

Shell de Python (filter & get)

python manage.py Shell

```
cats = Category.objects.all()
cats
QuerySet [<Category: Catego 1>, <
```

```
p1 = Post.objects.get(id=1)
```

 (Puede regresar un error, en caso de que no exista)

```
> c = Category.objects.get(name='Catego 1')
> c.name
```

```
c = Category(name='Categoría General')
c.save()
```

 (Actualiza la base de datos)

```
>> c = Category.objects.filter(name='Categoría General')
>> c
QuerySet [<Category: Categoría General>]
>> c[0]
Category: Categoría General
```

 (Regresa un “iterable” un Queryset)

Notación doble subrayado __

Consultas ORM

```
c = Category.objects.filter(created__year = 2024)
c
QuerySet [<Category: Catego 1>, <Category: Catego 2>]
```

```
c = Category.objects.filter(name__contains='Categoría')
c
QuerySet [<Category: Categoría General>, <Category: Categoría 1>]
```

```
> c = Category.objects.filter(name__startswith='Cat')
> c
QuerySet [<Category: Catego 1>, <Category: Catego 2>, <Category: Categoría General>]
```

```
p = Post.objects.filter(categories__name='Catego 1')
p
```

```
QuerySet [p1, p2]
p = Post.objects.filter(categories__name__contains='Catego')
p
```

Mejorar vista del administrador

list_display

En el archivo admin.py de la aplicación “post”, modificarlo de la siguiente manera.

```
readonly_fields = ('created', 'updated',)
list_display = ('title', 'author', 'published',)
```



TÍTULO	AUTOR	FECHA DE
post 1	admin	15 de jul

Observa que muestra lo que se tenga en `__str__` de autor, si se desea personaliza la salida, se puede crear un método de la siguientes 2 maneras

Opción 1. Dentro del mismo archivo `admin.py`

```
readonly_fields = ('created', 'updated',)
list_display = ('title', 'author_name', 'published',)

def author_name(self, obj):
    return obj.author.first_name
```



<input type="checkbox"/>	TÍTULO	AUTHOR NAME	FECHA DE PUBLICACIÓN
<input type="checkbox"/>	post 1	Francisco	15 de julio de 2024

Opción 2. Desde el modelo:

```
def author_name(self):
    return self.author.first_name
```

`search_fields`

```
class PostAdmin(admin.ModelAdmin):
    readonly_fields = ('created', 'updated',)
    list_display = ('title', 'author_name', 'published',)
    search_fields = ('title', 'author_username',)
```



Q admin Buscar

Acción: Ir 0 de 1 seleccionados

<input type="checkbox"/>	TÍTULO	AUTHOR NAME	FECHA DE PUBLICACIÓN
<input type="checkbox"/>	post 1	Francisco	15 de julio de 2024

Práctica: cambiar por

```
'author__first_name',)
```



Q fran

Acción: Ir 0 de 1 seleccionados

<input type="checkbox"/>	TÍTULO	AUTHOR NAME	FECHA DE PUBLICACIÓN
<input type="checkbox"/>	post 1	Francisco	15 de julio de 2024

Observa que también puedes incluir el nombre del modelo de categorías

```
search_fields = ('title', 'author__first_name', 'categories__name')
```

`date_hierarchy`

```
date_hierarchy = 'published'
```



< 2024 15 de julio

Acción:

<input type="checkbox"/>	TÍTULO
--------------------------	--------

list_filter

```
date_hierarchy = 'published'
list_filter = ('title', 'author__first_name', 'published',)
```



FILTRO

Mostrar recuentos

↓ Por Título

Todo
post 1

↓ Por nombre

Todo
Francisco

↓ Por Fecha de publicación

Cualquier fecha
Hoy
Últimos 7 días

Práctica list_display

```
list_display = ('title', 'author_name', 'published', 'categories_names')
```

```
def categories_names(self, obj):
    res = ''
    for category in obj.categories.all().order_by("name"):
        res += category.name + ", "
    res = res[:-2]
    return res
categories_names.short_description = 'Categorías'
```



<input type="checkbox"/>	TÍTULO	AUTHOR NAME	FECHA DE PUBLICACIÓN	CATEGORÍAS
<input type="checkbox"/>	post 1	Francisco	15 de julio de 2024 a las 07:47	Catego 1

Puedes optimizarlo

```
def categories_names(self, obj):
    return ", ".join([category.name for category in obj.categories.all().order_by("name")])
```