**ETAPA 8**

Agora, depois de ter feito todo o sistema de cadastro de doadores, devemos fazer agora o sistema de doação, para isso devemos levar em questão somente 2 coisas, o fazer doação e o listar doação, já que depois de feita, uma doação não pode ser alterada

**1 Fazendo o insert**

Antes de fazermos qualquer função devemos instanciar os valores do APP/models.py que tem a tabela doacao , devemos criar um forms que pegue quais colunas.

**1-1 Forms.py**

Vá para o file APP/forms.py , e vamos criar um novo forms que instancie o objeto doacao, que no caso o forms em questão é esse

class DoacaoForm(forms.ModelForm):

# Outros campos do formulário aqui

codigo\_doador = forms.ModelChoiceField(queryset=Doador.objects.all(), empty\_label=None)

class Meta:

model = Doacao

fields = ('Data', 'Hora', 'volume', 'situacao', 'codigo\_doador')

É por ele que vamos mandar os valores para dentro das tabelas, sem ele a view não será capaz de instanciar os valores

**1-2 Fazendo uma view**

De inicio devemos criar uma nova view, para isso vamos criar uma nova view dentro de APP/views.py de nome insertdoacao que seria esse :

def insertdoacao(request):

doadores = Doador.objects.all()

# Crie uma lista de tuplas (código, nome) para usar no campo de escolha

lista\_doadores = [(doador.codigo, doador.nome) for doador in doadores]

order\_forms = Doacao()

if request.method == 'POST':

forms = DoacaoForm(request.POST, request.FILES,

instance=order\_forms, prefix='main')

#codigo = int(request.POST.get("id\_main-codigo\_doador"))

#doador = Doador.objects.get(pk=codigo)

# Modifique o campo tipo\_rh\_corretos

#doador.tipo\_rh\_corretos = True

# doador.save()

if forms.is\_valid() :

teste\_instance = forms.save()

# return HttpResponseRedirect(resolve\_url('detalhe\_doador',teste\_instance.pk))

else:

forms = DoacaoForm(instance=order\_forms, prefix='main')

context = {

'forms': forms,

'lista\_doadores':lista\_doadores,

}

return render(request,'insere\_doacao.html',context)

**1-3 - HTMl**

Agora vamos criar uma html que vai ler os valores e inserir eles em doação, para isso vá para a pasta APP/templates, e crie o html insere\_doacao.html, e por ele vamos inserir o codigo em questão

{% extends 'base.html' %}

{% load bootstrap %}

{# Adding select2 css #}

{% block css %}

<link href="https://cdnjs.cloudflare.com/ajax/libs/select2/4.0.3/css/select2.min.css" rel="stylesheet"/>

{% endblock %}

{# Adding select2 js #}

{% block js %}

<script src="https://cdnjs.cloudflare.com/ajax/libs/select2/4.0.3/js/select2.min.js"></script>

{% endblock %}

{% block content %}

<div class="row">

<div class="col-xs-10 col-md-12">

<form action="" method="POST">

{% csrf\_token %}

<div class="row">

<div class="col-xs-8 col-md-5">

{{ forms|bootstrap }}

<select name="codigo\_doador">

{% for codigo, nome in lista\_doadores %}

<option value="{{ codigo }}">{{ nome }}</option>

{% endfor %}

</select>

</div>

</div>

<div class="form-inline buttons">

<i class="fa fa-times"></i>

Cancelar

</a>

<button class="btn btn-primary pull-right" value="Save">

<i class="fa fa-floppy-o"></i>

Salvar

</button>

</div>

</form>

</div>

</div>

{% endblock %}

**2-List**

Agora vamos ao list  **,** para isso vamos precisar de duas novas bibliotecas, pois precisamos buscar por data , uma ja é natural do python, e outra precisa ser instalada , e para instalar uma delas , rode o comando :

