

Investigación sobre Formularios en Django

Los formularios en HTML, son la columna vertebral de los sitios Web interactivos, los encontramos como simples caja de búsquedas como los que usa Google o como los inconfundibles formularios para subir comentarios siempre presentes en la mayoría de blogs, pero también existen complejas interfaces que permiten la entradas de datos interactivamente y de forma personalizada, y es que los formularios son una necesidad básica en la mayoría de aplicaciones Web modernas que necesitan recopilar datos a través de internet.

Generalmente, se necesitan dos partes para desarrollar un formulario: la interfaz de usuario en HTML y la vista que procesa los datos obtenidos o subidos por los usuarios.

```
biblioteca/views.py
from django.shortcuts import render

def formulario_buscar(request):
    return render(request, 'formulario_buscar.html')
```

La vista puede estar en cualquier lugar de la ruta de búsqueda de Python. Pero por convención este debe de ir en una vista, podemos colocarla en biblioteca/views.py.

Acompañada de una plantilla llamada formulario_buscar.html, que debe ubicarse en un directorio llamado templates (de forma predeterminada Django busca las plantillas en un directorio interno llamado template en cada aplicación registrada), dentro del directorio de la aplicación biblioteca, en el mismo nivel que el directorio migrations → → → → →

```
biblioteca/templates/formulario_buscar.html
<html>
<head>
  <title>Buscar</title>
</head>
<body>
  <form action="/buscar/" method="get">
    <input type="text" name="q">
    <input type="submit" value="Buscar">
  </form>
</body>
</html>
```

Y para acceder al formulario, necesitamos el patrón para la URL:

```
misitio/urls.py
from biblioteca import views

urlpatterns = [
    # ...
    url(r'^formulario-buscar/$', views.formulario_buscar),
]
```

GET y POST

Usa GET cuando el acto de subir un formulario solo sea para “pedir” datos. En cambio usa POST siempre que el acto de subir el formulario tenga efectos secundarios -- cambiando datos, enviando un e-mail o algo que vaya más allá de simplemente mostrar datos.

Bibliografía

Garcia M., S. (2015). *La Guía definitiva de Django*. Celayita México: Django Software Corporation.