

En urls de catalog he cambiado el `index` para que use la vista `IndexView` que se ha definido en `views.py`. Ahí le estamos diciendo que va a ser una lista de tipo lista del modelo director y que va a usar el template `index`. Al hacer eso `object_list` ya no está vacía y muestra los directores

Para mostrar los detalles de los directores hay que hacer lo mismo, añadir el view en `views.py` y la url en urls de catalog.

Algo no estaba bien, todo iba al `index`, ahora cuando se hace clic en el director, que es un enlace aunque no lo parezca, va a la pagina de detalle del director. Lo que se ha hecho, en urls del proyecto hemos puesto `path('', include('catalog.urls'))` en vez de `path('catalog/', include('catalog.urls'))` que era lo que estaba. Ahora ya no hace falta poner `catalog` en el navegador, el `index` está ahora en la raíz.

Además, `re_path` no funciona, ahora al añadir path de django si lo reconoce.

Si pongo `películas = models.ManyToManyField(Pelicula)`

dentro de `director`, tendré que elegir de entre todas las películas que hay que pueden ser muchísimas.

Es mejor que primero se defina el director y luego que una película tiene un director.

la característica `_set` es para darnos la relación al revés es decir, si tuviéramos una película podríamos saber su director a pesar de que la clase película no tiene campo director.

`.all` Es para darnos todos los campos de un `manyto many` o de un `sql`

el name que se usa en url patterns de la app es el que se usa en la base.html. así podemos decirle con que se tiene que rellenar

Si el `index` en `catalog/urls` tiene el path vacío `''` o sea `/` y luego ponemos que el detalle del director sea `<pk>`, es decir un número de director que es el id del mismo, entonces siempre esperará ese id después de la `/`. Para ir a películas que es como otra rama o forma de llegar a la info, tenemos que poner el listado de directores en un path distinto de `''` y todo lo que depende de él, como el detalle o las películas de ese director harán referencia a ese path, por ejemplo `directores`. De esta forma si definimos un path `/películas` ya no estará esperando un id y te dejará llegar al html.