pip install python-dateutil

pelo terminal da própria ide que é o visual studio, e depois disso, vá para APP/views.py e adcione esse seguinte imports:

from django.shortcuts import render, resolve\_url, get\_object\_or\_404, redirect

from django.http import HttpResponseRedirect

from django.contrib.auth.decorators import login\_required

from django.db.models import Count

from django.views.generic import TemplateView, ListView, DetailView

from django.forms.models import inlineformset\_factory

from dateutil.parser import parse

from datetime import timedelta

from django.core.paginator import Paginator

from django.contrib import messages

from datetime import datetime

from ..models import Venda, ItemVEN, Cadastro, Produto, CondPag,Setor,SetorPRO

from ..forms import VendaForm, ItemVenForm

from ..mixins import CounterMixin

Coloque eles no inicio do codigo, antes de qualquer view , agora depois disso vamos para a view e criar seu list

**2-1 List view**

Para criar ele, mantenha-se dentro da APP/views.py e crie o codigo que vai instanciar todas as doaçoes, e ainda pegar dos campos as pesquisas do codigo, e essa é a view:

def listar\_doacoes(request):

object\_list = Doacao.objects.all().order\_by('-codigo')

# filtro na lista por datas

data\_inicial = request.GET.get('data\_inicial')

data\_final = request.GET.get('data\_final')

print(data\_inicial)

print(data\_final)

if data\_inicial and data\_final:

data\_final = parse(data\_final) + timedelta(1)

object\_list = object\_list.filter(Data\_\_range=[data\_inicial, data\_final])

# pesquisa na lista de vendas pelo nome do cliente

search = request.GET.get('search\_box')

if search:

object\_list = object\_list.filter(codigo\_doador\_\_nome\_\_icontains=search)

paginator = Paginator(object\_list, 20)

page = request.GET.get('page', 1)

try:

vendas = paginator.page(page)

except PageNotAnInteger:

vendas = paginator.page(1)

except EmptyPage:

vendas = paginator.page(paginator.num\_pages)

context = {'object\_list':object\_list}

return render(request,'lista\_doacoes.html',context)

**2-2 Html**

Agora vamos a parte do html, e vamos criar uma html para ler todo a view em questão, vá para o APP/TEMPLATES novamente, e crie o html de nome listar\_doacoes.html, e insira esse html em questão :

{% extends 'base.html' %}

{% block title %}Lista de doacoes{% endblock title %}

{% block subtitle %}Lista de doacoes{% endblock subtitle %}

{% block content %}

<form class="navbar-form navbar" action="#" method="get">

<div class="pull-right">

<span>

<button type="button" action="#" id="btnFiltro" class="btn btn-info" title="Abrir menu de filtros"><span class="glyphicon glyphicon-filter"></span> Filtrar</button>

</span>

<span>

<!-- add -->

<a href="{% url 'insertdoacao' %}">

<button id="new\_sale" type="button" class="btn btn-primary" title="Criar uma nova venda"><span class="glyphicon glyphicon-plus"></span> Adicionar</button>

</a>

</span>

</div>

<div id="filtros" class="pull-left hidden">

<!-- filtro por datas -->

<label for="data\_inicial">Data Inicial: </label>

<input id="data\_inicial" type="date" name="data\_inicial" class="form-control ml-sm-2">

<label for="data\_final">Data Final: </label>

<input type="date" id ="data\_final" name="data\_final" class="form-control ml-sm-2">

<button id="btnFiltrarDatas" class="btn btn-success" type="submit" title="Filtrar vendas por intervalo de datas"><i class="fa fa-search"></i></button>

<!-- search form -->

<input id="search\_box" name="search\_box" type="text" placeholder="Localizar..." class="form-control" title="Filtrar as vendas realizadas pelo id do setor">

<button id="btnFiltrarNome" type="submit" class="btn btn-success form-control" title="Filtrar vendas pelo nome do cliente"><i class="fa fa-search"></i></button>

