

Lección 3: Actividad de portafolio nro. 3

“Gestión de relaciones en Django ORM”

Nicolás Jofré Andrade - Desarrollo Full-stack Python
05-2024

Objetivo: Implementar modelos en Django que demuestren el manejo de relaciones, incluyendo relaciones muchos a uno, muchos a muchos, y uno a uno.

1. Relación Muchos a Uno:

- Crea un modelo ‘Categoría’ que represente las categorías a las que pueden pertenecer varios productos.

```
models.py U X
1  from django.db import models
2  from django.contrib.auth.models import User
3
4  class Categoria(models.Model):
5      nombre = models.CharField(max_length=100)
6
7      def __str__(self):
8          return self.nombre
9
```

- Añade un modelo ‘Producto’ que tenga una relación muchos a uno con ‘Categoría’.

```
models.py U X
10 class Producto(models.Model):
11     nombre = models.CharField(max_length=100)
12     categoria = models.ForeignKey(Categoria, on_delete=models.CASCADE)
13
14     def __str__(self):
15         return self.nombre
16
```

2. Relación Muchos a Muchos:

- Crea un modelo 'Etiqueta' que represente las etiquetas que pueden aplicarse a varios productos.

```
models.py U X
17 class Etiqueta(models.Model):
18     nombre = models.CharField(max_length=100)
19     productos = models.ManyToManyField(Producto)
20
21     def __str__(self):
22         return self.nombre
```

- Modifica el modelo 'Producto' para tener una relación muchos a muchos con 'Etiqueta'.

```
10 class Producto(models.Model):
11     nombre = models.CharField(max_length=100)
12     categoria = models.ForeignKey(Categoria, on_delete=models.CASCADE)
13
14     def __str__(self):
15         return self.nombre
```

3. Relación Uno a Uno:

- Crea un modelo 'Perfil' que represente el perfil asociado a un usuario.

```
models.py U X
23
24 class Perfil(models.Model):
25     usuario = models.OneToOneField(User, on_delete=models.CASCADE)
26     nombre_completo = models.CharField(max_length=100)
27     fecha_nacimiento = models.DateField()
28     direccion = models.CharField(max_length=200)
29     telefono = models.CharField(max_length=20)
30
31     def __str__(self):
32         return self.usuario.username
```