

# MINICURSO DE INTRODUÇÃO À PROGRAMAÇÃO WEB COM DJANGO

APRENDA A CRIAR  
APLICAÇÕES WEB DO ZERO!



Vinicius Melchior



Gabriel Iuji



XVII SEMANA NACIONAL DE  
CIÊNCIA E TECNOLOGIA  
DE ITACOATIARA - 2024

BIOMAS DO BRASIL: DIVERSIDADE, SABERES E TECNOLOGIAS SOCIAIS.

# Roteiro

## 01 Fundamentos da Web

Introdução aos conceitos básicos

## 02 Introdução ao Django

Conceitos relacionados

## 03 Idealização do projeto

O que será feito

## 04 Início de um projeto web com Django

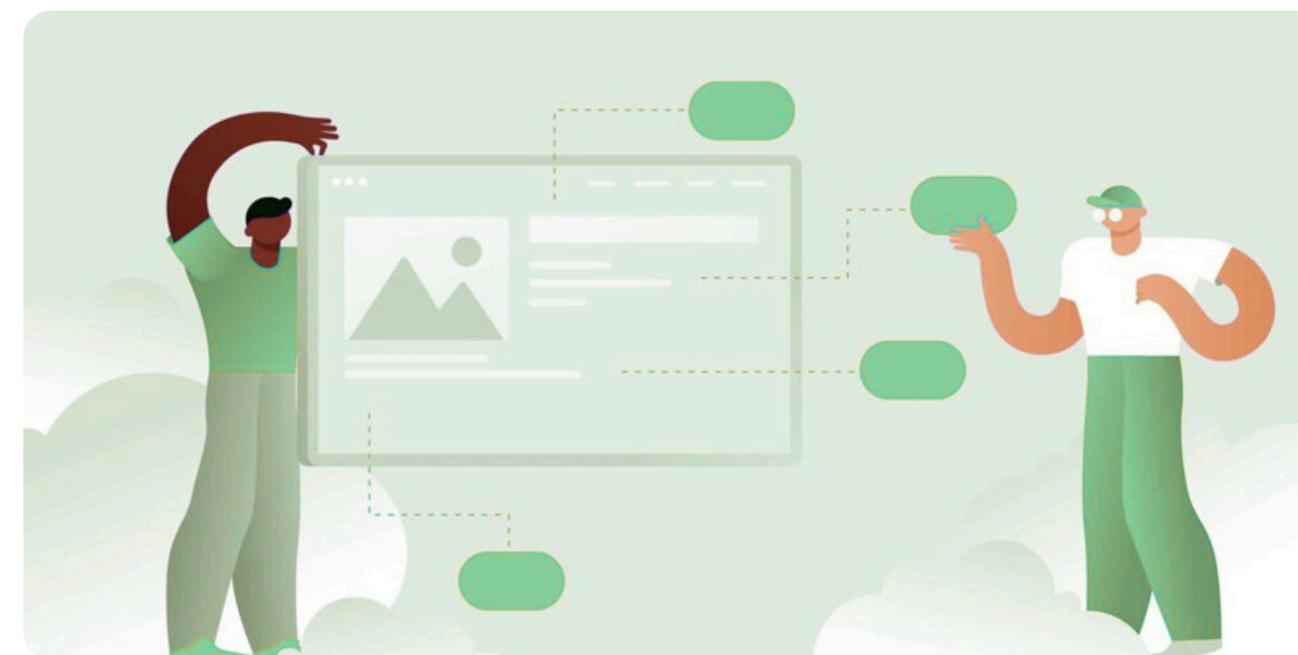
Instalação e a criação de um projeto

1.

# Fundamentos da Web

# O que é Programação Web?

A programação web é o conjunto de técnicas, linguagens, frameworks e ferramentas utilizados para criar e manter websites e aplicações acessíveis através da internet. Ela é um dos pilares do funcionamento da web moderna, possibilitando desde páginas simples e estáticas até plataformas complexas e dinâmicas





# Sites Web

X

# Aplicações Web

# Como a Web Funciona?

Requisição e Resposta



Cliente



Servidor



Protocolo

# 2.

## Introdução ao Django

Quanto maiores se tornam as aplicações web,  
mais os desenvolvedores devem lidar com sua  
crescente complexidade

- Loudon, Kyle (2018)

Frameworks surgiram para abstraírem a maioria  
dos processos do ambiente de  
desenvolvimento de aplicações

