

ENVIAR DATOS CON POST

Ejemplos de algunos controles de FORMULARIOS

Página de Saludo

Introduzca su nombre

Dame un número

Fecha

Rango

Deporte favorito ☐ Futbol ☐ Petanca ☐ Curling

☒ ¿Te gusta Python?

CODIGO HTML

```
<form method="post">
  {% csrf_token %}
  <label>Introduzca su nombre</label>
  <input type="text" name="cajanombre" required/>
  <br/>
  <label>Dame un número</label>
  <input type="number" name="cajanumero"/>
  <br/>
  <label>Fecha</label>
  <input type="date"/>
  <br/>
  <label>Rango</label>
  <input type="range" min="1" max="29" value="1"/>
  <br/>
  <label>Deporte favorito</label>
  <input type="radio" name="deporte" value="Futbol"/>Futbol
  <input type="radio" name="deporte" value="Petanca"/>Petanca
  <input type="radio" name="deporte" value="Curling"/>Curling
  <br/>
  <input type="checkbox"/>¿Te gusta Python?
  <button>
    Enviar datos
  </button>
</form>
```

- Enviar datos mediante POST se realiza mediante formularios, es decir, con etiquetas **<form>**.
- Dentro de un formulario se pueden incluir múltiples controles, entre ellos, el más común son cajas de texto, llamadas **<input>** en HTML.
- Para recibir información con POST se realiza con la siguiente sintaxis:

```
request.POST['variable']
```

- Al tener formularios en **django**, debemos incluir la siguiente instrucción dentro del formulario por motivos de seguridad:

```
<form method="post">
    {% csrf_token %}
```

- Vamos a crear un formulario donde pediremos un nombre y lo pintaremos en pantalla al pulsar un botón.

Su nombre es Alumno

- 1) Comenzamos creando una nueva página llamada **saludo.html** dentro de **templates/aplicación** y dibujamos un formulario con una caja llamada **cajanombre**.

SALUDO.HTML

```
<form method="post">
    {% csrf_token %}
    <input type="text" name="cajanombre"
    class="form-control"/>
    <button class="btn btn-info">
        Enviar nombre
    </button>
</form>
<h1>
    Su nombre es {{nombre}}
</h1>
```

- 2) Recibimos el valor del color dentro de **views.py**

Para recibir un valor debemos utilizar la siguiente sintaxis:

request.POST["variable"]

```
def saludo(request):
    # PREGUNTAMOS SI RECIBIMOS COLOR O NO...
    if ('cajanombre' in request.POST):
        nombreRecibido = request.POST['cajanombre']
        context = {
            'nombre': nombreRecibido
        }
```

```
        return render(request, "aplicacion/saludo.html",
context)
    else:
        #No enviamos nada
        return render(request, "aplicacion/saludo.html")
```

3) Incluimos dentro de **urls.py** la nueva ruta

```
from aplicacion import views
```

```
urlpatterns=[
    path('',views.index,name='index'),
    path('peliculas/',views.peliculas,name='peliculas'),
    path('test/',views.test,name='test'),
    path('futbol/',views.futbol,name='futbol'),
    path('jugadores/',views.jugadores,name='jugadores'),
    path('colores/',views.colores,name='colores'),
    path('saludo/',views.saludo,name='saludo'),
]
```

4) Dentro de la página **Layout.html** ponemos el nuevo Link

```
<li class="nav-item">
    <a class="nav-link" href="/saludo">Saludo POST</a>
</li>
```

A continuación, vamos a Sumar dos números
Comprobaremos que hemos enviado una variable **suma** para realizar el dibujo.
Para comprobar si hemos recibido algún dato en el **template** se utiliza el siguiente código

```
{% if variable%}
    NUESTRO DIBUJO
{% endif %}
```

Número 1

5

Número 2

2

Sumar números

La suma es 7

Sobre **templates/aplicación** creamos una nueva página llamada **sumarnumeros.html**

SUMARNUMEROS.HTML

```
<form method="post">
    {% csrf_token %}
    <label>Número 1</label>
    <input type="number" name="cajanumero1"
    class="form-control"/>
    <label>Número 2</label>
    <input type="number" name="cajanumero2"
    class="form-control"/>
    <button class="btn btn-info">
        Sumar números
    </button>
</form>
{% if suma%}
<h1 style="color:blue">
    La suma es {{suma}}
</h1>
{% endif %}
```

El siguiente paso es crear un nuevo método dentro de **views.py** que reciba los valores de **cajanumero1** y **cajanumero2**

VIEWS.PY

```
def sumarNumeros(request):
    if ('cajanumero1' in request.POST):
```

```
dato1 = request.POST['cajanumero1']
dato2 = request.POST['cajanumero2']
suma = int(dato1) + int(dato2)
context = {
    'suma': suma
}
return render(request, "aplicacion/
sumarnumeros.html", context)
else:
    #No enviamos nada
    return render(request, "aplicacion/
sumarnumeros.html")
```

Incluimos nuestra nueva ruta/path dentro de **urls.py** de **aplicacion**

URLS.PY

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

urlpatterns=[
    path('',views.index,name='index'),

    path('peliculas/',views.peliculas,name='peliculas'),
    path('futbol/',views.futbol,name='futbol'),

    path('jugadores/',views.jugadores,name='jugadores'),
    path('colores/',views.colores,name='colores'),
    path('saludo/',views.saludo,name='saludo'),
    path('sumar/',views.sumarNumeros,name='sumar'),

