

**NÚCLEO DE EDUCAÇÃO A DISTÂNCIA (NEAD)**

***TRABALHO DE PROGRAMAÇÃO EM***

***SISTEMAS DE INFORMAÇÃO (AP3)***

**APLICAÇÃO PRÁTICA**

|  |
| --- |
| **Turma: EIN 420-60** |
| **Professor: Oswaldo Borges Peres** |
| **Semestre: 2021/01** |
| **Aluno(a): 5405866 - Alexsandro Marciel**  **5405885 - Diego Carlos Menezes de Oliveira**  **0700830 - Márcio Lima** |

**Proposta de Trabalho**

Suponha que você seja um desenvolvedor de sistemas em Python e foi contratado,

por uma concessionária de porte médio, para implementar o sistema descrito abaixo:

• efetuar o CRUD de Clientes

• efetuar o CRUD de Motocicletas

• efetuar a venda de Motocicletas

• fornecer listagem de Vendas

• permitir a consulta de uma venda específica, incluindo o cliente e motocicleta

relacionada a mesma

**2ª ENTREGA:**

Link do vídeo postado no youTube: <https://youtu.be/zhx16ZkFNnI>

**Tela Inicial: base.html**

****

**Código:**

<!DOCTYPE html>

<html lang="en">

<head>

  <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/materialize/0.98.2/css/materialize.min.css">