</div>

</form>

<div>

{% if object\_list %}

<div>

<table class="table table-striped">

<thead>

<tr>

<th class="text-center">codigo</th>

<th class="text-center">Data</th>

<th class="text-center">Hora</th>

<th class="text-center">volume</th>

<th class="text-center">situacao</th>

<th class="text-center">codigo\_doador</th>

</tr>

</thead>

<tbody>

{% for doacao in object\_list %}

<td class="text-center" ><a>{{doacao.codigo }}</td>

<td class="text-center">{{doacao.Data}}</td>

<td class="text-center">{{doacao.Hora}}</td>

<td class="text-center">{{doacao.volume}}</td>

<td class="text-center">{{doacao.situacao}}</td>

<td class="text-center">{{doacao.codigo\_doador.nome}}</td>

</tr>

{% endfor %}

</tbody>

</table>

</div>

{% else %}

<p class="alert alert-warning">Sem itens na lista.</p>

{% endif %}

<hr>

<div>

<h3>{{ object\_list|length }} doacao {{ doacoes|length|pluralize }}</h3>

<h4><b>Total:</b> {{ doacao.paginator.count }} doacao{{ doacao.paginator.count|pluralize }}</h4>

</div>

{% if object\_list.has\_other\_pages %}

<div class="row text-center">

<div class="col-lg-12">

<ul class="pagination">

{% if object\_list.has\_previous %}

<li><a href="?page={{ object\_list.previous\_page\_number }}">&laquo;</a></li>

<li><a href="?page=1">1</a></li>

{% if object\_list.previous\_page\_number != 1 %}

<li><a href="?page={{ object\_list.previous\_page\_number }}">{{ vendas.previous\_page\_number }}</a></li>

{% endif %}

{% else %}

<li class="disabled"><span>&laquo;</span></li>

{% endif %}

<li class="active"><span>{{ vendas.number }} <span class="sr-only">(current)</span></span></li>

{% if object\_list.has\_next %}

{% if object\_list.next\_page\_number != vendas.paginator.num\_pages %}

<li><a href="?page={{ object\_list.next\_page\_number }}">{{ vendas.next\_page\_number }}</a></li>

{% endif %}

<li><a href="?page={{ object\_list.paginator.num\_pages }}">{{ vendas.paginator.num\_pages }}</a></li>

<li><a href="?page={{ object\_list.next\_page\_number }}">&raquo;</a></li>

{% endif %}

</ul>

</div>

</div>

{% endif %}

<script type="text/javascript">

const btnFiltros = document.querySelector("#btnFiltro");

btnFiltros.addEventListener('click', function(evento) {

evento.preventDefault();

const seletorFiltros = document.querySelector("#filtros");

if(seletorFiltros.classList.contains("hidden")) {

seletorFiltros.classList.remove('hidden');

} else {

seletorFiltros.classList.add('hidden');

}

});

</script>

{% endblock content %}

Esse código vai listar as doacoes e por ele ainda vai numerar paginas e filtrar elas pela data e pelo nome .

**3- URLS**

Agora como já e mencionado , vamos criar urls para todas nossas views que criamos, para isso, vá para o file APP/URLs.py ,e insira duas novas linhas que são elas :

path('insere\_doacao/',view=insertdoacao,name='insertdoacao'),

path('Lista\_Doacao/',listar\_doacoes,name='Doacao\_list'),

ou para facilitar, copie e cole esse codigo :