]
```

El siguiente paso será dibujar la conjetura de Collatz.

Recibiremos un número y debemos devolver una lista con todos los números de la conjetura

Recordamos que la conjetura de Collatz dice así:

Todo número natural siempre llegará a ser 1 siguiendo estas instrucciones:

- Si el número es Par, dividimos entre 2
- Si el número es Impar, multiplicamos por 3 y sumamos 1

Ejemplo con el número 6:

- 6,3,10,5,16,8,4,2,1

Conjetura de Collatz

Introduzca número

Collatz

- 3
- 10
- 5
- 16
- 8
- 4
- 2
- 1

Sobre **templates/aplicación** creamos una nueva página llamada **collatz.html**

COLLATZ.HTML

```
<h2 style="color:blue">
    Conjetura de Collatz
</h2>
<form method="post">
    {% csrf_token %}
    <label>Introduzca número</label>
    <input type="text" name="cajanumero"
    class="form-control"/>
    <button class="btn btn-warning">
        Collatz
    </button>
</form>
<ul>
    {% for num in listacollatz %}
        <li>{{ num }}</li>
    {% endfor %}
</ul>
```

Creamos un nuevo método dentro de **views.py**

VIEWS.PY

```
def collatz(request):
    if ('cajanumero' in request.POST):
        dato = request.POST['cajanumero']
        numero = int(dato)
        listacollatz = []
        while (numero != 1):
            if (numero % 2 == 0):
                #PAR
                numero = int(numero / 2)
            else:
                #IMPAR
                numero = int(numero * 3 + 1)
            listacollatz.append(numero)
        context = {
            'listacollatz': listacollatz
        }
        return render(request, "aplicacion/
collatz.html", context)
    else:
        #No enviamos nada
        return render(request, "aplicacion/
collatz.html")
```

Por último, incluimos una nueva ruta/path dentro de **urls.py** de aplicación

URLS.PY

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

urlpatterns=[
    path('',views.index,name='index'),

    path('peliculas/',views.peliculas,name='peliculas'),
    path('sumar/',views.sumarNumeros,name='sumar'),
    path('futbol/',views.futbol,name='futbol'),

    path('jugadores/',views.jugadores,name='jugadores'),
    path('colores/',views.colores,name='colores'),
    path('saludo/',views.saludo,name='saludo'),
```

```
path('collatz/', views.collatz, name='collatz'),  
]
```

Necesito una página que muestre la Tabla de multiplicar de un número.

Página: **tabla.html**

Views.py: **def tablaMultiplicar(request)**

Urls.py: path **tabla/**

Tendremos una caja de texto, escribiremos un número y mostraremos en una tabla el resultado de las operaciones de la tabla de multiplicar de dicho número.

VERSION 1

Número 5

RESULTADO
5
10
15
...
50

VERSION 2

OPERACION	RESULTADO
5 * 1	5
5 * 2	10
5 * 3	15
....
5 * 10	50

VERSION 1

VIEWS.PY

```
def tablaMultiplicar(request):  
    if ('cajanumero' in request.POST):  
        dato = request.POST["cajanumero"]  
        numero = int(dato)  
        listaTabla = []  
        for i in range(10):  
            resultado = numero * (i + 1)  
            listaTabla.append(resultado)  
        context = {  
            "listatabla": listaTabla  
        }  
        return render(request, 'informacion/tabla.html', context)  
    else:  
        return render(request, 'informacion/tabla.html')
```

TABLA.HTML

```
<!DOCTYPE html>  
<html lang="en">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-  
scale=1.0">  
    <title>Document</title>  
</head>  
<body>  
    {% include 'informacion/menu.html' %}
```



```

<h1>Tabla de multiplicar</h1>
<form method="post">
    {% csrf_token %}
    <label>Introduzca número</label>
    <input type="number" name="cajanumero"/>
    <button>
        Mostrar tabla
    </button>
</form>
<table border="1">
    <tr>
        <th>RESULTADO</th>
    </tr>
    {% for num in listatabla %}
    <tr>
        <td>{{num}}</td>
    </tr>
    {% endfor %}
</table>
</body>
</html>

```

VERSION 2

VIEWS.PY

```

def tablaMultiplicar(request):
    if ('cajanumero' in request.POST):
        dato = request.POST["cajanumero"]
        numero = int(dato)
        listaTabla = []
        for i in range(10):
            resultado = numero * (i + 1)
            operación = str(numero) + " * " + str((i + 1))
            listaTabla.append({
                "operación": operación,
                "resultado": resultado
            })
        context = {
            "listatabla": listaTabla
        }
        return render(request, 'informacion/tabla.html', context)
    else:
        return render(request, 'informacion/tabla.html')

```

TABLA.HTML

```

<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Document</title>
</head>
<body>
    {% include 'informacion/menu.html' %}
    <h1>Tabla de multiplicar</h1>
    <form method="post">
        {% csrf_token %}

```

```
<label>Introduzca número</label>
<input type="number" name="cajanumero"/>
<button>
    Mostrar tabla
</button>
</form>
<table border="1">
  <tr>
    <th>OPERACION</th>
    <th>RESULTADO</th>
  </tr>
  {% for dato in listatabla %}
  <tr>
    <td>{{dato.operacion}}</td>
    <td>{{dato.resultado}}</td>
  </tr>
  {% endfor %}
</table>
</body>
</html>
```