- Urano, Laís (2023)

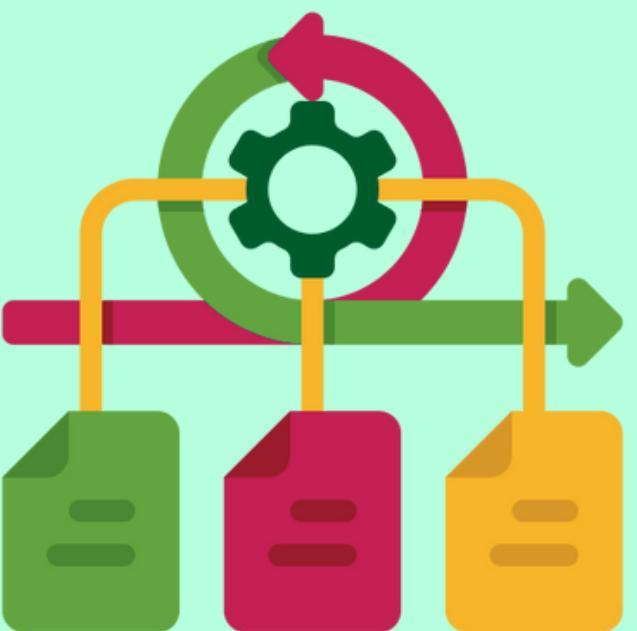
```
reProducts

    segment>
      className="py-5">
        div className="container">
          <Title name="our" title="product">
            <div className="row">
              <ProductConsumer>
                {(value) => {
                  |   |   |   console.log(value)
                }}
              </ProductConsumer>
            </div>
          </div>
        </div>
      <Segment>
```

Linguagem de programação de alto nível e código aberto amplamente utilizada em várias áreas como Aplicações Web, Ciência de Dados e Machine Learning.



Frameworks são ferramentas que fornecem estruturas prontas para facilitar e agilizar o desenvolvimento de sistemas