from django.urls import include, path

from django.contrib.auth.decorators import login\_required

from django.conf import settings

from django.conf.urls.static import static

#from django.conf.urls.static import media

from rest\_framework import routers

from .views import \*

urlpatterns = [

path('',home,name="home"),

path('conc2/',inserttest,name='formu'),

path('conc/',teste2,name='teste2'),

path('mar6/',mar6,name='mar6'),

path('men6/',men6,name='men6'),

path('Lista\_Doador/',Doador\_list,name='Doador\_list'),

path('Lista\_Doacao/',listar\_doacoes,name='Doacao\_list'),

path('formulario/',formulario,name='formulario'),

path('adcionar\_doador/',insertdoador,name='insertdoador'),

path('<int:pk>/',view=detalhe\_formulario,name='detalhe\_formulario'),

path('Doador/<int:pk>/',view=detalhe\_doador,name='detalhe\_doador'),

path('deleta\_Doador/<int:pk>/',view=deleta\_doador,name='deleta\_doador'),

path('altera\_doador/<int:pk>/',view=altera\_doador,name='altera\_doador'),

path('insere\_doacao/',view=insertdoacao,name='insertdoacao'),

]

obs: todas as linhas que estão dentro de uma função ou qualquer outra coisa disso, devem ter o mesmo numero de espaço, em outras palavras devem estar alinhadas

**4- De volta ao Menu**

Agora voltando ao menu, vamos colocar todas as ulrs em funcionamento? , para isso, voltamos para a APP/templates/index.html, e substitua todo o html por esse para facilitar :

{% load static %}

{% load bootstrap %}

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="utf-8">

<meta http-equiv="X-UA-Compatible" content="IE=edge">

<meta name="viewport" content="width=device-width, initial-scale=1">

<meta name="description" content="">

<meta name="author" content="">

<link rel="shortcut icon" href="{% static 'img/cart.png' %}">

{% block title %}

<title>sistema doacao</title>

{% endblock title %}

<!-- Bootstrap Core CSS -->

<link href="{% static 'css/bootstrap.min.css' %}" rel="stylesheet">

<!-- Custom CSS -->

<link href="{% static 'css/rgscale.css' %}" rel="stylesheet">

<!-- Custom Fonts -->

<link href="{% static 'font-awesome/css/font-awesome.min.css' %}" rel="stylesheet" type="text/css">

<link href="http://fonts.googleapis.com/css?family=Lora:400,700,400italic,700italic" rel="stylesheet" type="text/css">

<link href="http://fonts.googleapis.com/css?family=Montserrat:400,700" rel="stylesheet" type="text/css">

</head>

<body id="page-top" data-spy="scroll" data-target=".navbar-fixed-top">

<!-- MENU -->

{% include "menu\_landpage.html" %}

<!-- Intro Header -->

<header class="intro">

<div class="intro-body">

<div class="container" id="icones-container">

<div class="row">

<div class="col-md-8 col-md-offset-2">

<a href="{% url 'Doador\_list' %}" class="btn btn-circle" id="cadastro" title="Cadastro"><i class="glyphicon glyphicon-user"></i></a>

<a href="{% url 'Doacao\_list' %}" class="btn btn-circle" id="daocao" title="Doacao"><i class="glyphicon glyphicon-tint"></i></a>

</div>

</div>

<img src="{% static './img/Entrada SIG-2000.jpg' %}" class="pull-left" id="img-entrada">

</div>

</div>

</header>

<!-- Footer -->

<footer >

<div class="container text-center">

</div>

</footer>

<!-- jQuery -->

<script src="{% static 'js/jquery.js' %}"></script>

<!-- Bootstrap Core JavaScript -->

<script src="{% static 'js/bootstrap.min.js' %}"></script>

<!-- Plugin JavaScript -->

<script src="{% static 'js/jquery.easing.min.js' %}"></script>

<!-- Google Maps API Key - Use your own API key to enable the map feature. More information on the Google Maps API can be found at https://developers.google.com/maps/ -->

<script type="text/javascript" src="https://maps.googleapis.com/maps/api/js?key=AIzaSyCRngKslUGJTlibkQ3FkfTxj3Xss1UlZDA&sensor=false"></script>

<!-- Custom Theme JavaScript -->

<script src="{% static 'js/grayscale.js' %}"></script>

</body>

</html>