

¿Qué pasa si añadimos un nuevo campo? los anteriores registros no lo tendrían, entonces ¿Con qué valor se actualizarán?

Al añadir un nuevo campo a un modelo en Django, los registros existentes en la base de datos no tienen valores para este nuevo campo. Para resolver esto, puedes proporcionar un valor por defecto usando el argumento `default` en el modelo, permitiendo que el campo sea nulo (`null=True`), o interactuar con la consola para proporcionar un valor por defecto durante la migración. Si no se proporciona un valor por defecto y el campo no permite valores nulos, Django solicitará un valor por defecto al ejecutar la migración, y este se aplicará a todos los registros existentes.

```
PS F:\UNSA\3_Semestre\Pweb2\VistasDjango2_RodrigoQ\src\listaContactos> python manage.py makemigrations
It is impossible to add a non-nullable field 'donador' to persona without specifying a default. This is
because the database needs something to populate existing rows.
Please select a fix:
  1) Provide a one-off default now (will be set on all existing rows with a null value for this column)
  2) Quit and manually define a default value in models.py.
Select an option: 1
Please enter the default value as valid Python.
The datetime and django.utils.timezone modules are available, so it is possible to provide e.g. timezone
.now as a value.
Type 'exit' to exit this prompt
>>> True
Migrations for 'personas':
  personas\migrations\0002_persona_donador.py
  - Add field donador to persona
PS F:\UNSA\3_Semestre\Pweb2\VistasDjango2_RodrigoQ\src\listaContactos> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, auth, contenttypes, personas, sessions
Running migrations:
  Applying personas.0002_persona_donador... OK
PS F:\UNSA\3_Semestre\Pweb2\VistasDjango2_RodrigoQ\src\listaContactos>
```

¿Cómo crearía un campo que sea obligatorio?

Para crear un campo obligatorio en Django, debes definir el campo con `null=False` (que es el valor por defecto para la mayoría de los campos) y `blank=False` (que también es el valor por defecto). Esto asegura que el campo no pueda ser NULL en la base de datos y que no se permite dejarlo vacío en formularios. Por ejemplo, `telefono = models.CharField(max_length=15, null=False, blank=False)` define un campo teléfono como obligatorio en el modelo.

¿Cuáles de estos elementos afectarían a la base de datos? ¿Cuáles no?

Los elementos que afectan la base de datos incluyen añadir, eliminar, o modificar un campo, cambiar el tipo de un campo, cambiar las propiedades de un campo, renombrar un campo, y modificar las relaciones entre modelos (como ForeignKey, OneToOneField, ManyToManyField). Estos cambios alteran la estructura de las tablas en la base de datos. Por otro lado, los elementos que no afectan la base de datos son cambiar el orden de los campos, añadir o modificar métodos o propiedades no persistentes, cambiar metadatos del modelo en Meta, y agregar validadores. Estos cambios no impactan la estructura de la base de datos pero pueden influir en el comportamiento y la validación de los datos a nivel de la aplicación.