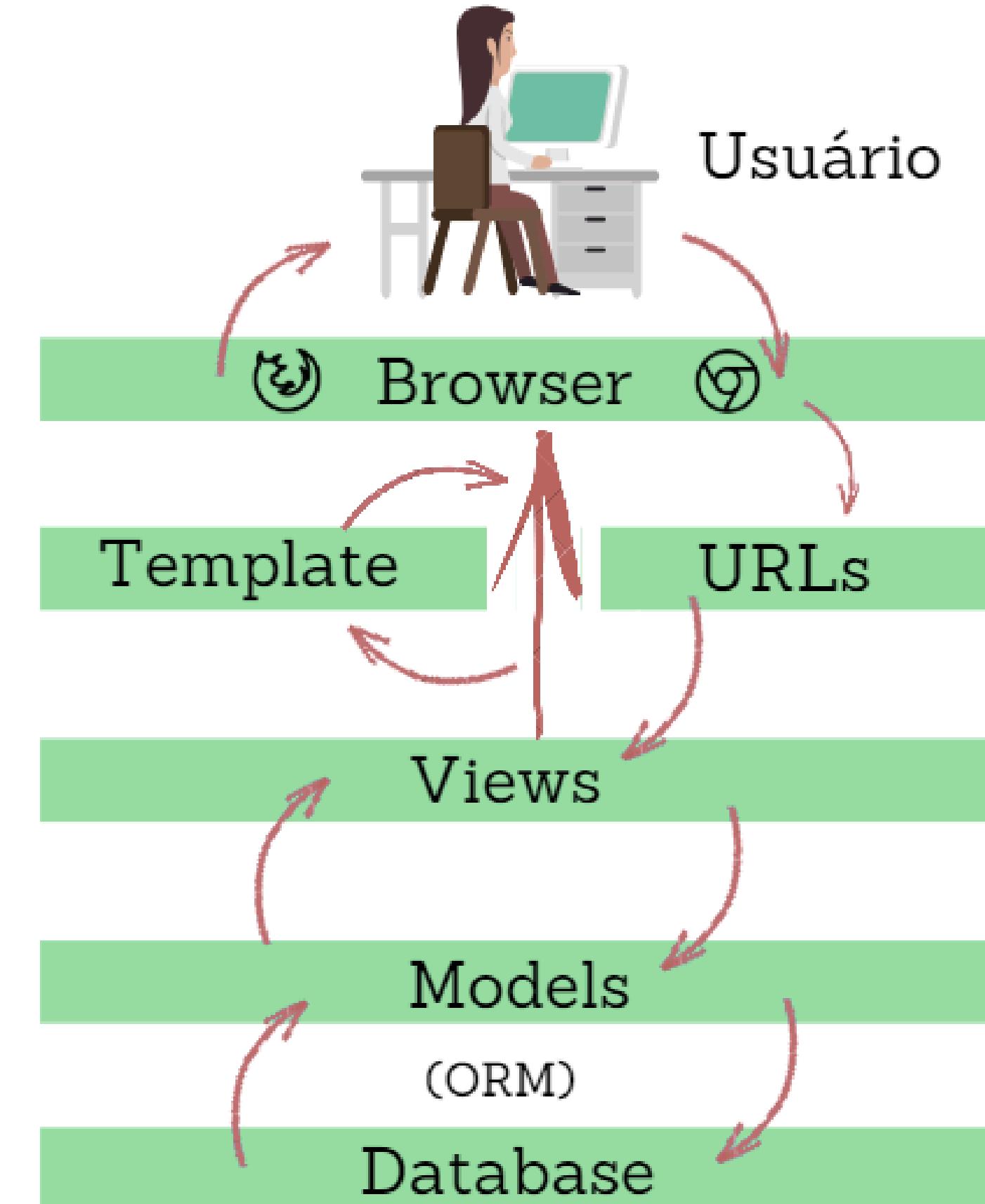


Framework web Python que promove o desenvolvimento rápido, ferramentas versáteis e eficientes para pessoas desenvolvedoras



# Arquitetura do Django

O Django utiliza a arquitetura MVT (Model, View e Template).



# O framework web para perfeccionistas com prazos

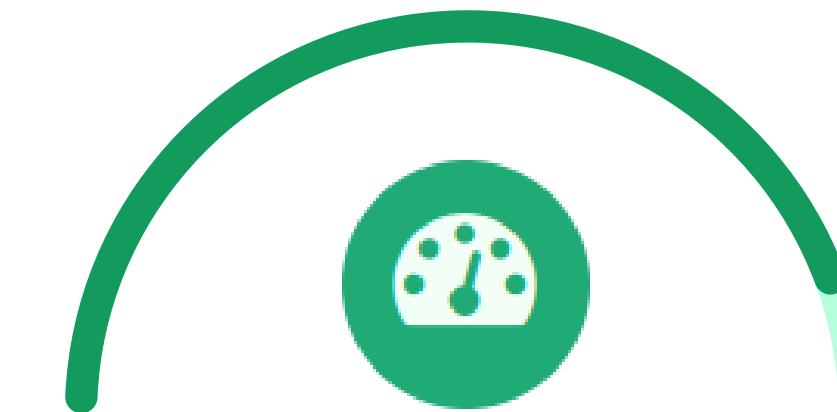
O Django facilita a criação de aplicativos da web melhores mais rapidamente e com menos código.



