**Resumen rápido de la lógica de envío de correo en el formulario de contacto**

1. **Configuración SMTP en settings.py**
   * Usamos EMAIL\_BACKEND = 'django.core.mail.backends.smtp.EmailBackend' y apuntamos a smtp.gmail.com con TLS.
   * Definimos EMAIL\_HOST\_USER = 'tucuenta@gmail.com' y EMAIL\_HOST\_PASSWORD = 'TU\_APP\_PASSWORD\_16CARACT.' (App Password de Google).
   * Ajustamos DEFAULT\_FROM\_EMAIL para que sea nuestro remitente.
2. **Formulario con Django Forms**
   * Creamos ContactForm en main/forms.py con campos nombre, email y mensaje.
   * Esto nos da validación automática y helpers para renderizar en plantilla.
3. **Vista en views.py**
   * En contacto(request) procesamos GET/POST:
     + **GET** → renderizamos el formulario vacío.
     + **POST** → instanciamos form = ContactForm(request.POST), validamos con form.is\_valid().
   * Si es válido, extraemos nombre, email, mensaje y construimos un message con saltos de línea (\n).
   * Llamamos a send\_mail(subject, message, from\_email, [recipient\_list]).
   * Añadimos un aviso al usuario con messages.success() o messages.error() y redirigimos (Post-Redirect-Get).
4. **Enrutado en urls.py**
   * Definimos path('contacto/', views.contacto, name='contacto') para exponer la vista.
5. **Plantilla contacto.html y manejo de mensajes**
   * Mostramos el formulario usando {{ form.field }} y mostramos errores inline.
   * Al inicio de base.html (o justo antes del bloque {% block contenido %}) insertamos:

django

CopyEdit

{% if messages %}

{% for msg in messages %}

<div class="alert alert-{{ msg.tags }} alert-dismissible fade show">

{{ msg }} <button class="btn-close" data-bs-dismiss="alert"></button>

</div>

{% endfor %}

{% endif %}

* + Esto renderiza la confirmación de “¡Tu mensaje ha sido enviado!” o posibles errores.

1. **Prueba final**
   * Levantar runserver, rellenar y enviar el formulario en /contacto/.
   * Ver el alert de éxito y comprobar que llega el correo a la cuenta configurada.

Con estos pasos tienes el **flujo completo**: desde la configuración de correo, pasando por el formulario validado, hasta el envío real usando send\_mail y el feedback al usuario con el framework de mensajes de Django.