



Tipos de Campos de un Modelo en Django - Fields in Models



Cristhian Santa Cruz

hace 2 años

En este artículo veamos los campos más utilizados en un **modelo Django**, antes asumimos que implementaremos la base de datos de una biblioteca, por esta razón los ejemplos serán basados en ello.

Toda la información es extraída desde la **documentación oficial de Django**
3: Model field reference
(<https://docs.djangoproject.com/en/3.1/ref/models/fields/>).

models.CharField:

Utilizado para cadenas (String), puedes especificar cadenas pequeñas o grandes, mediante el parámetro max-length

Parámetros:

max-length: Espera un número entero, que indica el tamaño de la cadena.

blank: Espera un valor Booleano, que indica si el campo puede recibir valores vacíos o no.

choices: Espera un arreglo de String, espera un array de parejas, el cual será las únicas opciones para un valor en este campo.

```
class Autor(models.Model):
    full_name = models.CharField(max_length=60)
    adresse = models.CharField(max_length=80, blank=True)
    gender = models.CharField(max_length=80, choices=array_choices)
```

models.BooleanField:

Utilizado para especificar campos de tipo booleano (Verdadero o Falso)

Parámetros:

blank = Especifica si este campo puede o no ser vacío.

default = Especifica el valor por defecto si se intenta ingresar un valor vacío.

```
class Libro(models.Model):
    titulo = models.CharField(max_length=60)
    publicado = models.BooleanField(default=False)
```

models.DateField:

Cursos

Destacados:



vue JS

3612 estudiantes

(/cursos/detail/vue)



3580 estudiantes

(/cursos/detail/django-profesional)



2496 estudiantes

(/cursos/detail/vue-basico)



1422 estudiantes

(/cursos/detail/pinia-vue)

Utilizado para campos que requieran el registro de fechas (objetos de tipo fecha sin hora)

Parámetros:

null: se especifica valor booleano para indicar si puede o no ser vacío.

auto_now: valor booleano que indica que si la fecha se completa automáticamente basándose en el servidor, requiere configuración adicional.

```
class Libro(models.Model):
    titulo = models.CharField(max_length=60)
    publicado = models.BooleanField(default=False)
    fecha_publicacion = models.DateField(null=True)
```

models.DateTimeField:

A diferencia del DateField este aparte registrar la fecha, también registra la hora y trae procesos y métodos extra para manipular los datos de tipo hora.

Parámetros:

null: se especifica valor booleano para indicar si puede o no ser vacío.

auto_now: valor booleano que indica que si la fecha se completa automáticamente basándose en el servidor, requiere configuración adicional.

```
class Libro(models.Model):
    titulo = models.CharField(max_length=60)
    publicado = models.BooleanField(default=False)
    fecha_publicacion = models.DateTimeField(null=True, blank=True)
```

models.EmailField:

Como ya da a suponer el nombre, es un campo exclusivo para cadenas de tipo email, incluye validaciones extra para que solo acepte cadenas con un correcto formato de email. No necesita especificar el tamaño máximo de la cadena.

Parámetros:

blank: se pone en True si deseas que este campo no sea obligatorio

```
class Autor(models.Model):
    full_name = models.CharField(max_length=60)
    correo = models.EmailField(blank=True)
```

models.URLField:

Muy similar al EmailField, recibe una cadena que tenga el formato correcto de una url válida. No es necesario especificar el tamaño máximo de la cadena.

Parámetros:

null: Internamente esto se transforma en un objeto, por ello para que no sea obligatorio se deberá poner la propiedad null=True

```
class Autor(models.Model):
    full_name = models.CharField(max_length=60)
    correo = models.EmailField(blank=True)
    perfil = models.URLField(null=True)
```

models.ImageField:

Se utiliza cuando se desea subir y almacenar imágenes en un servidor, al procesar este atributo se creará un objeto que luego puedes manipular con facilidad, accediendo a la url de la imagen, el tamaño, ruta y otros datos más.

Parámetros:

null: recibe Booleano

blank: recibe Booleano

upload: recibe cadena, ahí puedes especificar el nombre de la carpeta donde deseas que se guarden las imágenes.

NOTA: puedes aplicar parámetros y funciones especial al guardar para procesar imágenes, un ejemplo de ello sería: si deseas comprimir el peso de una imagen para no sobrecargar tu servidor.

```
class Autor(models.Model):
    full_name = models.CharField(max_length=60)
    correo = models.EmailField(blank=True)
    perfil = models.URLField(null=True)
    foto = models.ImageField(null=True, upload='fotos')
```

models.PositiveIntegerField:

Parte del tipo de campo IntegerField, con la diferencia que este solo permitirá almacenar valores positivos enteros. Se puede usar por ejemplo en campos donde almacenes edad.

Parámetros:

default: puedes asignar un valor por defecto o nulo si no deseas que sea obligatorio.

```
class Autor(models.Model):
    full_name = models.CharField(max_length=60)
    correo = models.EmailField(blank=True)
    perfil = models.URLField(null=True)
    foto = models.ImageField(null=True, upload='fotos')
    edad = models.PositiveIntegerField(default=0)
```

¿La respuesta fue útil?