Ridiculamente  
rápido



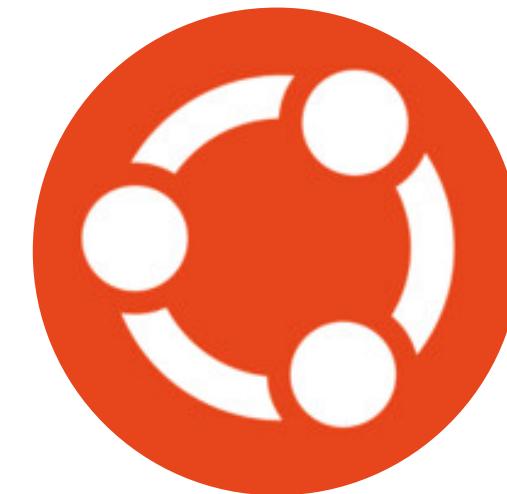
Tranquilizadoramente  
seguro



Extremamente  
escalável

# Empresas que usam Django

Exemplo de empresas que adotam o Django em seus sistemas

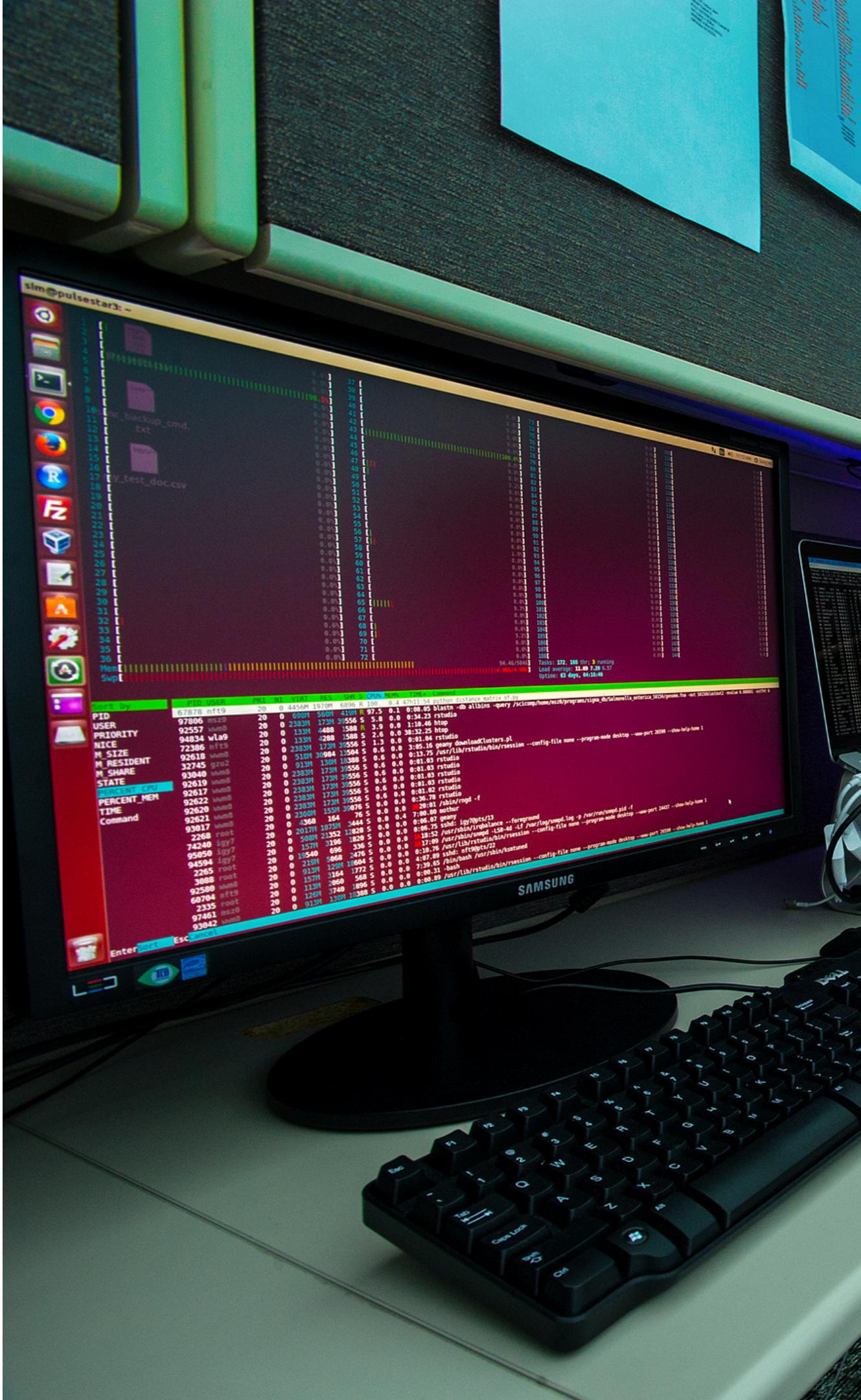


3.

# Idealização do projeto

# O que será feito?

Neste minicurso, iremos desenvolver um sistema web básico que consiste em cadastrar, excluir, editar e concluir tarefas usando as ferramentas do Django.



## Listar de Tarefas

Título	Criado em	Data de entrega	Data de finalização	Ações
Escrever um poema para Julia	1 de Dezembro de 2024	2 de Dezembro de 2024	-	<a href="#">Conduir</a> <a href="#">Editar</a> <a href="#">Excluir</a>
Estudar Django	1 de Dezembro de 2024	10 de Outubro de 2025	-	<a href="#">Conduir</a> <a href="#">Editar</a> <a href="#">Excluir</a>
Jogar RPG de mesa em cima de um carro em movimento	1 de Dezembro de 2024	10 de Outubro de 2028	-	<a href="#">Conduir</a> <a href="#">Editar</a> <a href="#">Excluir</a>

[Nova Tarefa](#)

XVII SEMANA NACIONAL DE  
CIÊNCIA E TECNOLOGIA  
DE ITACOATIBA - 2024

4.

