# Django: objects.filter()

## 1️ Uso básico de filter()

El método filter() se utiliza para obtener registros que cumplan una o más condiciones en un modelo de Django. Devuelve un QuerySet (una lista de objetos).

* Ejemplo básico:

Persona.objects.filter(nombre="Francisco")  
 Devuelve todas las personas cuyo nombre sea “Francisco”.

## 2️ Filtros sencillos

Algunos ejemplos de filtros básicos:

Persona.objects.filter(edad=25) # Igualdad  
Persona.objects.filter(edad\_\_lt=30) # Menor que  
Persona.objects.filter(edad\_\_gt=50) # Mayor que  
Persona.objects.filter(nombre\_\_icontains='an') # Contiene (sin distinción de mayúsculas)  
Persona.objects.filter(ciudad\_\_startswith='M') # Empieza por 'M'

## 3️ Filtros con fechas

Persona.objects.filter(fecha\_nacimiento\_\_year=1990)  
Persona.objects.filter(fecha\_nacimiento\_\_range=["1990-01-01", "2000-12-31"])

## 4️ Filtros combinados

Puedes combinar varios filtros con AND (por defecto), OR o NOT:

# AND (implícito)  
Persona.objects.filter(nombre="Ana", edad=30)  
  
# OR (usando Q)  
from django.db.models import Q  
Persona.objects.filter(Q(nombre="Ana") | Q(nombre="Luis"))  
  
# NOT (usando exclude)  
Persona.objects.exclude(nombre="Ana")

## 5️ Métodos útiles del QuerySet

.count() → Devuelve cuántos resultados hay.  
.exists() → Devuelve True/False si hay resultados.  
.first() → Devuelve el primer objeto.  
.last() → Devuelve el último objeto.  
.all() → Devuelve todos los registros.

# Bibliografía

<https://docs.djangoproject.com/en/5.2/topics/db/queries/>