### Páginas de temas individuales

A continuación, necesitamos crear una página que pueda centrarse en un único tema, mostrando el nombre del tema y todas las entradas de ese tema. Definiremos un nuevo patrón URL, escribiremos una vista y crearemos una plantilla. También modificaremos la página de temas para que cada elemento de la lista con viñetas enlace a su correspondiente página de temas.

#### El patrón URL de temas

El patrón URL para la página de temas es un poco diferente de los patrones URL anteriores porque utilizará el atributo id del tema para indicar qué tema se ha solicitado. Por ejemplo, si el usuario quiere ver la página detallada del tema Ajedrez (donde id es 1), la URL será *http://localhost:8000/topics/1/*. Aquí tienes un patrón para que coincida con esta URL, que debes colocar en *learning\_logs/urls.py*:

**learning\_logs/urls.py**

--snip--  
urlpatterns = [  
 --snip--  
 # Detail page for a single topic.  
 path('topics/<int:topic\_id>/', views.topic, name='topic'),  
]

Examinemos la cadena 'topics/<int:topic\_id>/' en este patrón de URL. La primera parte de la cadena indica a Django que busque URL que tengan la palabra *topics* después de la URL base. La segunda parte de la cadena, /<int:topic\_id>/, busca un número entero entre dos barras diagonales y asigna el valor entero a un argumento llamado topic\_id.

Cuando Django encuentra una URL que coincide con este patrón, llama a la función de vista topic() con el valor asignado a topic\_id como argumento. Utilizaremos el valor de topic\_id para obtener el tema correcto dentro de la función.

#### La vista Tema

La función topic() necesita obtener el tema y todas las entradas asociadas de la base de datos, de forma muy similar a lo que hicimos anteriormente en el shell de Django:

**views.py**

--snip--  
❶ def topic(request, topic\_id):  
 """Show a single topic and all its entries."""  
❷ topic = Topic.objects.get(id=topic\_id)  
❸ entries = topic.entry\_set.order\_by('-date\_added')  
❹ context = {'topic': topic, 'entries': entries}  
❺ return render(request, 'learning\_logs/topic.html', context)

Esta es la primera función de la vista que requiere un parámetro distinto del objeto request. La función acepta el valor capturado por la expresión /<int:topic\_id>/ y lo asigna a topic\_id ❶. A continuación, utilizamos get() para recuperar el tema, tal y como hicimos en el intérprete de comandos Django ❷. A continuación, obtenemos todas las entradas asociadas a este tema y las ordenamos según date\_added ❸. El signo menos delante de date\_added ordena los resultados en orden inverso, lo que mostrará primero las entradas más recientes. Almacenamos el tema y las entradas en el diccionario context ❹ y llamamos a render() con el objeto request, la plantilla *topic.html* y el diccionario context ❺.

[anterior](c18_36.html)[Subtema 37 de 41: (Ver todo)](c18.html)[siguiente](c18_38.html)