Criando um projeto padrão no Django

1. Pip install

pip install django whitenoise gunicorn psycopg2-binary dj-static django-stdimage django-bootstrap-v5

[Quickstart — django-bootstrap-v5 1.0.0 documentation](https://django-bootstrap-v5.readthedocs.io/en/latest/quickstart.html)

pip freeze > requirements.txt

1. Criando projeto e aplicativo

django-admin startproject djangoBase .

Django-admin startapp core

1. Configurando settings.py

ALLOWED\_HOSTS = ['\*']

INSTALLED\_APPS = [ 'django.contrib.admin', 'django.contrib.auth', 'django.contrib.contenttypes', 'django.contrib.sessions', 'django.contrib.messages', 'django.contrib.staticfiles', 'core',]

TEMPLATES = [ { 'BACKEND': 'django.template.backends.django.DjangoTemplates', 'DIRS': ['templates'], 'APP\_DIRS': True, 'OPTIONS': { 'context\_processors': [ 'django.template.context\_processors.debug', 'django.template.context\_processors.request', 'django.contrib.auth.context\_processors.auth', 'django.contrib.messages.context\_processors.messages', ], }, },]

LANGUAGE\_CODE = 'pt-br'TIME\_ZONE = 'America/Araguaina'

MEDIA\_URL = 'media/'STATIC\_ROOT = os.path.join(BASE\_DIR,'staticfiles')MEDIA\_ROOT = os.path.join(BASE\_DIR,'media')

1. Criando os diretórios

Dentro do app ‘core’ crie uma pasta chamada ‘static’ e outra chamada ‘templates’

1. Arquivos html

Crie um arquivo index.html dentro da pasta ‘templates’

1. Configurando o arquivo views.py

from django.views.generic import TemplateView  
  
class IndexView(TemplateView):  
 template\_name = 'index.html'

1. Configurando os arquivos urls.py

dentro do arquivo urls.py da raiz do projeto, digite:

from django.contrib import admin  
from django.urls import path, include  
  
from django.conf.urls.static import static  
from django.conf import settings  
  
urlpatterns = [  
 path('admin/', admin.site.urls),  
 path('', include('core.urls')),  
] + static(settings.MEDIA\_URL, document\_root=settings.MEDIA\_ROOT)

Agora, dentro do app ‘core’ crie um arquivo chamado urls.py e digite:

from django.urls import path  
  
from .views import IndexView  
  
urlpatterns = [  
 path('',IndexView.as\_view(),name = 'index'),  
]

1. Estruturas Básicas dos HTMLS

Exemplos

{% load static %}   
<!DOCTYPE html>  
<html lang="en">  
 <head>

…

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

<a href="{% url 'index' %}" class="navbar-brand"><img src="{% static 'img/logo.png' %}" alt=""></a>

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

Inserindo dados de outro arquivo html

{% extends 'base.html' %}  
{% load static %}  
  
{% block content %}  
<!-- Hero Area Start -->  
 <!-- Hero Area End -->  
  
  
 <!-- Header Area wrapper End -->  
 {% include 'hero.html' %}  
 <!-- Services Section Start -->  
 {% include 'servicos.html' %}  
 <!-- Services Section End -->  
  
 <!-- About Section start -->  
  
 {% include 'about.html' %}  
  
 <!-- About Section End -->  
  
 <!-- Features Section Start -->  
  
 {% include 'features.html' %}  
  
 <!-- Features Section End -->  
  
 <!-- Team Section Start -->  
  
 {% include 'team.html' %}  
  
 <!-- Team Section End -->  
  
 <!-- Pricing section Start -->  
  
 {% include 'pricing.html' %}  
  
 <!-- Pricing Table Section End -->  
  
 <!-- Testimonial Section Start -->  
  
 {% include 'testimonial.html' %}  
  
 <!-- Testimonial Section End -->  
  
 <!-- Call To Action Section Start -->  
  
 {% include 'call.html' %}  
  
 <!-- Call To Action Section Start -->  
  
 <!-- Contact Section Start -->  
  
 {% include 'contact.html' %}  
  
 <!-- Contact Section End -->  
  
{% endblock %}

1. Definindo os Models

from django.db import modelsfrom stdimage.models import StdImageFieldimport uuiddef get\_file\_path(\_instance, filename): ext = filename.split('.') filename = f'{uuid.uuid4()}.{ext}' return filenameclass Base(models.Model): criados = models.DateField('Criação', auto\_now\_add=True) modificado = models.DateField('Atualizaçao', auto\_now=True) ativo = models.BooleanField('Ativo', default=True) class Meta: abstract=Trueclass Servico(Base): ICONE\_CHOICES = ( ('lni-cog', 'Engrenagem'), ('lni-stats-up', 'Gráfico'), ('lni-users', 'Usuários'), ('lni-layers', 'Design'), ('lni-mobile', 'Mobile'), ('lni-rocket', 'Foguete'), ) servico = models.CharField('Serviço', max\_length=100) descricao = models.TextField('Descrição', max\_length=200) icone = models.CharField('Icone', max\_length=12, choices=ICONE\_CHOICES) class Meta: verbose\_name = 'Serviço' verbose\_name\_plural = 'Serviços' def \_\_str\_\_(self): return self.servicoclass Cargo(Base): cargo = models.CharField('Cargo', max\_length=100) class Meta: verbose\_name = 'Cargo' verbose\_name\_plural = 'Cargos' def \_\_str\_\_(self): return self.cargoclass Equipe(Base): nome = models.CharField('Nome', max\_length=100) cargo = models.ForeignKey('core.Cargo', verbose\_name='Cargo', on\_delete=models.CASCADE) bio = models.TextField('Bio', max\_length=200) imagem = StdImageField('Imagem', upload\_to=get\_file\_path, variations = {'thumb': {'width': 480, 'height': 480, 'crop': True}}) facebook = models.CharField('Facebook', max\_length=100, default = '#') twitter = models.CharField('Twitter', max\_length=100, default='#') instagram = models.CharField('Instagram', max\_length=100, default='#') class Meta: verbose\_name = 'Equipe' verbose\_name\_plural = 'Equipes' def \_\_int\_\_(self): return self.nome

