

```
from django import forms
from .models import Boletin, PlantillaBoletin, Fuente
class PlantillaBoletinForm(forms.ModelForm):
    archivo_html = forms.FileField(required=False, help_text="Solo se permiten archivos .html")
    class Meta:
        model = PlantillaBoletin
        fields = ['nombre', 'contenido_html', 'archivo_html']
        widgets = { 'contenido_html': forms.Textarea(attrs={'rows': 20, 'cols': 100}), }
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        # Hacemos que el contenido_html no sea obligatorio en el formulario
        self.fields['contenido_html'].required = False
    def clean_archivo_html(self):
        archivo = self.cleaned_data.get('archivo_html')
        if archivo:
            if not archivo.name.lower().endswith('.html'):
                raise forms.ValidationError("Solo se permiten archivos con extensión .html")
            return archivo
        def clean(self):
            cleaned_data = super().clean()
            contenido = cleaned_data.get('contenido_html')
            archivo = cleaned_data.get('archivo_html')
            if not contenido and not archivo:
                raise forms.ValidationError("Debe proporcionar contenido HTML directamente o subir un archivo .html.")
            return cleaned_data
class CrearBoletinForm(forms.ModelForm):
    plantilla = forms.ModelChoiceField(queryset=PlantillaBoletin.objects.all(), required=False, label="Plantilla disponible")
    contenido_html = forms.CharField(widget=forms.Textarea(attrs={'rows': 20, 'cols': 80}), label="Contenido HTML editable", required=False)
    class Meta:
        model = Boletin
        fields = ['titulo', 'ciudad_tratada', 'tematica', 'fuente_boletin']
    def __init__(self, *args, **kwargs):
        super().__init__(*args, **kwargs)
        self.fields['fuente_boletin'].queryset = Fuente.objects.filter(fuente_activa=True)
        self.fields['fuente_boletin'].label_from_instance = lambda obj: obj.titulo
```