínimo (kg)', # alerta si baja de 0.5 kg
        default=0.5
    )
    proveedor_id = fields.Many2one(
        'res.partner',             # usa el modelo de contactos de Odoo
        string='Proveedor'
    )
    bajo_stock = fields.Boolean(
        string='Bajo stock',
        compute='_compute_bajo_stock', # campo calculado automáticamente
        store=True                  # se guarda en BD para poder filtrar
    )

@api.depends('stock_kg', 'stock_minimo_kg')
def _compute_bajo_stock(self):
    for rec in self:           # 'self' es el conjunto de registros
        rec.bajo_stock = rec.stock_kg < rec.stock_minimo_kg
```

💡 ¿Qué es `@api.depends`?

Es un decorador de Odoo. Le dice: 'recalcula `bajo_stock` cada vez que cambie `stock_kg` o `stock_minimo_kg`'. Es automático, no tienes que llamarlo manualmente.

2.2 Modelo de Pedido de Impresión

Crea el archivo models/print_order.py:

```
from odoo import models, fields, api
from odoo.exceptions import ValidationError

class PrintOrder(models.Model):
    _name = 'print.order'
    _description = 'Pedido de Impresión 3D'
    _order = 'create_date desc' # los más nuevos primero

    name = fields.Char(
        string='Referencia',
        readonly=True,
        default='Nuevo'
    )
    cliente_id = fields.Many2one(
        'res.partner',
        string='Cliente',
        required=True
    )
    descripcion = fields.Text(string='Descripción de la pieza')
    filamento_id = fields.Many2one(
        'print.filament', # enlaza con nuestro modelo de filamento
        string='Filamento a usar',
        required=True
    )
    cantidad_kg = fields.Float(
        string='Filamento necesario (kg)',
        digits=(10, 3)
    )
    precio_unitario = fields.Float(string='Precio (€)')
    fecha_entrega = fields.Date(string='Fecha de entrega prevista')
    notas = fields.Text(string='Notas internas')
    state = fields.Selection([
        ('draft', 'Borrador'),
        ('queue', 'En cola'),
        ('printing', 'Imprimiendo'),
        ('done', 'Entregado'),
        ('cancelled', 'Cancelado'),
    ], default='draft', string='Estado', tracking=True)

# — Secuencia automática ——————
@api.model_create_multi
def create(self, vals_list):
    for vals in vals_list:
        if vals.get('name', 'Nuevo') == 'Nuevo':
            vals['name'] = self.env['ir.sequence'].next_by_code(
                'print.order') or 'Nuevo'
    return super().create(vals_list)
```

```

# — Botones de cambio de estado ——————
def action_queue(self):
    self.state = 'queue'

def action_start_printing(self):
    for rec in self:
        # Comprobar que hay suficiente filamento
        if rec.filamento_id.stock_kg < rec.cantidad_kg:
            raise ValidationError(
                f'Stock insuficiente: hay {rec.filamento_id.stock_kg} kg '
                f'pero el pedido necesita {rec.cantidad_kg} kg.'
            )
        # Descontar filamento del stock
        rec.filamento_id.stock_kg -= rec.cantidad_kg
        rec.state = 'printing'

def action_done(self):
    self.state = 'done'

def action_cancel(self):
    self.state = 'cancelled'

```

ValidationError

Cuando lanzas un ValidationError, Odoo muestra automáticamente un mensaje de error en pantalla al usuario. No hace falta código extra en la vista.

2.3 Registrar los modelos en __init__.py

Edita models/__init__.py para que Python cargue tus clases:

```
from . import print_filament
from . import print_order
```

Y el __init__.py raíz del módulo (print3d/__init__.py):

```
from . import models
```

3. Seguridad: Permisos de Acceso

Si no existe este archivo, Odoo no deja acceder a tus modelos aunque estén bien creados. Crea security/ir.model.access.csv con exactamente estas columnas:

```
id,name,model_id:id,group_id:id,perm_read,perm_write,perm_create,perm_unlink
access_print_filament_user,print.filament.user,model_print_filament,,1,1,1,0
access_print_order_user,print.order.user,model_print_order,,1,1,1,0
```

Explicación de las columnas

Las últimas 4 columnas son: leer, escribir, crear, eliminar (1=sí, 0=no). La columna group_id vacía significa 'todos los usuarios'. Puedes restringirlo a grupos concretos si lo necesitas más adelante.

4. Secuencia Automática para Pedidos

Para que los pedidos tengan referencias como IMP/2025/0001, IMP/2025/0002..., añade un fichero de datos XML. Crea data/sequences.xml:

Primero, crea la carpeta data/ dentro de print3d/ y el archivo sequences.xml:

```
<?xml version='1.0' encoding='UTF-8'?>
<odoo>
    <data noupdate='1'>
        <record model='ir.sequence' id='seq_print_order'>
            <field name='name'>Pedido Impresión 3D</field>
            <field name='code'>print.order</field>
            <field name='prefix'>IMP/%(year)s</field>
            <field name='padding'>4</field>
        </record>
    </data>
</odoo>
```

Luego añade este archivo al __manifest__.py, en la lista 'data', antes que las vistas:

```
'data': [
    'security/ir.model.access.csv',
    'data/sequences.xml',           # <-- añadir aquí
    'views/print_filament_views.xml',
    'views/print_order_views.xml',
],
```

5. Las Vistas XML (Lo Que Ve el Usuario)

Las vistas definen cómo se muestran los datos. En Odoo hay tres vistas principales: formulario (un registro), lista (tabla) y búsqueda (filtros). Las defines en XML.

5.1 Vista de Filamento

Crea views/print_filament_views.xml:

```

<?xml version='1.0' encoding='UTF-8'?>
<odoo>

    <!-- — Formulario -->
    <record id='view_print_filament_form' model='ir.ui.view'>
        <field name='name'>print.filament.form</field>
        <field name='model'>print.filament</field>
        <field name='arch' type='xml'>
            <form string='Filamento'>
                <group>
                    <field name='name'/>
                    <field name='material'/>
                    <field name='color'/>
                    <field name='proveedor_id'/>
                </group>
                <group string='Stock'>
                    <field name='stock_kg'/>
                    <field name='stock_minimo_kg'/>
                    <field name='bajo_stock' readonly='1' />
                </group>
            </form>
        </field>
    </record>

    <!-- — Lista -->
    <record id='view_print_filament_list' model='ir.ui.view'>
        <field name='name'>print.filament.list</field>
        <field name='model'>print.filament</field>
        <field name='arch' type='xml'>
            <list string='Filamentos' decoration-danger='bajo_stock==True'>
                <field name='name'/>
                <field name='material'/>
                <field name='color'/>
                <field name='stock_kg'/>
                <field name='stock_minimo_kg'/>
                <field name='bajo_stock' />
            </list>
        </field>
    </record>

    <!-- — Acción de ventana -->
    <record id='action_print_filament' model='ir.actions.act_window'>
        <field name='name'>Filamentos</field>
        <field name='res_model'>print.filament</field>
        <field name='view_mode'>list,form</field>
    </record>

</odoo>

```

💡 decoration-danger

La propiedad `decoration-danger='bajo_stock==True'` pinta la fila en rojo cuando el stock está por debajo del mínimo. No necesitas código Python para esto.

5.2 Vista de Pedidos

Crea `views/print_order_views.xml`:

```
<?xml version='1.0' encoding='UTF-8'?>
<odoo>

    <!-- — Formulario con barra de estado —-->
    <record id='view_print_order_form' model='ir.ui.view'>
        <field name='name'>print.order.form</field>
        <field name='model'>print.order</field>
        <field name='arch' type='xml'>
            <form string='Pedido de Impresión'>
                <header>
                    <!-- Botones que aparecen según el estado -->
                    <button name='action_queue' string='Poner en cola'
                            type='object' invisible='state != "draft"'
                            class='oe_highlight'/>
                    <button name='action_start_printing' string='Iniciar impresión'
                            type='object' invisible='state != "queue"'
                            class='oe_highlight'/>
                    <button name='action_done' string='Marcar como entregado'
                            type='object' invisible='state != "printing"' />
                    <button name='action_cancel' string='Cancelar'
                            type='object'
                            invisible='state in ["done", "cancelled"]' />
                    <!-- Barra de estado visual -->
                    <field name='state' widget='statusbar'
                            statusbar_visible='draft,queue,printing,done' />
                </header>
                <sheet>
                    <div class='oe_title'>
                        <h1><field name='name' readonly='1' /></h1>
                    </div>
                    <group>
                        <field name='cliente_id' />
                        <field name='fecha_entrega' />
                        <field name='precio_unitario' />
                    </group>
                    <group string='Impresión'>
                        <field name='filamento_id' />
                        <field name='cantidad_kg' />
                        <field name='descripcion' />
                    </group>
                    <group string='Notas internas'>
                        <field name='notas' nolabel='1' />
                    </group>
                </sheet>
            </form>
        </field>
    </record>
</odoo>
```

```

        </group>
    </sheet>
</form>
</field>
</record>

<!-- — Lista -->
<record id='view_print_order_list' model='ir.ui.view'>
    <field name='name'>print.order.list</field>
    <field name='model'>print.order</field>
    <field name='arch' type='xml'>
        <list string='Pedidos'>
            decoration-warning='state == "queue"'
            decoration-info='state == "printing"'
            decoration-success='state == "done"'>
                <field name='name'/>
                <field name='cliente_id'/>
                <field name='filamento_id'/>
                <field name='cantidad_kg'/>
                <field name='fecha_entrega'/>
                <field name='state'/>
            </list>
        </field>
    </record>