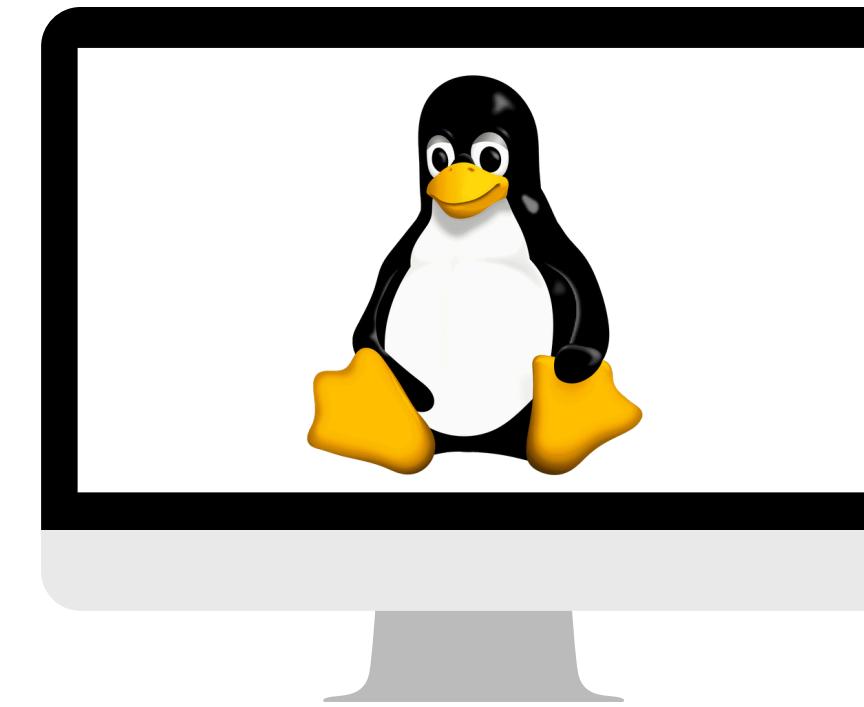
Início de um projeto  
web com Django

# Verificar o sistema operacional

Windows



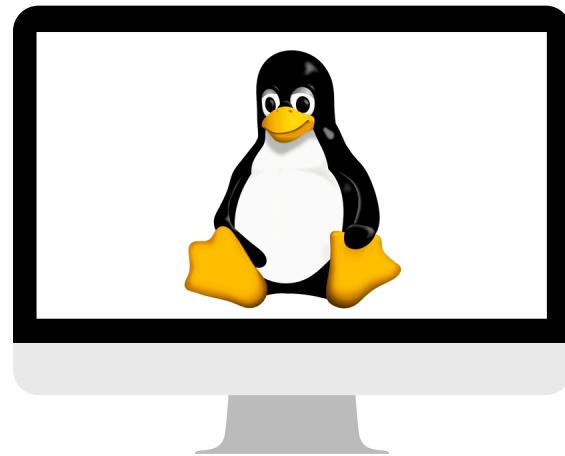
Linux



# Verificar python instalado



`python --version`

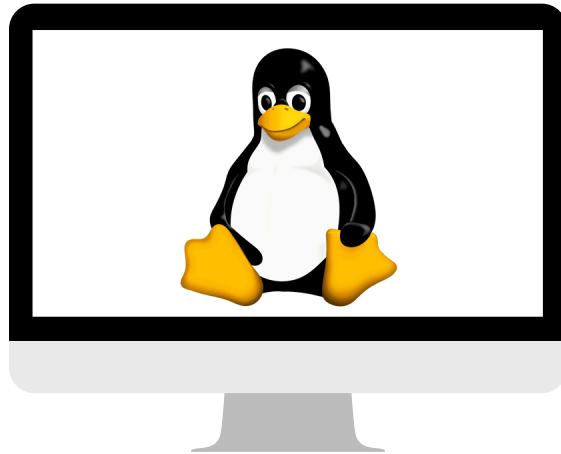


`python3 --version`

# Criar um ambiente virtual de desenvolvimento



`python -m venv venv`

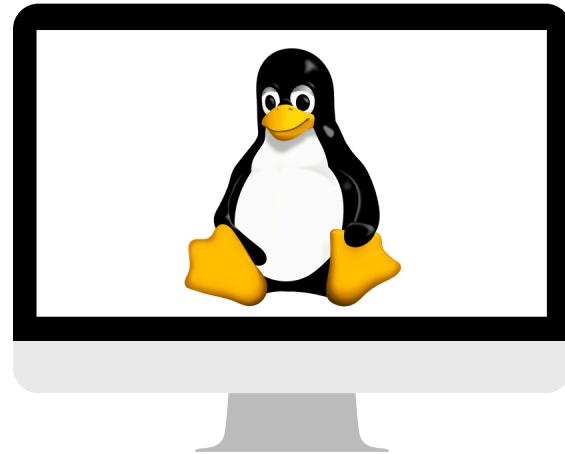


`python3 -m venv venv`

# Ativar o ambiente virtual de desenvolvimento



`venv\Scripts\activate`



`source venv/bin/activate`

# Erro ao ativar o ambiente virtual



```
