**Ciclo de Desarrollo Típico para una App en Django**

Pensemos en la aplicación de **Turnos**.**Paso 1: Definir los Modelos (models.py)**

* **Qué es**: El plano o el "molde" de tus datos. Aquí defines qué información quieres guardar en la base de datos (qué campos tendrá una tabla, de qué tipo son, cómo se relacionan entre sí).
* **Tu Proceso**: "inicia con los modelos" -> **¡Correcto!** Es el primer paso porque todo lo demás (vistas, formularios, etc.) dependerá de cómo estructures tus datos.
* **En tu proyecto**: En turnos/models.py, definirías una clase Turno con campos como empleado, fecha\_inicio, hora\_inicio, tipo\_turno, etc.

**Paso 2: Crear y Aplicar las Migraciones (Terminal)**

* **Qué es**: Este es el paso que traduce tus modelos de Python a lenguaje de base de datos.
* python manage.py makemigrations: Django mira tus archivos models.py, los compara con el estado actual de la base de datos y crea un archivo de "instrucciones" para actualizarla.
* python manage.py migrate: Django toma esas instrucciones y las ejecuta, creando o modificando las tablas en tu base de datos (en tu caso, db.sqlite3).
* **Por qué es crucial**: Sin este paso, tus modelos solo existen en el código de Python; la base de datos no sabe nada de ellos.

**Paso 3: (Opcional pero Recomendado) Crear la Lógica de Negocio (services.py)**

* **Qué es**: En lugar de poner toda la lógica compleja en las vistas, la extraemos a funciones o clases en un archivo de "servicios".
* **Por qué**: Mantiene tus vistas limpias. Las vistas deben preocuparse por manejar la petición web (request) y devolver una respuesta (response), no por hacer cálculos complejos. Además, esta lógica se vuelve reutilizable y mucho más fácil de probar.
* **En tu proyecto**: Crearíamos un TurnoService con funciones como crear\_turno(), cancelar\_turno(), buscar\_turnos\_por\_empleado(), etc.

**Paso 4: Crear las Vistas (views.py)**

* **Qué es**: El "cerebro" que conecta tus datos con lo que ve el usuario. Una vista recibe una petición del navegador, utiliza los modelos (o los servicios) para obtener los datos necesarios, y se los pasa a una plantilla para que los muestre.
* **Tu Proceso**: "una vez se crean los modelos, creo las vistas" -> **¡Correcto!** Necesitas los modelos definidos para poder interactuar con ellos en las vistas.
* **En tu proyecto**: Crearías TurnoListView para mostrar una lista de turnos, o TurnoCreateView para mostrar un formulario que permita crear un turno nuevo.

**Paso 5: Crear las Plantillas (templates/nombre\_app/)**

* **Qué es**: El HTML que el usuario final ve en su navegador. Las plantillas reciben los datos que les pasa la vista y los muestran de forma dinámica.
* **En tu proyecto**: Crearías templates/turnos/lista\_turnos.html, que heredaría de tu base.html y usaría un bucle {% for turno in turnos %} para mostrar la información.

**Paso 6: Definir las URLs de la App (turnos/urls.py)**

* **Qué es**: El mapa de tu aplicación. Aquí le dices a Django: "Cuando un usuario vaya a la dirección .../turnos/crear/, ejecuta la vista TurnoCreateView".
* **Tu Proceso**: "una vez creada las vista creo las url de esas vistas" -> **¡Perfectamente correcto!** No puedes apuntar una URL a una vista que todavía no existe.
* **En tu proyecto**: Fue el paso que hicimos para turnos, permisos y solicitudes creando los archivos urls.py que faltaban.

**Paso 7: Conectar las URLs de la App al Proyecto (config/urls.py)**

* **Qué es**: El último paso para que las URLs sean accesibles. Usas include() para "enchufar" el archivo de URLs de tu aplicación (turnos/urls.py) al archivo de URLs principal del proyecto.
* **En tu proyecto**: La línea path('turnos/', include('turnos.urls')) hace exactamente esto.

**En Resumen:**

Tu intuición es muy buena. El ciclo completo y ordenado sería:

1. **Modelos**: Define los datos.
2. **Migraciones**: Crea las tablas en la BD.
3. **Servicios** (Opcional): Escribe la lógica de negocio.
4. **Vistas**: Conecta la lógica con la presentación.
5. **Plantillas**: Muestra los datos al usuario.
6. **URLs de la App**: Crea las rutas para tus vistas.
7. **URLs del Proyecto**: Conecta tu app al resto del sitio.

Es un ciclo: **Datos -> Lógica -> Presentación -> Ruta**.