

Lección 5: Actividad de portafolio nro. 5

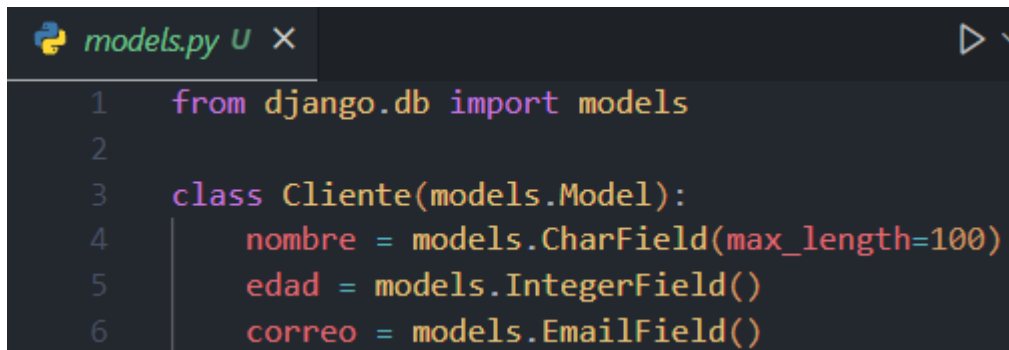
“Consultas personalizadas con Django ORM y SQL directo”

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

Objetivo: Explorar el uso de consultas personalizadas con Django ORM y ejecutar sentencias SQL directamente para recuperar registros de una base de datos.

1. Creación del modelo:

- Crea un modelo 'Cliente' que represente la información de los clientes.

A screenshot of a code editor window titled 'models.py'. The code defines a Django model named 'Cliente' that inherits from 'models.Model'. It has three fields: 'nombre' (CharField with max_length=100), 'edad' (IntegerField), and 'correo' (EmailField).

```
1 from django.db import models
2
3 class Cliente(models.Model):
4     nombre = models.CharField(max_length=100)
5     edad = models.IntegerField()
6     correo = models.EmailField()
```

2. Consultas con Django ORM:

- Realiza consultas utilizando Django ORM para recuperar clientes mayores

de 25 años y con un correo específico.

```
views.py U X
1  from django.shortcuts import render, redirect
2  from django.contrib import messages
3  from .models import Cliente
4  from .forms import ClienteForm
5  from django.db import connection
6
7  def clientes_mayores_con_correo_especifico(request):
8      correo = request.GET.get('correo', '')
9      edad = request.GET.get('edad', '')
10
11     clientes = Cliente.objects.all()
12
13     if edad:
14         try:
15             edad = int(edad)
16             clientes = clientes.filter(edad__gt=edad)
17         except ValueError:
18             pass
19
20     if correo:
21         clientes = clientes.filter(correo=correo)
22
23     return render(request, 'clientes.html', {'clientes': clientes, 'correo': correo, 'edad': edad})
```

3. Ejecución de sentencias SQL directas.

- Utiliza el método `raw()` de Django ORM para ejecutar sentencias SQL directas.

```
25  def consulta_sql_raw(request):
26      if request.method == 'GET':
27          edad = request.GET.get('edad', '')
28          if edad:
29              try:
30                  edad = int(edad)
31                  with connection.cursor() as cursor:
32                      cursor.execute("SELECT * FROM ormsqlapp_cliente WHERE edad > %s", [edad])
33                      clientes = cursor.fetchall()
34              except ValueError:
35                  clientes = []
36          else:
37              clientes = []
38
39      return render(request, 'clientes.html', {'clientes': clientes})
```

4. Invocación a procedimientos almacenados:

- Crea un procedimiento almacenado en tu base de datos y úsalo desde

Django.

```
41 def invocar_procedimiento_almacenado(request):
42     if request.method == 'GET':
43         argumento1 = request.GET.get('argumento1', '')
44         argumento2 = request.GET.get('argumento2', '')
45         if argumento1 and argumento2:
46             with connection.cursor() as cursor:
47                 cursor.callproc('nombre_del_procedimiento', [argumento1, argumento2])
48                 resultados = cursor.fetchall()
49         else:
50             resultados = []
51
52     return render(request, 'resultados.html', {'resultados': resultados})
```