venv\Scripts\activate : O arquivo
não pode ser carregado porque a execução de scripts foi desabilitada neste sistema. Para obter mais informações, consulte
about_Execution_Policies em https://go.microsoft.com/fwlink/?LinkID=135170.
No linha:1 caractere:1
+ venv\Scripts\activate
+ ~~~~~
+ CategoryInfo          : Erro de Segurança: () [], PSSecurityException
+ FullyQualifiedErrorId : UnauthorizedAccess
```

# Erro ao ativar o ambiente virtual

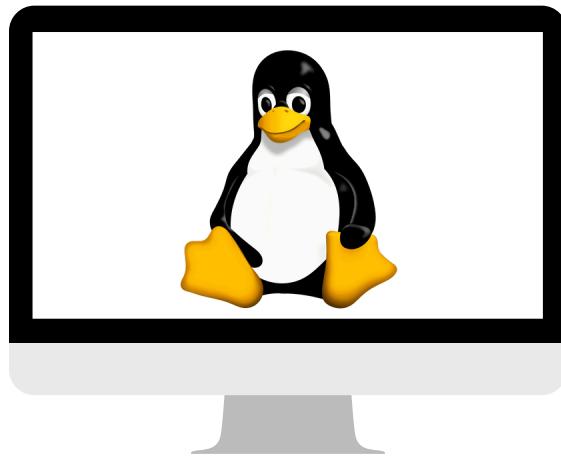


`Set-ExecutionPolicy -Scope Process -ExecutionPolicy Bypass`

