

PAQUETERÍA

admin.py

```
from django.contrib import admin
from entregas.models import Destinatario
```

```
admin.site.register(Destinatarior)
```

forms.py

```
from django import forms
from entregas.models import *

from django.forms import ModelForm

class NewDestForm(ModelForm):
    direccion = forms.CharField()
    ciudad = forms.CharField()
    distancia = forms.IntegerField()

    class Meta:
        model = Destinatario
        fields = ['direccion', 'ciudad', 'distancia']

    def __init__(self, *args, **kwargs):
        super(NewDestForm, self).__init__(*args, **kwargs)
        self.fields['direccion'].label = "Direccion"
        self.fields['ciudad'].label = "Ciudad"
        self.fields['distancia'].label = "Distancia"

class NewPaqueteForm(ModelForm):
    contenido = forms.CharField()
    valor = forms.IntegerField()

    class Meta:
        model = Paquete
        fields = ['contenido', 'valor', 'destinatario']
```

models.py

```
from django.db import models

# Create your models here

# Un destinatario es una persona a la que se debe entregar un paquete
class Destinatario(models.Model):
    direccion = models.CharField(max_length=100)
    ciudad = models.TextField()
    distancia = models.IntegerField()

class Paquete(models.Model):
    contenido = models.CharField(max_length=150)
    valor = models.IntegerField()
    destinatario = models.ForeignKey(Destinatarior)
```

views.py

```
from entregas.models import *
from django.shortcuts import render
from django.http import HttpResponseRedirect
from entregas.forms import NewDestForm, NewPaqueteForm
from django.contrib.auth.models import User
from django.contrib.auth.forms import UserCreationForm,
AuthenticationForm
from django.contrib.auth import login, authenticate, logout
from django.contrib.auth.decorators import login_required

# Create your views here.

def DestinatariosView(request):
    destinatarios = Destinatario.objects.all()
    return render(request, 'destinatarios.html', {'destinatarios':
destinatarios, })

@login_required(login_url = '/login')
def NewDestView(request):
    if not request.user.is_anonymous():
        return HttpResponseRedirect('/')
    if request.method=='POST':
        form = NewDestForm(request.POST)

        if form.is_valid():
            form.save()
            return HttpResponseRedirect('/')
    else:
        form = NewDestForm()
    return render(request, 'newDest.html', {'form':form, })

def DetallesDest(request, id_dest):
    dest = Destinatario.objects.get(pk=id_dest)
    return render(request, 'detallesDest.html', {'dest': dest, })

def PaquetesView(request):
    paquetes = Paquete.objects.all()
    return render(request, 'paquetes.html', {'paquetes': paquetes, })

@login_required(login_url = '/login')
def NewPaqueteView(request):
    if not request.user.is_anonymous():
        return HttpResponseRedirect('/')
    if request.method=='POST':
        form = NewPaqueteForm(request.POST)

        if form.is_valid():
            form.save()
            return HttpResponseRedirect('/')
    else:
        form = NewPaqueteForm()
    return render(request, 'newPaquete.html', {'form':form, })

def DetallesPaquete(request, id_paquete):
    paquete = Paquete.objects.get(pk=id_paquete)
    return render(request, 'detallesPaquete.html', {'paquete':
paquete, })

@login_required(login_url = '/login')
```

```

def EditarPaquete(request, id_paquete):
    paquete = Paquete.objects.get(pk=id_paquete)
    if not request.user.is_anonymous():
        return HttpResponseRedirect('/')
    if request.method == 'POST':
        form = NewPaqueteForm(request.POST, instance = paquete)

        if form.is_valid():
            form.save()
            return HttpResponseRedirect('/')
    else:
        form = NewPaqueteForm(instance = paquete)

    return render(request, 'editarPaquete.html',{'form':form, })

def login_view(request):
    if not request.user.is_anonymous():
        return HttpResponseRedirect('/')
    if request.method == 'POST':
        form = AuthenticationForm(request.POST)
        if form.is_valid():
            username = request.POST['username']
            password = request.POST['password']
            user = authenticate(username=username,
password=password)
            if user is not None:
                if user.is_active:
                    login(request, user)
                    return HttpResponseRedirect('/')
                else:
                    return render(request, 'banned.html')
            else:
                return render(request, 'not_exist.html')
    else:
        form = AuthenticationForm()
    return render(request, 'login.html', {'form': form, })

def newUser(request):
    if not request.user.is_anonymous():
        return HttpResponseRedirect('/')
    if request.method=='POST':
        form = UserCreationForm(request.POST)

        if form.is_valid():
            form.save()
            return HttpResponseRedirect('/')
    else:
        form = UserCreationForm()
    return render(request, 'register.html', {'form': form, })

@login_required(login_url = '/login')
def logout_view(request):
    logout(request)
    return HttpResponseRedirect('/')