1. Configurando o Admin

from django.contrib import admin  
  
from .models import Cargo, Servico, Equipe  
  
@admin.register(Cargo)  
class CargoAdmin(admin.ModelAdmin):  
 list\_display = ('cargo', 'ativo', 'modificado')  
  
@admin.register(Servico)  
class ServicoAdmin(admin.ModelAdmin):  
 list\_display = ('servico', 'icone', 'ativo', 'modificado')  
  
@admin.register(Equipe):  
class EquipeAdmin(admin.ModelAdmin):  
 list\_display = ('nome', 'cargo', 'ativo','modificado')

1. Carregando dados do Banco de Dados no HTML

Ajustando views.py para passar os dados de contexto

from django.views.generic import TemplateView  
  
from .models import Servico, Equipe  
  
class IndexView(TemplateView):  
 template\_name = 'index.html'  
  
 def get\_context\_data(self, \*\*kwargs):  
 context = super(IndexView, self).get\_context\_data(\*\*kwargs)  
 context['servicos'] = Servico.objects.order\_by('?').all()  
 context['equipes'] = Equipe.objects.all()  
  
 return context

Arquivo servicos.html

{% load static %}  
  
<section id="services" class="section-padding">  
 <div class="container">  
 <div class="section-header text-center">  
 <h2 class="section-title wow fadeInDown" data-wow-delay="0.3s">Nossos Serviços</h2>  
 <div class="shape wow fadeInDown" data-wow-delay="0.3s"></div>  
 </div>  
 <div class="row">  
  
 {% for s in servicos %}  
 <!-- Services item -->  
 <div class="col-md-6 col-lg-4 col-xs-12">  
 <div class="services-item wow fadeInRight" data-wow-delay="0.3s">  
 <div class="icon">  
 <i class="{{ s.icone }}"></i>  
 </div>  
 <div class="services-content">  
 <h3><a href="#">{{ s.servico }}</a></h3>  
 <p>{{ s.descricao }}</p>  
 </div>  
 </div>  
 </div>  
 {% endfor %}  
  
  
 </div>  
 </div>  
 </section>

Arquivo team.html

{% load static %}  
  
 <section id="team" class="section-padding bg-gray">  
 <div class="container">  
 <div class="section-header text-center">  
 <h2 class="section-title wow fadeInDown" data-wow-delay="0.3s">Conheça nossa equipe</h2>  
 <div class="shape wow fadeInDown" data-wow-delay="0.3s"></div>  
 </div>  
 <div class="row">  
 {% for f in equipes %}  
  
 <div class="col-lg-6 col-md-12 col-xs-12">  
  
 <!-- Team Item Starts -->  
 <div class="team-item wow fadeInRight" data-wow-delay="0.2s">  
 <div class="team-img">  
 <img class="img-fluid" src="{{ f.imagem.thumb.url }}" alt="{{ f.nome }}">  
 </div>  
 <div class="contetn">  
 <div class="info-text">  
 <h3><a href="#">{{ f.nome }}</a></h3>  
 <p>{{ f.cargo }}</p>  
 </div>  
 <p>{{ f.bio }}</p>  
 <ul class="social-icons">  
 <li><a href="{{ f.facebook }}"><i class="lni-facebook-filled" aria-hidden="true"></i></a></li>  
 <li><a href="{{ f.twitter }}"><i class="lni-twitter-filled" aria-hidden="true"></i></a></li>  
 <li><a href="{{ f.instagram }}"><i class="lni-instagram-filled" aria-hidden="true"></i></a></li>  
 </ul>  
 </div>  
 </div>  
 <!-- Team Item Ends -->  
 </div>  
 {% endfor %}  
  
 </div>  
 </div>  
 </section>

<https://django-portuguese.readthedocs.io/en/1.0/ref/templates/builtins.html>

1. Passando valores iniciais ao formulário

Interface gráfica do usuário, Texto, Aplicativo

Descrição gerada automaticamente

Outra forma ->

Interface gráfica do usuário, Texto

Descrição gerada automaticamente

1. Deploy Heroku

[Deploy de uma aplicação Django no Heroku | Blog TreinaWeb](https://www.treinaweb.com.br/blog/deploy-de-uma-aplicacao-django-no-heroku)