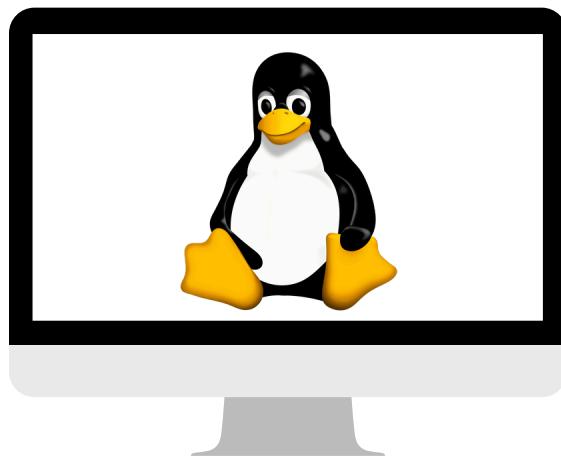
# Instalar o Django



**pip install django**



# Iniciar um Projeto Django

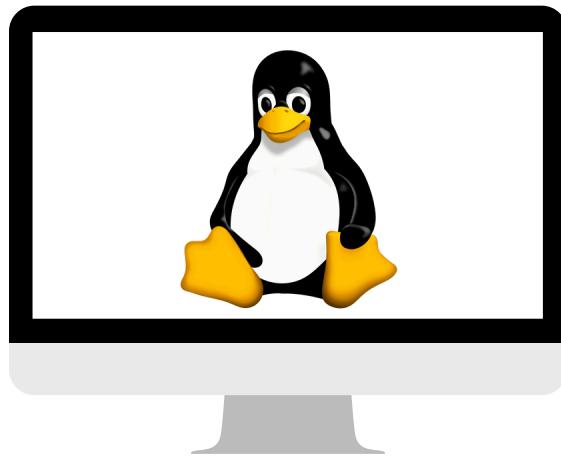


**django-admin startproject core .**

# Testar o Projeto Django



`python manage.py runserver`



`python3 manage.py runserver`

# Testar o Projeto Django



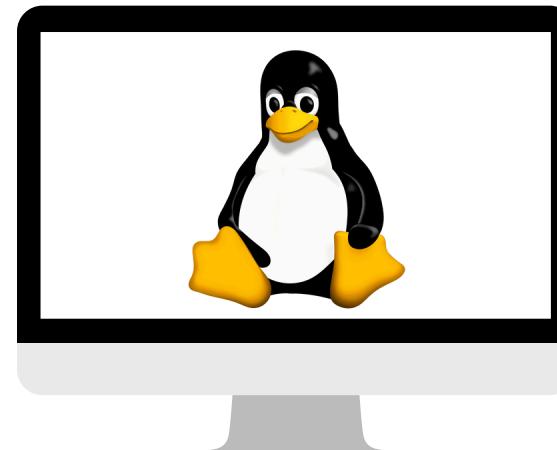
The install worked successfully! Congratulations!

[View release notes for Django 5.1](#)

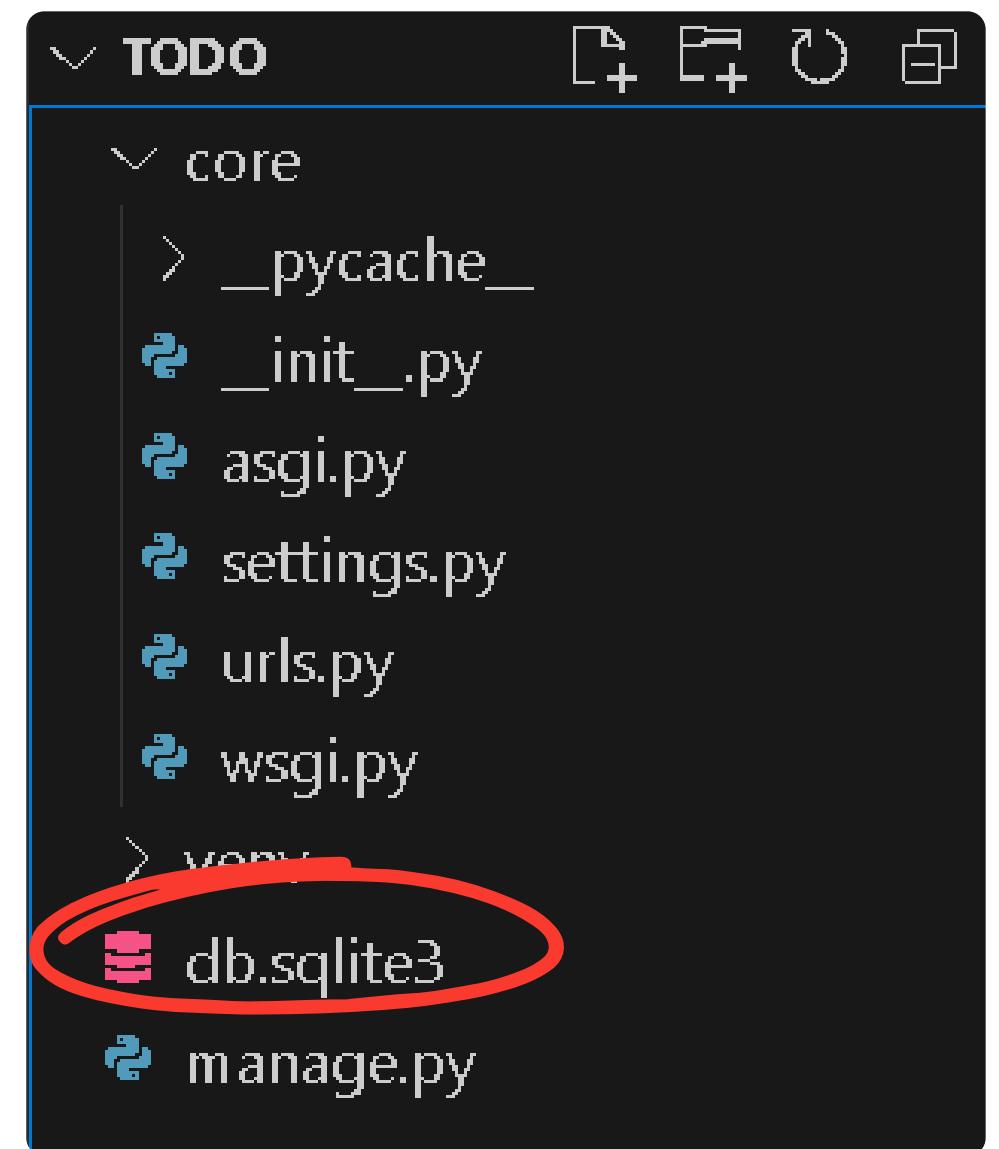
You are seeing this page because `DEBUG=True` is in your settings file and you have not configured any URLs.

