

¿Qué son los formularios en Django?

Los formularios en Django te permiten recopilar y validar datos del usuario. Django tiene dos formas principales de trabajar con formularios:

1. **forms.Form** – Usado para formularios simples no ligados a un modelo.
 2. **forms.ModelForm** – Usado para crear formularios basados en modelos de base de datos.
-

1. Crear un modelo

```
from django.db import models

class Contacto(models.Model):
    nombre = models.CharField(max_length=100)
    email = models.EmailField()
    mensaje = models.TextField()

    def __str__(self):
        return self.nombre
```

Luego aplica las migraciones:

```
python manage.py makemigrations
python manage.py migrate
```

2. Crear un formulario

Opción A: Usando **forms.Form** (no ligado a modelo)

Archivo: `mi_app/forms.py`

```
from django import forms

class FormularioContacto(forms.Form):
    nombre = forms.CharField(max_length=100, label="Tu nombre")
    email = forms.EmailField(label="Tu correo")
    mensaje = forms.CharField(widget=forms.Textarea, label="Mensaje")
```

Opción B: Usando **forms.ModelForm**

```
from django import forms
from .models import Contacto

class FormularioContacto(forms.ModelForm):
    class Meta:
        model = Contacto
        fields = ['nombre', 'email', 'mensaje']
```

3. Crear la vista

Archivo: `mi_app/views.py`

```
from django.shortcuts import render, redirect
from .forms import FormularioContacto

def contacto_view(request):
    if request.method == 'POST':
        form = FormularioContacto(request.POST)
        if form.is_valid():
            form.save() # Guarda en la base de datos si es ModelForm
            return redirect('gracias')
        else:
            form = FormularioContacto()

    return render(request, 'contacto.html', {'form': form})
```

4. Crear plantilla HTML

Archivo: `mi_app/templates/contacto.html`

```
<!DOCTYPE html>
<html>
<head>
    <title>Formulario de Contacto</title>
</head>
<body>
    <h1>Contáctanos</h1>
    <form method="post">
        {% csrf_token %}
        {{ form.as_p }}
        <button type="submit">Enviar</button>
    </form>
</body>
</html>
```

5. Configurar URLs

Archivo: `mi_app/urls.py`

```
from django.urls import path
from .views import contacto_view

urlpatterns = [
    path('contacto/', contacto_view, name='contacto'),
]
```

Y en `mi_proyecto/urls.py`:

```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('mi_app.urls')),
]
```

Extras: Personalización y validación

Campos personalizados en `ModelForm`

```
class FormularioContacto(forms.ModelForm):
    class Meta:
        model = Contacto
        fields = ['nombre', 'email', 'mensaje']
        widgets = {
            'mensaje': forms.Textarea(attrs={'rows': 4, 'cols': 40}),
        }
        labels = {
            'nombre': 'Nombre completo',
        }
```

Validación personalizada

```
def clean_email(self):
    email = self.cleaned_data.get('email')
    if not email.endswith('@ejemplo.com'):
        raise forms.ValidationError("El correo debe terminar en @ejemplo.com")
    return email
```