

PASOS PARA LA RESOLUCIÓN

1. Se levanta el Docker.

```
● PS C:\Users\Usuario\Desktop\Alicia\2. DAM\4. SGE\SGE_practicas> docker compose up -d
[+] Running 2/2
  ✓ Container db      Started
  ✓ Container odoo_16  Started
```

2. Establecimiento de credenciales.

Warning, your Odoo database manager is not protected. To secure it, we have generated the following master password for it:

qgcy-cdq2-gjtw

You can change it below but be sure to remember it, it will be asked for future operations on databases.

Master Password	<input type="button" value="eye"/>
Database Name	db_gestion_biblioteca	
Email	alicia@alicia.com	
Password	<input type="button" value="eye"/>
Phone number	666666666	
Language	Spanish / Español	<input type="button" value="▼"/>
Country	Spain	<input type="button" value="▼"/>
Demo data	<input type="checkbox"/>	

3. Instalación de una aplicación para poder activar el modo de desarrollador.

The screenshot shows the Odoo App Store interface. On the left, there's a sidebar with 'CATEGORÍAS' and a 'Todos' section listing categories like Ventas, Servicios, Contabilidad, etc., with counts (e.g., Ventas: 7). The main area displays a grid of applications:

- Ventas**: De presupuestos a facturas. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Facturación**: Facturas y pagos. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- CRM**: Seguimiento de leads y oportunidades próximas. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- MRP II**: Órdenes de trabajo, planificación, informes de inventario. Includes 'APRENDA MÁS' and 'Actualizar' buttons.
- Sitio web**: Creador de sitios web para empresas. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Inventario**: Gestione sus actividades de inventario y logística. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Fabricación**: Órdenes de fabricación y listas de materiales. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Información**: Centralice, gestione, comparta y haga crecer su librería de conocimiento. Includes 'APRENDA MÁS' and 'Actualizar' buttons.
- Compra**: Pedidos de compra, licitaciones y acuerdos. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Comercio electrónico**: Venda sus productos en línea. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Proyecto**: Organice y planifique sus proyectos. Includes 'ACTIVAR' and 'APRENDA MÁS' buttons.
- Partes de horas**: Seguimiento del tiempo de empleados en las tareas. Includes 'APRENDA MÁS' and 'Actualizar' buttons.
- Gastos**
- Studio**
- Ausencias**

3. Activación del modo de desarrollados en Ajustes.

The screenshot shows the Odoo Settings (Ajustes) page. On the left, there's a sidebar with 'Opciones generales' and 'Facturación / Contabilidad'. The main area has sections for 'Autenticación OAuth', 'Autenticación LDAP', 'Geolocalización', and 'Herramientas de desarrollador'. In the 'Herramientas de desarrollador' section, the 'Activar modo de desarrollador' button is highlighted with a red box. Below it are other options: 'Activar modo de desarrollador (con activos)' and 'Activar el modo de desarrollador (con activos de prueba)'. At the bottom, there are download links for Google Play and the App Store.

