📌 Campos de Modelo en Django y sus Parámetros

🔹 Campos de Texto

`CharField`

Uso: Almacena texto corto con longitud máxima.
Ejemplo:

```
nombre = models.CharField(max_length=255, unique=True, blank=True, null=True)
```

Parámetros:

- `max_length`: (**Obligatorio**) Longitud máxima del texto.
- `unique`: Evita valores duplicados.
- `blank`: Permite que el campo esté vacío en formularios.
- `null`: Permite `NULL` en la base de datos.
- `default`: Valor predeterminado.

`choices`: Lista de opciones predefinidas.

```
ESTADOS = [
    ('A', 'Activo'),
    ('I', 'Inactivo'),
]
estado = models.CharField(max_length=1, choices=ESTADOS, default='A')
```

- `help_text`: Mensaje de ayuda en los formularios.

`TextField`

Uso: Para almacenar texto largo sin límite de caracteres.
Ejemplo:

```
descripcion = models.TextField(blank=True, null=True)
```

Parámetros: `blank`, `null`, `default`, `help_text`.

`SlugField`

Uso: Guarda valores aptos para URLs.
Ejemplo:

```
slug = models.SlugField(unique=True, max_length=100)
```

Parámetros: `max_length`, `unique`, `allow_unicode`.

`EmailField`

Uso: Almacena correos electrónicos con validación automática.
Ejemplo:

```
email = models.EmailField(unique=True)
```

Parámetros: `max_length`, `unique`, `blank`, `null`, `help_text`.

`URLField`

Uso: Almacena URLs y valida su formato.
Ejemplo:

```
website = models.URLField(blank=True, null=True)
```

Parámetros: `max_length`, `blank`, `null`, `help_text`.

🔹 Campos Numéricos

`IntegerField`

Uso: Almacena números enteros.
Ejemplo:

```
edad = models.IntegerField(default=18)
```

Parámetros: `default`, `unique`, `blank`, `null`, `help_text`, `validators`.

`PositiveIntegerField`

Uso: Solo permite números enteros positivos.
Ejemplo:

```
cantidad = models.PositiveIntegerField(default=1)
```

`DecimalField`

Uso: Almacena números decimales con precisión controlada.
Ejemplo:

```
precio = models.DecimalField(max_digits=10, decimal_places=2)
```

Parámetros:

- `max_digits`: Máximo de dígitos (incluyendo decimales).
- `decimal_places`: Cantidad de decimales permitidos.

`FloatField`

Uso: Almacena números flotantes.
Ejemplo:

```
porcentaje = models.FloatField()
```

🔹 Campos de Fecha y Hora

`DateField`

Uso: Almacena solo fechas.
Ejemplo:

```
fecha_nacimiento = models.DateField(auto_now_add=True)
```

Parámetros: `auto_now`, `auto_now_add`.

`TimeField`

Uso: Almacena solo horas.
Ejemplo:

```
hora_inicio = models.TimeField(auto_now_add=True)
```

`DateTimeField`

Uso: Almacena fecha y hora.
Ejemplo:

```
creado_en = models.DateTimeField(auto_now_add=True)
actualizado_en = models.DateTimeField(auto_now=True)
```

🔹 Campos Booleanos

`BooleanField`

Uso: Almacena `True` o `False`.
Ejemplo:

```
activo = models.BooleanField(default=True)
```

🔹 Campos de Archivo e Imagen

`FileField`

Uso: Permite la carga de archivos.
Ejemplo:

```
archivo = models.FileField(upload_to='archivos/')
```

Parámetros: `upload_to`, `max_length`.

`ImageField`

Uso: Similar a `FileField`, pero valida que sea una imagen.
Ejemplo:

```
imagen = models.ImageField(upload_to='imagenes/')
```

Nota: Requiere instalar Pillow:

```
pip install pillow
```

🔹 Campos de Relaciones

`ForeignKey`

Uso: Relación de muchos a uno.
Ejemplo:

```
categoria = models.ForeignKey(Categoria, on_delete=models.CASCADE)
```

Parámetros:

- `on_delete`: Define qué hacer si el objeto relacionado se elimina.
  - `CASCADE`: Borra los registros relacionados.
  - `SET_NULL`: Pone el campo en `NULL`.
  - `PROTECT`: Evita la eliminación si hay relaciones.
- `related_name`: Nombre para acceder desde el modelo relacionado.

`OneToOneField`

Uso: Relación uno a uno.
Ejemplo:

```
perfil = models.OneToOneField(Perfil, on_delete=models.CASCADE)
```

Parámetros: `on_delete`, `primary_key`.

`ManyToManyField`

Uso: Relación muchos a muchos.
Ejemplo:

```
etiquetas = models.ManyToManyField(Etiqueta)
```

Parámetros: `related_name`, `blank`, `help_text`.

🔹 Otros Campos Útiles

`UUIDField`

Uso: Identificadores únicos universales.
Ejemplo:

```
import uuid
id_unico = models.UUIDField(default=uuid.uuid4, editable=False,
unique=True)
```

`JSONField`

Uso: Almacena datos en formato JSON.
Ejemplo:

```
datos_extra = models.JSONField(default=dict)
```

📌 Resumen de Parámetros Comunes

Parámetr	Descripción
`max_lengt`	Longitud máxima del campo.
`unique`	Evita valores duplicados.

`blank`	Permite que el campo esté vacío en formularios.
`null`	Permite `NULL` en la base de datos.
`default`	Valor por defecto.
`choices`	Opciones predefinidas en una lista de tuplas.
`help_text`	Texto de ayuda en Django Admin.
`validators`	Validaciones personalizadas.
`on_delete`	Qué hacer si el objeto relacionado se borra.
`upload_to`	Ubicación de archivos en `FileField` e `ImageField`.
