Práctica Guiada 1: Inicia tu sistema de PLN con Django

🎯 Objetivo

Que el alumno cree un proyecto base en Django para su sistema de PLN, configure el entorno de trabajo usando Pipenv, cree su primer modelo y vista funcional, y prepare la base para futuras prácticas.

☑️ Requisitos previos

- Tener Python 3.8 o superior instalado.

- Tener Git instalado.

📦 Parte 1: Crear y configurar el entorno virtual con Pipenv

Usar un entorno virtual permite mantener separadas las dependencias de este proyecto respecto a otros en tu máquina. Pipenv es una herramienta que automatiza la creación del entorno virtual y la gestión de dependencias, y es ampliamente utilizada en proyectos profesionales.

1. Instalar Pipenv (si no lo tienes):

```
pip install pipenv
```

2. Crear la carpeta del proyecto y entrar en ella:

```
mkdir sistema_pln
cd sistema_pln
```

3. Crear entorno virtual e instalar Django:

```
pipenv install django
```

4. Activar el entorno virtual:

```
pipenv shell
```

🚀 Parte 2: Inicializar proyecto Django

Una vez activado el entorno virtual, puedes crear tu proyecto Django. Se recomienda nombrar la configuración principal como `config` para mantener buenas prácticas.

1. Iniciar el proyecto Django:

```
django-admin startproject config .
```

2. Crear una app llamada 'analisis' que será la base del sistema:

```
python manage.py startapp analisis
```

3. Agregar la app al archivo de configuración `config/settings.py`:

```
INSTALLED_APPS = [
    ...
    'analisis',
]
```

4. Realizar la primera migración de base de datos:

```
python manage.py migrate
```

Parte 3: Crear el modelo para subir textos

En esta parte se crea el modelo que permitirá almacenar los textos que serán analizados. El archivo puede ser de tipo `.txt` y se almacenará en una carpeta llamada `textos/` dentro de `media/`.

Archivo: `analisis/models.py`

```
from django.db import models

class TextoAnalizado(models.Model):
    titulo = models.CharField(max_length=200)
    archivo = models.FileField(upload_to='textos/')
    fecha_subida = models.DateTimeField(auto_now_add=True)

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

Parte 4: Crear el formulario y vistas

Se requiere un formulario para que el usuario pueda subir textos desde una interfaz gráfica. Este formulario será manejado por una vista que recibe y procesa los datos.

Archivo: `analisis/forms.py`

```
from django import forms
from .models import TextoAnalizado

class TextoAnalizadoForm(forms.ModelForm):
    class Meta:
```

```
        model = TextoAnalizado
        fields = ['titulo', 'archivo']
```

Archivo: `analisis/views.py`

```
from django.shortcuts import render, redirect
from .forms import TextoAnalizadoForm
from .models import TextoAnalizado

def subir_texto(request):
    if request.method == 'POST':
        form = TextoAnalizadoForm(request.POST, request.FILES)
        if form.is_valid():
            form.save()
            return redirect('lista_textos')
    else:
        form = TextoAnalizadoForm()
    return render(request, 'analisis/subir.html', {'form': form})

def lista_textos(request):
    textos = TextoAnalizado.objects.all().order_by('-fecha_subida')
    return render(request, 'analisis/lista.html', {'textos': textos})
```

🌐 Parte 5: Configurar URLs y media

Se definen las rutas de acceso para las vistas de carga y visualización. Además, se configura Django para servir archivos de texto desde una carpeta especial (`media/`).

Archivo: `analisis/urls.py`

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

urlpatterns = [
    path('subir/', views.subir_texto, name='subir_texto'),
    path('', views.lista_textos, name='lista_textos'),
]
```

Archivo: `config/urls.py`

```
from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

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

```
if settings.DEBUG:
    urlpatterns += static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)
```

Archivo: `config/settings.py` (al final)

```
import os

MEDIA_URL = '/media/'
MEDIA_ROOT = os.path.join(BASE_DIR, 'media')
```