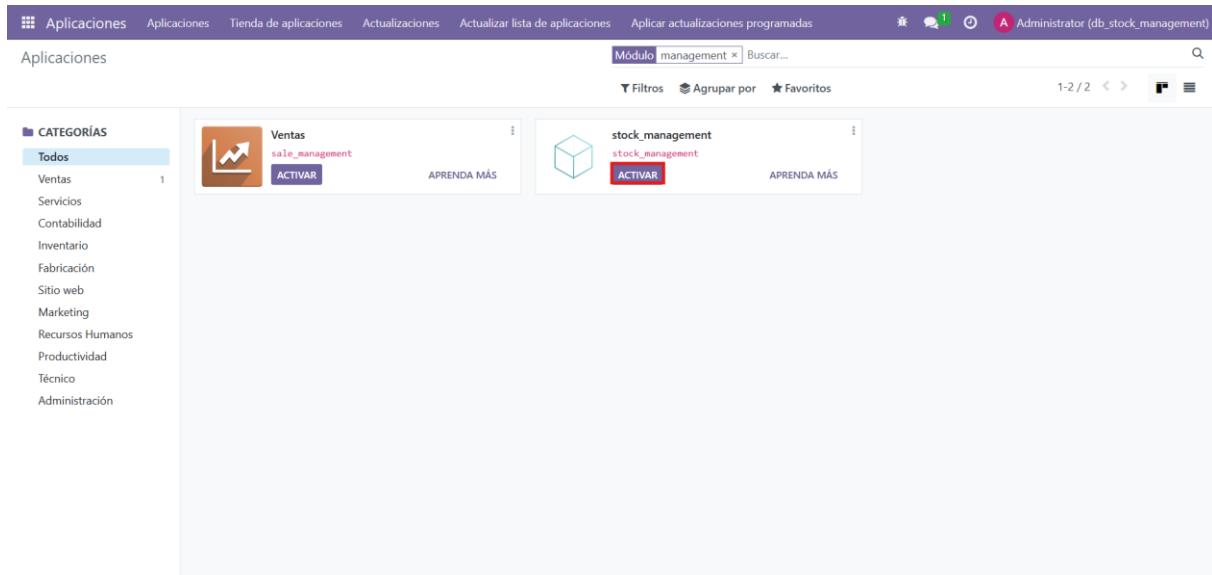
4. Acceso, mediante la línea de comandos, al contenedor Odoo.

```
PS C:\Users\Usuario\Desktop\Alicia\2. DAM\4. SGE\SGE_practicas> docker exec -it f385905d6a2a bash
odoo@f385905d6a2a:/$
```

5. Creación de la estructura básica del módulo con scaffold.

```
odoo@f385905d6a2a:/$ odoo scaffold stock_management /mnt/extra-addons/
odoo@f385905d6a2a:/$
```

6. Instalación del módulo.



7. Edición de los archivos necesarios para la correcta implementación del módulo.

manifest.py

```
# -*- coding: utf-8 -*-
{
    'name': "stock_management",

    'summary': 'Gestión de stock de productos',

    'description': 'Módulo para gestionar el stock de productos, incluyendo categorías, precios y cantidades.',

    'author': 'Alicia',
    'website': 'https://www.alicia.com',

    # Categories can be used to filter modules in modules listing
    # Check
    https://github.com/odoo/odoo/blob/16.0/odoo/addons/base/data/ir_module_category_data.xml
    # for the full list
    'category': 'Uncategorized',
    'version': '0.1',

    # any module necessary for this one to work correctly
    'depends': ['base'],

    # always loaded
    'data': [
```

```

        '# security/ir.model.access.csv',
        'security/ir.model.access.csv',
        'views/stock_product_views.xml',
    ],
    # only loaded in demonstration mode
    'demo': [
        'demo/demo.xml',
    ],
}

```

__init.py__

```

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

from . import stock_product
ir.model.access.csv

```

	id	name	model_id:id	group_id:id	perm_read	perm_write	perm_create	perm_unlink
1	access_stock_product	access_stock_product	model_stock_product_stock_product	base.group_user	1	1	1	1
2								

stock_product.py

```

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

class StockProduct(models.Model):
    _name = 'stock_product.stock_product'
    _description = 'Gestión de Stock'

    _sql_constraints = [
        ('unique_name', 'unique(name)', 'El nombre debe ser único'),
        ('check_quantity', 'CHECK(quantity >= 0)', 'La cantidad debe ser positiva'),
    ]

    name = fields.Char(
        string='Nombre producto',
        required=True)
    category = fields.Selection([
        ('television', 'Televisión'),
        ('pc', 'PC'),
        ('monitor', 'Monitor'),
        ('smartphone', 'Smartphone'),
    ],
        string='Categoria',
        required=True)

    price = fields.Float(string='Precio unitario')
    quantity = fields.Float(string='Cantidad de stock')

