//formularios

1.-Crear el formulario dentro de la aplicación

forms.py

2.-Crear formularios con modelos

from django import forms

from .models import Product

class ProductForm(forms.ModelForm):

class Meta:

model = Product

fields = '\_\_all\_\_'

3.-Ligar formulario a una vista

en views.py crear una funcion

from .forms import ProductForm

def new\_product(request):

template = loader.get\_template('new\_product.html')

form = ProductForm()

context = {

'form': form

}

return HttpResponse(template.render(context,request))

4.-Crear Url en urls.py de Product

urlpatterns =[

url(r'^$',views.hello\_world, name='hello'),

url(r'^product/(?P<pk>[0-9]+)/$', views.product\_detail),

url(r'^product/new',views.new\_product,name="new\_product")

]

5.- Crear template new\_product.html

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<title>New article</title>

</head>

<body>

<h1>New Article</h1>

<form method="POST" enctype="multipart/form-data">

{% csrf\_token %}

{{ form.as\_p }}

<button type="submit">Submit</button>

</form>

</body>

</html>

6.-Agregar en views que guarde datos

(traer variable HttpResponseRedirect)

from django.http import HttpResponse, HttpResponseRedirect

def new\_product(request):

if request.method == 'POST':

form = ProductForm(request.POST, request.FILES)

if form.is\_valid():

product = form.save()

product.save()

return HttpResponseRedirect('/')

else:

form = ProductForm()

template = loader.get\_template('new\_product.html')

context = {

'form': form

}

return HttpResponse(template.render(context,request))

7.-agregar en index.html

version 1.9 //<a href="{% url 'products.views.new\_product' %}">New</a>

version 1.10 //<a href="{% url 'new\_product' %}">New</a>