**1. Modelos**

Primero, definirías tus modelos en models.py según la estructura que discutimos:

python

Copiar código

from django.db import models

class Formulario(models.Model):

estado = models.CharField(max\_length=100)

fecha\_creacion = models.DateTimeField(auto\_now\_add=True)

fecha\_revision = models.DateTimeField(null=True, blank=True)

fecha\_impresion = models.DateTimeField(null=True, blank=True)

class Comentario(models.Model):

contenido = models.TextField()

fecha = models.DateTimeField(auto\_now\_add=True)

formulario = models.ForeignKey(Formulario, related\_name='comentarios', on\_delete=models.CASCADE)

tipo = models.CharField(max\_length=20, choices=[('creación', 'Creación'), ('revisión', 'Revisión'), ('impresión', 'Impresión')])

class Archivo(models.Model):

nombre = models.CharField(max\_length=255)

ruta = models.FileField(upload\_to='archivos/')

formulario = models.ForeignKey(Formulario, related\_name='archivos', on\_delete=models.CASCADE)

tipo = models.CharField(max\_length=20, choices=[('creación', 'Creación'), ('revisión', 'Revisión'), ('impresión', 'Impresión')])

class Revision(models.Model):

formulario = models.OneToOneField(Formulario, on\_delete=models.CASCADE)

fecha\_revision = models.DateTimeField(auto\_now\_add=True)

resultado = models.CharField(max\_length=100)

class Impresion(models.Model):

formulario = models.OneToOneField(Formulario, on\_delete=models.CASCADE)

fecha\_impresion = models.DateTimeField(auto\_now\_add=True)

**2. Migraciones**

Después de definir tus modelos, necesitarás ejecutar las migraciones:

bash

Copiar código

python manage.py makemigrations

python manage.py migrate

**3. Vistas**

En views.py, puedes crear vistas para manejar la lógica de tus formularios y la consulta de archivos y comentarios:

python

Copiar código

from django.shortcuts import render, get\_object\_or\_404

from .models import Formulario

def formulario\_detail(request, id):

formulario = get\_object\_or\_404(Formulario, id=id)

comentarios = formulario.comentarios.all()

archivos = formulario.archivos.all()

revision = formulario.revision if hasattr(formulario, 'revision') else None

impresion = formulario.impresion if hasattr(formulario, 'impresion') else None

return render(request, 'formulario\_detail.html', {

'formulario': formulario,

'comentarios': comentarios,

'archivos': archivos,

'revision': revision,

'impresion': impresion,

})

**4. Templates**

En tu plantilla formulario\_detail.html, puedes mostrar la información:

html

Copiar código

<h1>Formulario {{ formulario.id }}</h1>

<h2>Comentarios de Creación</h2>

<ul>

{% for comentario in comentarios %}

<li>{{ comentario.fecha }}: {{ comentario.contenido }}</li>

{% endfor %}

</ul>

<h2>Archivos de Creación</h2>

<ul>

{% for archivo in archivos %}

<li>{{ archivo.nombre }} - <a href="{{ archivo.ruta.url }}">Descargar</a></li>

{% endfor %}

</ul>

{% if revision %}

<h2>Revisión</h2>

<p>{{ revision.resultado }} el {{ revision.fecha\_revision }}</p>

<h3>Comentarios de Revisión</h3>

<ul>

{% for comentario in revision.comentarios.all %}

<li>{{ comentario.fecha }}: {{ comentario.contenido }}</li>

{% endfor %}

</ul>

<h3>Archivos de Revisión</h3>

<ul>

{% for archivo in revision.archivos.all %}

<li>{{ archivo.nombre }} - <a href="{{ archivo.ruta.url }}">Descargar</a></li>

{% endfor %}

</ul>

{% endif %}

{% if impresion %}

<h2>Impresión</h2>

<p>Imprimido el {{ impresion.fecha\_impresion }}</p>

<h3>Comentarios de Impresión</h3>

<ul>

{% for comentario in impresion.comentarios.all %}

<li>{{ comentario.fecha }}: {{ comentario.contenido }}</li>

{% endfor %}

</ul>

<h3>Archivos de Impresión</h3>

<ul>

{% for archivo in impresion.archivos.all %}

<li>{{ archivo.nombre }} - <a href="{{ archivo.ruta.url }}">Descargar</a></li>

{% endfor %}

</ul>

{% endif %}

**5. Rutas**

Asegúrate de agregar la ruta a tu archivo urls.py:

python

Copiar código

from django.urls import path

from .views import formulario\_detail

urlpatterns = [

path('formulario/<int:id>/', formulario\_detail, name='formulario\_detail'),

]

**Entidades y Relaciones**

1. **Formulario**
   * id\_formulario (PK)
   * estado
   * fecha\_creacion
   * fecha\_revision
   * fecha\_impresion
2. **Comentario**
   * id\_comentario (PK)
   * contenido
   * fecha
   * id\_formulario (FK, referencia a Formulario)
   * tipo (puede ser "creación", "revisión", "impresión")
3. **Archivo**
   * id\_archivo (PK)
   * nombre
   * ruta
   * id\_formulario (FK, referencia a Formulario)
   * tipo (puede ser "creación", "revisión", "impresión")
4. **Revision**
   * id\_revision (PK)
   * id\_formulario (FK, referencia a Formulario)
   * fecha\_revision
   * resultado (puede ser "aprobado", "rechazado", etc.)
5. **Impresion**
   * id\_impresion (PK)
   * id\_formulario (FK, referencia a Formulario)
   * fecha\_impresion

**Relaciones**

* Un **Formulario** puede tener múltiples **Comentarios** y **Archivos** asociados.
* Un **Formulario** puede tener una **Revisión** asociada, y cada revisión puede tener sus propios **Comentarios** y **Archivos**.
* Un **Formulario** también puede tener una **Impresión** asociada, que igualmente puede tener sus **Comentarios** y **Archivos**.

**Diagrama ER Simplificado**

scss

Copiar código

Formulario (1) --- (N) Comentario

Formulario (1) --- (N) Archivo

Formulario (1) --- (1) Revision

Revision (1) --- (N) Comentario

Revision (1) --- (N) Archivo

Formulario (1) --- (1) Impresion

Impresion (1) --- (N) Comentario

Impresion (1) --- (N) Archivo

**Consideraciones**

* Cada entidad tiene su propia tabla, lo que permite una gestión modular y flexible.
* Utiliza campos de tipo datetime para almacenar fechas.
* Puedes agregar índices y restricciones según las necesidades de tu aplicación.
* Piensa en la normalización para evitar redundancias.

Con esta estructura, podrás llevar un control claro de los formularios, revisiones, impresiones, archivos y comentarios asociados, permitiendo un seguimiento adecuado a lo largo del proceso.