CÓDIGOS EDITADOS

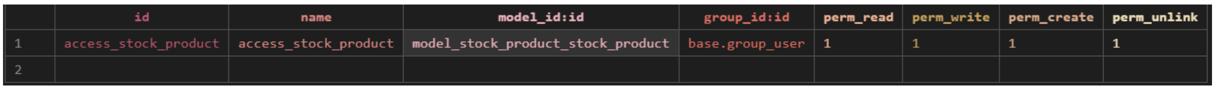
__manifest.py__

```
# -*- 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_categor
y_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								

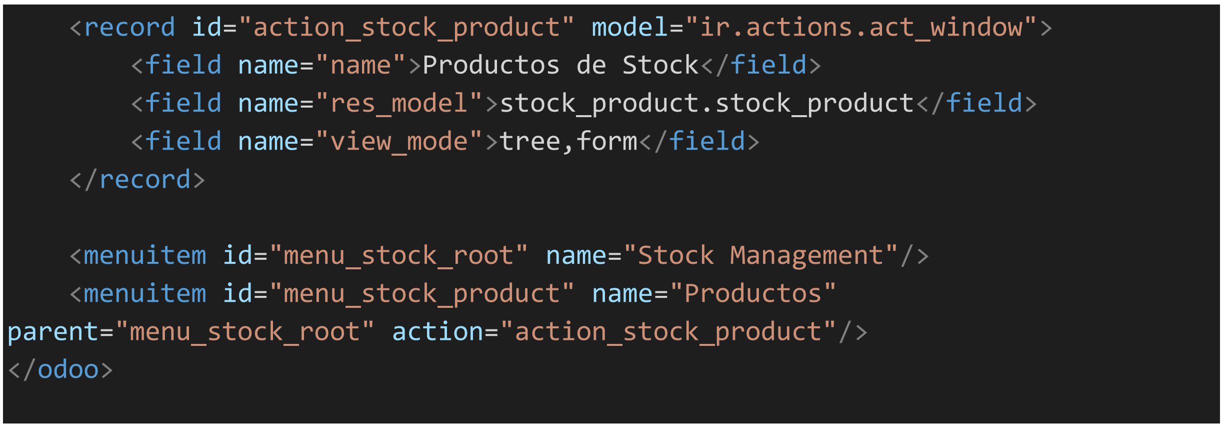
stock_product.py

```
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')
```

```python
    @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

```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>
```