### Activar modelos

Para utilizar nuestros modelos, tenemos que decirle a Django que incluya nuestra aplicación en el proyecto general. Abre *settings.py* (en el directorio *ll\_project* ); verás una sección que indica a Django qué aplicaciones están instaladas en el proyecto:

**settings.py**

--snip--  
INSTALLED\_APPS = [  
 'django.contrib.admin',  
 'django.contrib.auth',  
 'django.contrib.contenttypes',  
 'django.contrib.sessions',  
 'django.contrib.messages',  
 'django.contrib.staticfiles',  
]  
--snip--

Añade nuestra app a esta lista modificando INSTALLED\_APPS para que tenga este aspecto:

--snip--  
INSTALLED\_APPS = [  
 # My apps.  
 'learning\_logs',  
  
 # Default django apps.  
 'django.contrib.admin',  
 --snip--  
]  
--snip--

Agrupar aplicaciones en un proyecto ayuda a seguirlas cuando el proyecto crece e incluye más aplicaciones. Aquí empezamos una sección llamada My apps, que por ahora sólo incluye 'learning\_logs'. Es importante que coloques tus propias aplicaciones antes de las aplicaciones por defecto, por si necesitas sustituir algún comportamiento de las aplicaciones por defecto por tu propio comportamiento personalizado.

A continuación, tenemos que decirle a Django que modifique la base de datos para que pueda almacenar información relacionada con el modelo Topic. Desde el terminal, ejecuta el siguiente comando:

(ll\_env)learning\_log$ python manage.py makemigrations learning\_logs  
Migrations for 'learning\_logs':  
 learning\_logs/migrations/0001\_initial.py  
 - Create model Topic  
(ll\_env)learning\_log$

El comando makemigrations indica a Django que averigüe cómo modificar la base de datos para que pueda almacenar los datos asociados a los nuevos modelos que hayamos definido. La salida aquí muestra que Django ha creado un archivo de migración llamado *0001\_initial.py*. Esta migración creará una tabla para el modelo Topic en la base de datos.

Ahora aplicaremos esta migración y haremos que Django modifique la base de datos por nosotros:

(ll\_env)learning\_log$ python manage.py migrate  
Operations to perform:  
 Apply all migrations: admin, auth, contenttypes, learning\_logs, sessions  
Running migrations:  
 Applying learning\_logs.0001\_initial... OK

La mayor parte de la salida de este comando es idéntica a la salida de la primera vez que ejecutamos el comando migrate. Tenemos que comprobar la última línea de esta salida, donde Django confirma que la migración para learning\_logs funcionó OK.

Siempre que queramos modificar los datos que gestiona Learning Log, seguiremos estos tres pasos: modificar *models.py*, llamar a makemigrations en learning\_logs, y decirle a Django que migrate el proyecto.

[anterior](c18_17.html)[Subtema 18 de 41: (Ver todo)](c18.html)[siguiente](c18_19.html)