```

```

total_value = fields.Float(
    string='Valor total',
    compute='_compute_total', inverse='_inverse_total',
    store=True)
minimun_quantity = fields.Integer(
    string='Cantidad minima')

stock_status = fields.Selection([
    ('normal', 'Normal'),
    ('low stock', 'Low Stock'),
],
string="Estado",
compute='_compute_stock_status')

full_name = fields.Char(
    string='Nombre Completo', compute='_compute_full_name')

@api.depends('price', 'quantity')
def _compute_total(self):
    for record in self:
        record.total_value = record.price * record.quantity

def _inverse_total(self):
    for record in self:
        if record.quantity and record.quantity != 0:
            record.price = record.total_value / record.quantity

@api.depends('quantity', 'minimun_quantity')
def _compute_stock_status(self):
    for rec in self:
        if rec.quantity < rec.minimun_quantity:
            rec.stock_status = 'low stock'
        else:
            rec.stock_status = 'normal'

@api.depends('name', 'category')
def _compute_full_name(self):
    for rec in self:
        rec.full_name = f'{rec.name} ({rec.category})' if rec.name and
rec.category else rec.name

@api.constrains('price')
def _check_price(self):
    for rec in self:
        if rec.price < 0:
            raise ValidationError('El precio debe ser mayor que 0')

```

stock_product_views.xml

```

<odoo>
    <record id="view_stock_product_tree" model="ir.ui.view">
        <field name="name">stock.product.tree</field>
        <field name="model">stock_product.stock_product</field>
        <field name="arch" type="xml">
            <tree>
                <field name="full_name"/>
                <field name="category"/>
                <field name="price"/>
                <field name="quantity"/>
                <field name="total_value"/>
                <field name="stock_status"/>
            </tree>
        </field>
    </record>

    <record id="action_stock_product" model="ir.actions.act_window">
        <field name="name">Productos de Stock</field>
        <field name="res_model">stock_product.stock_product</field>
        <field name="view_mode">tree,form</field>
    </record>

    <menuitem id="menu_stock_root" name="Stock Management"/>
    <menuitem id="menu_stock_product" name="Productos"
parent="menu_stock_root" action="action_stock_product"/>
</odoo>

```

8. Agregación de registros para comprobar el correcto funcionamiento

The screenshot shows the Odoo Stock Management interface. The top navigation bar includes 'Stock Management' and 'Productos'. The main view displays a product record for 'TV Samsung' with the following details:

Nombre producto?	TV Samsung	Categoría?	Televisión
Precio unitario?	500,00	Cantidad de stock?	2,00
Valor total?	1.000,00	Cantidad mínima?	2
Estado?	Normal	Nombre Completo?	TV Samsung (televisión)

At the bottom right of the interface, there are buttons for 'Acción', 'Nuevo', and page navigation.

Stock Management Productos

Productos de Stock

NUEVO

Buscar...

Filtros Agrupar por Favoritos

1-1 / 1

<input type="checkbox"/> Nombre Completo	Categoría	Precio unitario	Cantidad de stock	Valor total	Estado
<input type="checkbox"/> TV Samsung (television)	Televisión	500,00	2,00	1.000,00	Normal