<!-- — Búsqueda con filtros -->
<record id='view_print_order_search' model='ir.ui.view'>
    <field name='name'>print.order.search</field>
    <field name='model'>print.order</field>
    <field name='arch' type='xml'>
        <search>
            <field name='name' string='Referencia'/>
            <field name='cliente_id'/>
            <filter name='en_col' string='En cola'
domain="['state','=','=','queue'])"/>
            <filter name='imprimiendo' string='Imprimiendo'
domain="['state','=','=','printing'])"/>
            <filter name='entregados' string='Entregados'
domain="['state','=','=','done'])"/>
            <group expand='0' string='Agrupar por'>
                <filter name='by_client' string='Cliente'
context="{'group_by':'cliente_id'}"/>
                <filter name='by_state' string='Estado' context="{'group_by':'state'}"/>
            </group>
        </search>
    </field>
</record>

<!-- — Acción -->
<record id='action_print_order' model='ir.actions.act_window'>
    <field name='name'>Pedidos de Impresión</field>
    <field name='res_model'>print.order</field>
    <field name='view_mode'>list,form</field>

```

```

<field name='search_view_id' ref='view_print_order_search'/>
<field name='context'>{'search_default_imprimiendo': 1}</field>
</record>

<!-- — Menú -->
<menuitem id='menu_print3d_root'      name='Impresión 3D'
          sequence='10'/>
<menuitem id='menu_print_orders'     name='Pedidos'
          parent='menu_print3d_root'
          action='action_print_order'
          sequence='10'/>
<menuitem id='menu_print_filament'   name='Filamentos'
          parent='menu_print3d_root'
          action='action_print_filament'
          sequence='20'/>

</odoor>

```

6. Instalar el Módulo en Odoo

Una vez que tienes todos los archivos creados, sigue estos pasos para activarlo:

1. Reinicia el servidor de Odoo. Esto es imprescindible para que Odoo detecte el nuevo módulo. En terminal:

```

sudo systemctl restart odoo
# o si usas un entorno de desarrollo:
python odoo-bin -d TU_BASE_DE_DATOS -u print3d

```

2. Entra en Odoo con tu usuario administrador.
3. Ve al menú superior → Apps.
4. Haz clic en Actualizar lista de apps (botón en la parte superior).
5. Busca 'Gestión Impresión 3D' en la barra de búsqueda.
6. Haz clic en Instalar.
7. Cuando termine, verás el nuevo menú 'Impresión 3D' en la barra superior.

Si hay errores al instalar

Odoo muestra el error en pantalla y también en el log del servidor (odoo.log). Los errores más comunes son: typos en el CSV de seguridad, XML mal cerrado, o un campo referenced que no existe. Lee el mensaje de error con calma — casi siempre te dice exactamente qué línea falla.

7. Ciclo de Desarrollo: Editar y Actualizar

Modo desarrollador desde Odoo

8. Ve a Ajustes → Activar el modo desarrollador.
9. Ve a Apps → busca tu módulo → Actualizar.

💡 Consejo

Activa siempre el modo desarrollador cuando estés construyendo el módulo. Te permite ver IDs técnicos, depurar vistas y acceder a herramientas extra en el menú de Depuración (ícono de insecto en la barra superior).

8. Resultados finales

The figure consists of three vertically stacked screenshots of the Odoo application interface, specifically the 'Filaments' module.

- Screenshot 1:** Shows a list view of 'Pedidos de Impresión'. The columns are 'Referencia', 'Cliente', 'Filamento a usar', 'Filamento n...', 'Fecha de e...', and 'Estado'. There are two entries: 'Basic' (PLA, Negro, Stock: 100,000, Estado: Cancelado) and 'asdiasd' (ABS, Stock: 10.120,000, Estado: Confirmado). A search bar at the top is set to 'Buscar...'.
- Screenshot 2:** Shows a list view of 'Filamentos'. The columns are 'Nombre', 'Material', 'Color', 'Stock (kg)', 'Stock mínim...', 'Bajo stock', and 'Estado'. It lists 'Basic' (PLA, Negro, Stock: 100,000, Estado: Cancelado) and 'asdiasd' (ABS, Stock: 10.120,000, Estado: Confirmado). A search bar at the top is set to 'Buscar...'. A page navigation bar shows '1-2 / 2'.
- Screenshot 3:** Shows a 'Nuevo' (New) form for creating a new print order. The form fields include 'Cliente', 'Fecha de entrega prevista', 'Precio (€)' (set to 0,00), 'IMPRESIÓN' section with 'Filamento a usar' (set to 0,000 kg), and a 'NOTAS INTERNAS' (Internal Notes) section. At the bottom, there are buttons for 'Poner en cola' (Put in Queue), 'Cancelar' (Cancel), and status indicators: 'Borrador' (Draft), 'En cola' (In Queue), 'Imprimiendo' (Printing), and 'Entregado' (Delivered).

Nuevo

Filamentos
Nuevo 

Nombre

Material

Color

Proveedor

Estado

STOCK

Stock (kg) 0,000

Stock mínimo (kg) 0,50

Bajo stock