      <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">

    <meta charset="UTF-8">

    <title>

        {% block title %}

        {% endblock %}

    </title>

</head>

<body>

  <nav>

    <div class="nav-wrapper">

      <a href="#" class="brand-logo center">Venda de Motos</a>

      <ul id="nav-mobile" class="left hide-on-med-and-down">

      </ul>

    </div>

  </nav>

<div class="container">

{% block content %}

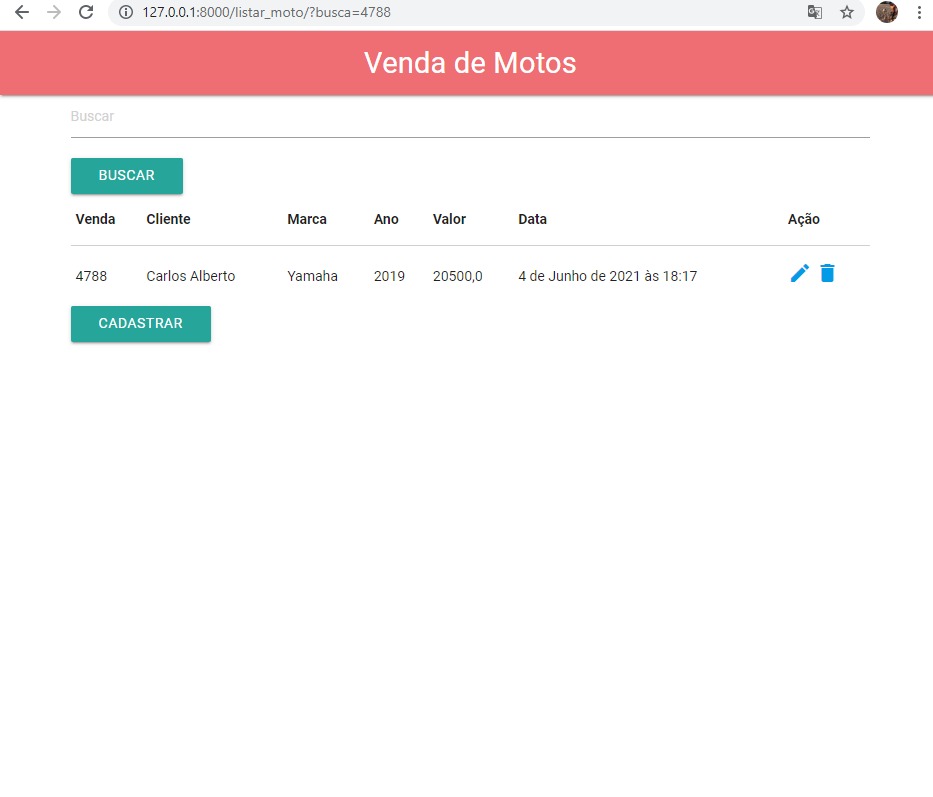
{% endblock %}

</div>

</body>

</html>

**Cadastro de Motos: form.html**

****

**Código:**

{% extends 'base.html' %}

{% block title %}

Cadastro de Motos

{% endblock title %}

{% block content %}

<form method="post">

        {% csrf\_token %}

    {{ form.as\_p }}

    <button type="submit" class="btn btn-primary">

        Salvar

    </button>

</form>

{% endblock content %}

**Lista de Motos: moto\_list.html**

****

**Código:**

{% extends 'base.html' %}

{% block title %}

Listagem de Motos

{% endblock title %}

{% block content %}

<form method="get">

    <input type="text" name="busca" placeholder="Buscar">

    <button type="submit" class="btn btn-primary">

      Buscar

    </button>

</form>

<table class="table">

  <thead>

    <tr>

      <th>Venda</th>

      <th>Cliente</th>

      <th>Marca</th>

      <th>Ano</th>

      <th>Valor</th>

      <th>Data</th>

      <th>Ação</th>

    </tr>

  </thead>

  <tbody>

  {% for moto in lista %}

    <tr>

      <td>{{ moto.venda }}</td>

      <td>{{ moto.cliente }}</td>

      <td>{{ moto.marca }}</td>

      <td>{{ moto.ano }}</td>

      <td>{{ moto.valor}}</td>

      <td>{{ moto.data\_cadastro}}</td>

      <td>

        <a href="{% url 'editar\_moto' pk=moto.pk %}">

      <i class="material-icons">mode\_edit</i>

        </a>

        <a href="{% url 'remover\_moto' pk=moto.pk %}">

      <i class="material-icons">delete</i>

        </a>

      </td>

    </tr>

  {% endfor %}

  </tbody>

</table>

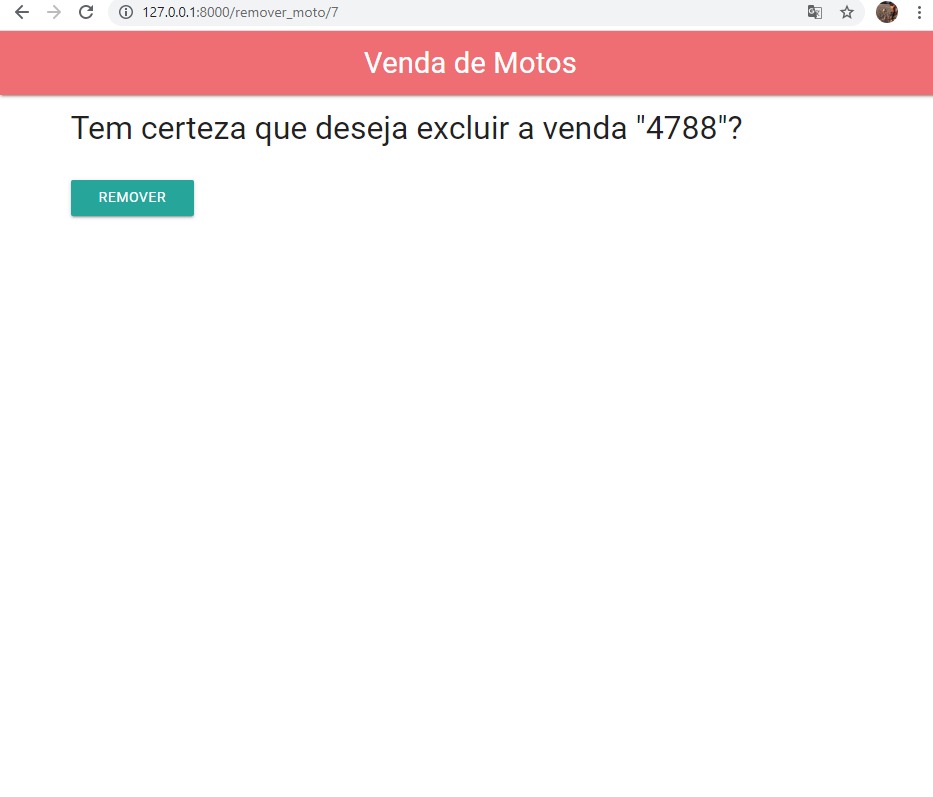
<a href="{% url 'cadastrar\_moto' %}" class="btn btn-primary">

