

Paso 1) primeramente renombramos la carpeta donde teníamos el modul automatic o como le hayamos llamado y le nombramos “component”

y como podrás ver la estructura del módulo no se modifica pero si debemos modificar el fichero venda_component.py a component.py y crear en él una relación con la clase “Marca”.

```
root@debian:/home/john/odoo-modulos/addons/component# ls
controllers demo __init__.py __manifest__.py models security views
root@debian:/home/john/odoo-modulos/addons/component#
```

Paso 2) entramos en la carpeta models una vez modificado el nombre de nuestra carpeta modulAutomatic y y listamos los 2 ficheros de ese directorio con un “ls” y podras ver 2 ficheros venda_component.py e __init__.py.

renombraremos a venda_component.py por component.py mas que todo por un orden y accederemos a cada uno de ellos y les haremos cambios(ir al siguiente paso)

```
root@debian:/home/john/odoo-modulos/addons/component# cd models
root@debian:/home/john/odoo-modulos/addons/component/models# ls
__init__.py venda_component.py
root@debian:/home/john/odoo-modulos/addons/component/models# mv venda_component.py component.py
root@debian:/home/john/odoo-modulos/addons/component/models# ls
component.py __init__.py
root@debian:/home/john/odoo-modulos/addons/component/models# _
```

paso 3) Rellenamos los ficheros __init__.py y component.py con los siguientes contenidos

```
# -*- coding: utf-8 -*-
#Hola soy Jonathan Calderon dejo esto aqui para validar mi autenticidad de los cambios
from . import component
```

```

# -*- coding: utf-8 -*-
from odoo import models, fields, api
#Soy Jonathan Calderon

COLOR=[('negre','Negre'),
       ('plata','Plata'),
       ('gris','Gris'),]

class Component(models.Model):
    _name='gestio.component'
    name=fields.Char(string='Nom del component',required=True)
    marca=fields.Many2one('gestio.marca',string='Marca')
    color=fields.Selection(COLOR,string='Color',default='negre')
    is_offer=fields.Boolean('Té descompte?')

class Marca(models.Model):
    _name='gestio.marca'
    name=fields.Char(string='Nombre',required=True)
    nac_fab=fields.Many2one('res.country',string='Nacionalitat')

```

hasta aqui hemos terminado la primera parte de la practica.

HERENCIA

en este apartado tenemos que hacer los siguientes pasos:

- Fes una còpia de la carpeta del component a component_preu.
- Canvia el nom del model component.py a component_preu.py
- Canvia el nom del __init__.py del model a component_preu.

Paso 1-hacer una copia de la carpeta component

creamos un nuevo módulo que es básicamente una copia de Component por tanto se recomienda hacer una copia de esta carpeta.

```

root@debian:/home/john/odoo-modulos/addons# ls
component component_preu
root@debian:/home/john/odoo-modulos/addons# _

```

como se puede ver en la imagen component_preu es una copia de la carpeta component.

Paso 2-Renombrar fichero component.py

como tenemos una copia exacta de la carpeta component tambien tenemos los ficheros con los mismo nombres.

Lo siguiente es acceder a la carpeta models y modificar el fichero component.py a component_preu.py

Nota:Recordad que este es una nueva carpeta y que la vieja aun sigue conteniendo el fichero component.py

```
# -*- coding: utf-8 -*-
from odoo import models, fields, api
#Hola soy Jonathan Calderon dejo esto aqui para validar mi trabajo.

COLOR=[ ('negre','Negre'),
        ('plata','Plata'),
        ('gris','Gris'),
    ]
class Component_Preu(models.Model):
    _inherit='gestio.component'
    precio=fields.Float(string='Preu del component')
    color=fields.Selection(COLOR,string='Color',default='gris')
```

Paso 3-Agregar en el __init__.py la clase component_preu

```
GNU nano 3.2
# -*- coding: utf-8 -*-
@from . import Component_Preu
```