```

urls.py

```
from django.conf.urls import patterns, include, url
from django.views.generic import TemplateView

# Uncomment the next two lines to enable the admin:
from django.contrib import admin
admin.autodiscover()

urlpatterns = patterns('',
    # Examples:
    url(r'^$', TemplateView.as_view(template_name='index.html'),
name='index'),
    # url(r'^Paqueteria/', include('Paqueteria.foo.urls')),
    url(r'^destinatarios$', 'entregas.views.DestinatariosView'),
    url(r'^destinatarios/new$', 'entregas.views.NewDestView'),
    url(r'^destinatarios/(?P<id_dest>\d+)$',
'entregas.views.DetallesDest'),
    url(r'^paquetes$', 'entregas.views.PaquetesView'),
    url(r'^paquetes/new$', 'entregas.views.NewPaqueteView'),
    url(r'^paquetes/(?P<id_paquete>\d+)$',
'entregas.views.DetallesPaquete'),
    url(r'^paquetes/(?P<id_paquete>\d+)/settings$',
'entregas.views.EditarPaquete'),

    url(r'^login$', 'entregas.views.login_view'),
    url(r'^register$', 'entregas.views.newUser'),
    url(r'^logout$', 'entregas.views.logout_view'),

    # Uncomment the admin/doc line below to enable admin
documentation:
    url(r'^admin/doc/', include('django.contrib.admindocs.urls')),

    # Uncomment the next line to enable the admin:
    url(r'^admin/', include(admin.site.urls)),
)
```

index.html

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset='utf-8'>
    <title>Empresa de reparto de paquetes</title>
</body>
<h2>Bienvenido a la agencia de transportes</h2>
<ul>
    <li><a href="/">Home</a></li>
    <li><a href="{% url 'entregas.views.DestinatariosView'
%}">Destinatarios</a></li>
    <li><a href="{% url 'entregas.views.NewDestView'
%}">Nuevo Destinatario</a></li>
    <li><a href="{% url 'entregas.views.PaquetesView'
%}">Paquetes</a></li>
    <li><a href="{% url 'entregas.views.NewPaqueteView'
%}">Nuevo Paquete</a></li>
    {% if not user.is_authenticated %}
    <li><a href="{% url 'entregas.views.login_view'
%}">Login</a></li>
    <li><a href="{% url 'entregas.views.newUser'
%}">Register</a></li>
    {% endif %}
</ul>
```

```

        {% endif %}
        {% if user.is_authenticated %}
        <li><a href="{% url 'entregas.views.logout_view'
%}">Logout</a></li>
        {% endif %}
        <li><a href="/about">About</a></li>
        <li><a href="/rules">Rules</a></li>
    </ul>
{% block content %}
{% endblock content %}
</body>
</html>

```

(sólo hay que cambiar los valores que queremos mostrar)

destinatarios.html && paquetes.html

```

{% extends 'index.html' %}
{% block content %}
    {% for dest in destinatarios %}
    <p>
        Destinatario: {{ dest.id }}<a href="/destinatarios/{{ dest.id
}}"> Detalles</a>
    </p>
    {% endfor %}
{% endblock content %}

```

detallesDest.html && detallesPaquetes.html

```

{% extends 'index.html' %}
{% block content %}
<p>
    Ciudad: {{dest.ciudad}}<br>
    Direccion: {{dest.direccion}}<br>
    Distancia: {{dest.distancia}} Km<br>
</p>
{% endblock content %}

```

newDest.html && newPaquete.html && editarPaquete.html && login.html && register.html

```

{% extends 'index.html' %}
{% block content %}
<form action="" method="post">{% csrf_token %}
    {{form.as_ul}}
    <input type="submit" value="<nombre del botón">">
</form>
{% endblock content %}

```