Stock Management Productos

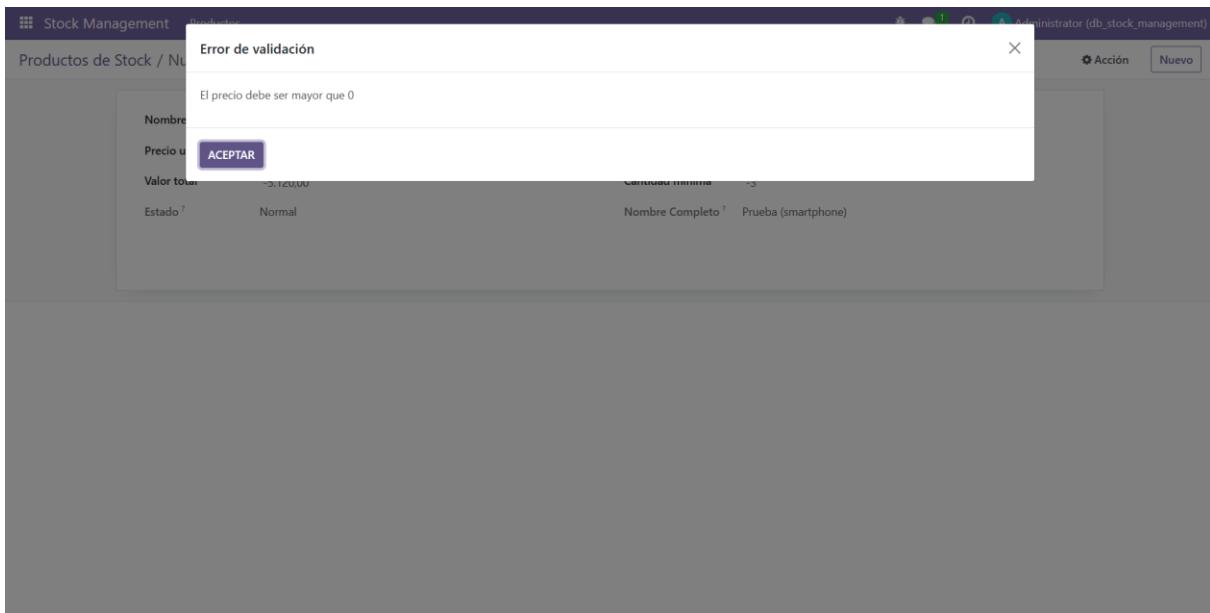
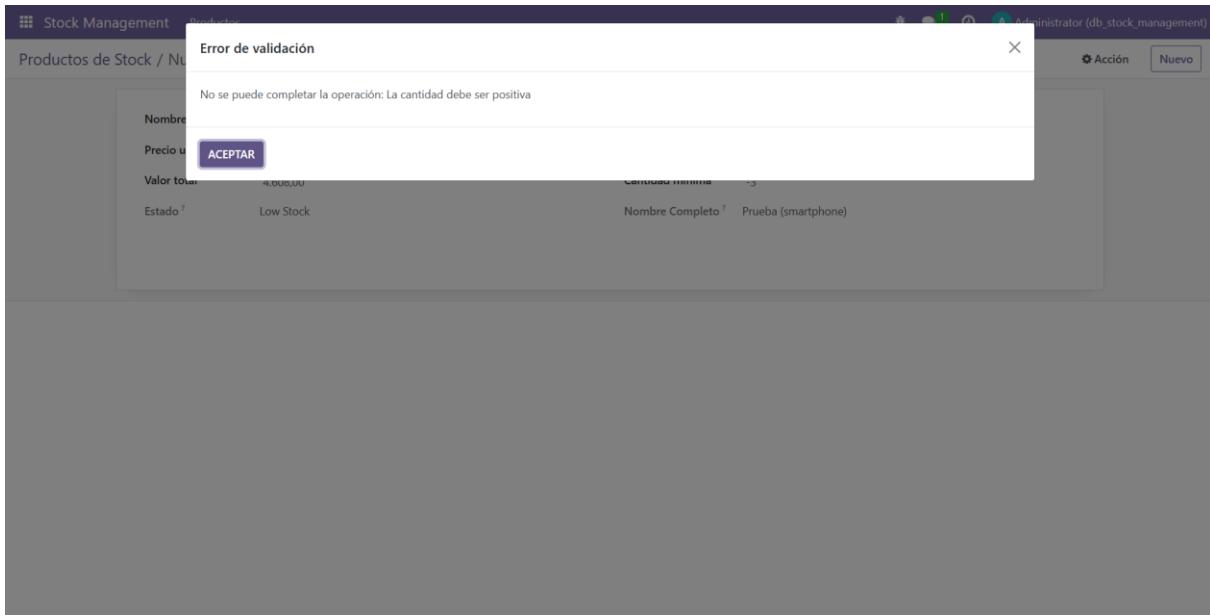
Productos de Stock / Nuevo

Guardar manualmente

Nombre producto	Categoría
Precio unitario 0,00	Cantidad de stock 0,00
Valor total 0,00	Cantidad mínima 0
Estado Normal	Nombre Completo

Campos no válidos:

- Nombre producto
- Categoría



Stock Management Productos

Productos de Stock

NUEVO

Buscar...

Filtros Agrupar por Favoritos

1-2 / 2

<input type="checkbox"/> Nombre Completo	Categoría	Precio unitario	Cantidad de stock	Valor total	Estado
<input type="checkbox"/> TV Samsung (television)	Televisión	500,00	2,00	1.000,00	Normal
<input type="checkbox"/> Prueba (smartphone)	Smartphone	700,00	10,00	7.000,00	Normal