    Cadastrar

</a>

{% endblock content %}

**Excluir: moto\_delete.html**

****

**Código:**

{% extends 'base.html' %}

{% block title %}

Remoção de Motos

{% endblock title %}

{% block content %}

<form method="post">{% csrf\_token %}

    <h4>Tem certeza que deseja excluir a venda "{{ moto.venda }}"?</h4>

    <br/>

<button type="submit" class="btn btn-primary">

    Remover

</button>

</form>

{% endblock content %}

**Códigos Python:**

**0001\_initial.py:**

# Generated by Django 3.2.4 on 2021-06-04 17:02

import datetime

import django.core.validators

from django.db import migrations, models

class Migration(migrations.Migration):

    initial = True

    dependencies = [

    ]

    operations = [

        migrations.CreateModel(

            name='Moto',

            fields=[

                ('id', models.BigAutoField(auto\_created=True, primary\_key=True, serialize=False, verbose\_name='ID')),

                ('venda', models.CharField(max\_length=50)),

                ('modelo', models.CharField(max\_length=50)),

                ('marca', models.CharField(max\_length=50)),

                ('ano', models.PositiveIntegerField(validators=[django.core.validators.MinValueValidator(2005)])),

                ('valor', models.FloatField()),

                ('data\_cadastro', models.DateTimeField(default=datetime.datetime(2021, 6, 4, 14, 2, 49, 417042))),

            ],

        ),

    ]

**admin.py:**

from django.contrib import admin

# Register your models here.

**apps.py:**

from django.apps import AppConfig

class AppConfig(AppConfig):

    default\_auto\_field = 'django.db.models.BigAutoField'

    name = 'app'

**models.py:**

from django.core.validators import MinValueValidator

from django.db import models

# Create your models here.

from django.utils.datetime\_safe import datetime

class Moto(models.Model):

    venda = models.CharField(max\_length=50, null=False)

    cliente = models.CharField(max\_length=50, null=False)

    marca = models.CharField(max\_length=50, null=False)

    ano = models.PositiveIntegerField(validators=[MinValueValidator(2005)], null=False)

    valor = models.FloatField(null=False)

    data\_cadastro = models.DateTimeField(default=datetime.now(), null=False)

**tests.py:**

from django.test import TestCase

# Create your tests here.

**Urls.py:**

from django.conf.urls import url

from .views import \*

urlpatterns = [

    url(r'^cadastrar\_moto/', cadastrar\_moto, name='cadastrar\_moto'),

    url(r'^listar\_moto/', listar\_moto, name='moto\_list'),

    url(r'^editar\_moto/(?P<pk>[0-9]+)', editar\_moto, name='editar\_moto'),

    url(r'^remover\_moto/(?P<pk>[0-9]+)', remover\_moto, name='remover\_moto'),

]

**views.py:**

from django.forms import ModelForm

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

# Create your views here.

from .models import \*

class MotoForm(ModelForm):

    class Meta:

        model = Moto

        fields = ['venda', 'cliente', 'marca', 'ano', 'valor', 'data\_cadastro']

def cadastrar\_moto(request, template\_name='moto\_form.html'):

    form = MotoForm(request.POST or None)

    if form.is\_valid():

        form.save()

        return redirect('moto\_list')

    return render(request, template\_name, {'form': form})

def listar\_moto(request, template\_name="moto\_list.html"):

    query = request.GET.get("busca")

    if query:

        moto = Moto.objects.filter(venda\_\_icontains=query)

    else:

        moto = Moto.objects.all()

    motos = {'lista': moto}

    return render(request, template\_name, motos)

def editar\_moto(request, pk, template\_name='moto\_form.html'):

    moto = get\_object\_or\_404(Moto, pk=pk)

    if request.method == "POST":

        form = MotoForm(request.POST, instance=moto)

        if form.is\_valid():

            form.save()

            return redirect('moto\_list')

    else:

        form = MotoForm(instance=moto)

    return render(request, template\_name, {'form': form})

def remover\_moto(request, pk, template\_name='moto\_delete.html'):

    moto = Moto.objects.get(pk = pk)

    if request.method == "POST":

        moto.delete()

        return redirect('moto\_list')

    return render(request, template\_name, {'moto': moto})