

Proyecto: Calculadora Sencilla

Objetivo: Crear una calculadora que pueda realizar operaciones básicas como suma, resta, multiplicación y división, utilizando Django como backend, Python para la lógica, y HTML/CSS para la interfaz.

1. Configuración del Proyecto Django

Primero, asegúrenos de que tienes Django instalado y configuras tu proyecto:

```
pip install django
django-admin startproject calculadora
cd calculadora
```

Ahora, crea una aplicación dentro de tu proyecto:

```
python manage.py startapp operaciones
```

2. Configurar la Aplicación en el Proyecto

En el archivo `settings.py`, debes registrar la aplicación que acabas de crear, añadiéndola a la lista de `INSTALLED_APPS`:

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

3. Crear la Vista de la Calculadora

Vamos a definir la vista que controlará la lógica de la calculadora. Este código se encuentra en el archivo `operaciones/views.py`:

```
from django.shortcuts import render

def calculadora_view(request):
    resultado = None
    if request.method == 'POST': # Verifica si la solicitud es POST
        numero1 = float(request.POST.get('numero1')) # Obtiene el primer número del formulario
        numero2 = float(request.POST.get('numero2')) # Obtiene el segundo número del formulario
        operacion = request.POST.get('operacion') # Obtiene la operación seleccionada

        # Realiza la operación seleccionada
        if operacion == 'suma':
            resultado = numero1 + numero2
        elif operacion == 'resta':
            resultado = numero1 - numero2
        elif operacion == 'multiplicacion':
            resultado = numero1 * numero2
        elif operacion == 'division':
            if numero2 != 0:
                resultado = numero1 / numero2
            else:
```

```
        resultado = 'Error: División por cero'

# Renderiza la plantilla HTML, pasando el resultado
return render(request, 'calculadora.html', {'resultado': resultado})
```

Explicación:

- `request.method == 'POST'` : Verifica si la solicitud es un envío de formulario (POST).
- `request.POST.get('nombre_del_campo')` : Extrae los valores de los campos del formulario.
- `resultado` : Dependiendo de la operación seleccionada, se realiza el cálculo correspondiente y se almacena en esta variable.
- `render()` : Renderiza la plantilla HTML y pasa el resultado al contexto.

4. Crear la Plantilla HTML

Ahora, necesitamos una plantilla para la interfaz de la calculadora. Dentro de la carpeta `operaciones`, crea una carpeta llamada `templates`, y dentro de ella, un archivo `calculadora.html` con el siguiente contenido:

```
<!DOCTYPE html>
<html lang="es">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Calculadora Sencilla</title>
</style>
    body {
        font-family: Arial, sans-serif;
        margin: 50px;
        background-color: #f0f0f0;
        text-align: center;
    }
    .container {
        background-color: white;
        padding: 20px;
        border-radius: 8px;
        box-shadow: 0px 0px 10px rgba(0, 0, 0, 0.1);
        max-width: 300px;
        margin: auto;
    }
    input, select, button {
        width: 100%;
        padding: 10px;
        margin: 10px 0;
        border-radius: 4px;
        border: 1px solid #ccc;
    }
    button {
        background-color: #4CAF50;
        color: white;
        border: none;
        cursor: pointer;
```

```

    }
    button: hover {
        background-color: #45a049;
    }
}
</style>
</head>
<body>
    <div class="container">
        <h1>Calculadora</h1>
        <form method="post">
            {% csrf_token %} <!-- Token de seguridad requerido por Django -->
            <input type="number" name="numero1" placeholder="Número 1" required>
            <input type="number" name="numero2" placeholder="Número 2" required>
            <select name="operacion">
                <option value="suma">Suma</option>
                <option value="resta">Resta</option>
                <option value="multiplicacion">Multiplicación</option>
                <option value="division">División</option>
            </select>
            <button type="submit">Calcular</button>
        </form>
        {% if resultado is not None %}
            <h2>Resultado: {{ resultado }}</h2>
        {% endif %}
    </div>
</body>
</html>

```

Explicación:

- **HTML:** Estructura básica de la página. Incluye un formulario con dos campos de entrada (`input`), un menú desplegable (`select`) para elegir la operación, y un botón de envío (`button`).
- **CSS:** Estiliza la interfaz, centrando el contenido y dándole un aspecto limpio. Aquí se aplica el diseño responsivo con `max-width` y `background-color` para mejorar la estética.
- `{% csrf_token %}`: Es un token de seguridad que Django requiere en los formularios para prevenir ataques CSRF.

5. Configurar las URL

En `operaciones/urls.py`, define la ruta para la vista de la calculadora:

```

from django.urls import path
from .views import calculadora_view

urlpatterns = [
    path('', calculadora_view, name='calculadora'),
]

```

Luego, en `urls.py` del proyecto principal, incluye las rutas de la aplicación:

```

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

```

```
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', include('operaciones.urls')), # Incluye las rutas de la app operaciones  
]
```

6. Ejecutar el Servidor

Ahora, ejecuta el servidor de desarrollo de Django para ver la calculadora en acción:

```
python manage.py runserver
```

Dirígete a `http://localhost:8000` en tu navegador. Deberías ver la calculadora y poder realizar operaciones básicas.

Resumen

- **Django:** Actúa como el framework backend que gestiona las solicitudes HTTP, la lógica de negocio y el enrutamiento.
- **Python:** Se encarga de la lógica de cálculo en la vista.
- **HTML:** Define la estructura de la interfaz de usuario.
- **CSS:** Aplica el diseño visual para hacer que la interfaz sea agradable y fácil de usar.

Este proyecto es un excelente punto de partida para familiarizarte con el desarrollo web usando Django, y puedes seguir construyendo sobre esta base para proyectos más avanzados. Si tienes preguntas o quieres expandir esta calculadora, ¡estoy aquí para ayudarte!