

Para el desarrollo de esta aplicación de reportes urbanos he utilizado SQLite como sistema de base de datos, ya que viene integrado por defecto con Django y permite trabajar de forma rápida y sencilla durante el desarrollo. Aun así, la estructura del modelo está pensada para que pueda funcionar también con bases de datos más avanzadas como MySQL o PostgreSQL si hiciera falta en producción.

El modelo principal que he creado se llama Reporte, y representa cada incidencia que un usuario puede registrar desde la interfaz. Cada reporte está asociado a un usuario del sistema mediante una relación de tipo ForeignKey con el modelo User que proporciona Django por defecto.

### **Campos del modelo Reporte:**

- **id:** identificador automático generado por Django (clave primaria).
- **título:** campo de texto corto para indicar el nombre del reporte.
- **descripción:** campo de texto más largo donde se detalla la incidencia.
- **ubicación:** cadena de texto con la localización del problema.
- **estado:** número entero que representa el estado del reporte (1 = pendiente, 2 = en proceso, 3 = resuelto).
- **fecha:** se guarda automáticamente la fecha y hora de creación del reporte.
- **imagen:** campo opcional para adjuntar una imagen de la incidencia.
- **usuario:** relación con el modelo User de Django, para saber quién ha creado el reporte.

También se utiliza la tabla auth\_user , que es la que gestiona los usuarios del sistema (incluye nombre de usuario, contraseña cifrada, email, etc.).

### **Relación entre tablas:**

- Un mismo usuario puede crear varios reportes (relación uno a muchos).
- Cada reporte pertenece siempre a un único usuario.

Django genera automáticamente las tablas al ejecutar migrate, por lo que no es necesario crear manualmente un archivo .sql.