**django**

# Testar o Projeto Django



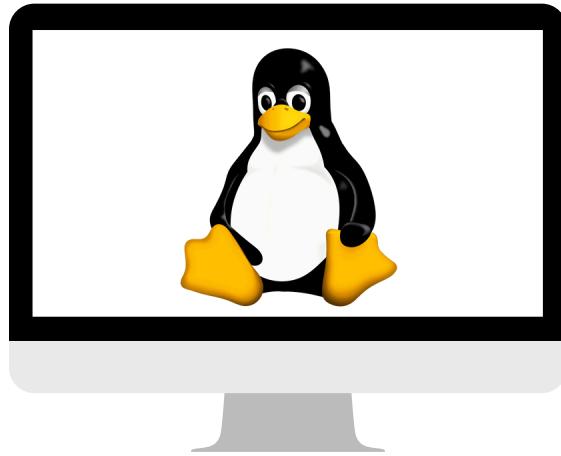
Banco de dados criado



# Testar o Projeto Django



`python manage.py startapp todo`

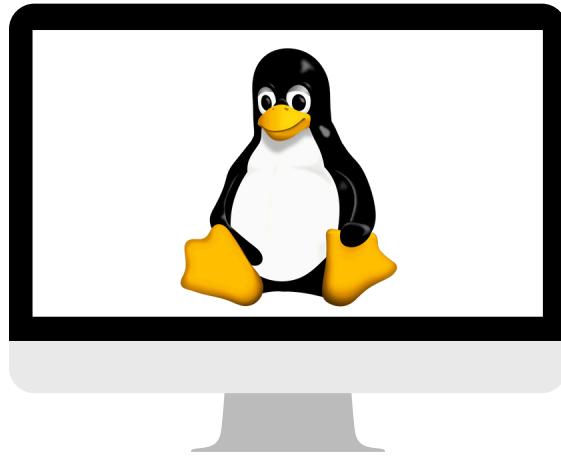


`python3 manage.py startapp todo`

# Testar o Projeto Django

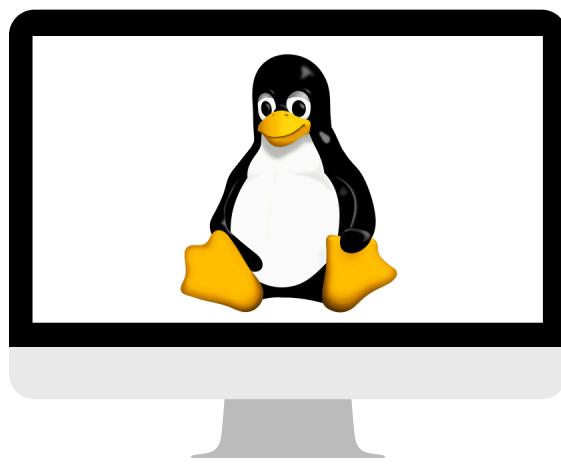


`python manage.py makemigrations`



`python manage.py migrate`

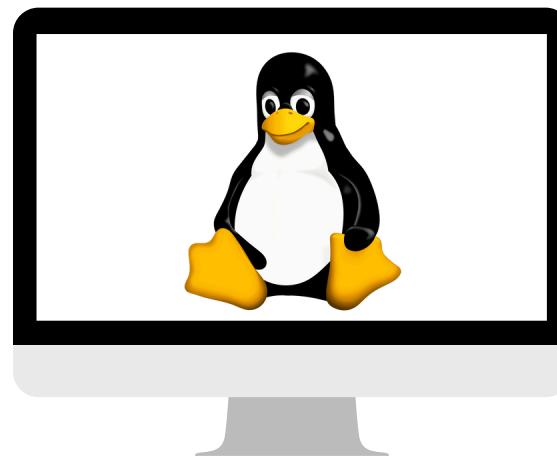
# Testar o Projeto Django



**Estrutura esperada**

```
✓ TODO
  > core
  ✓ todo
    > migrations
    ✎ __init__.py
    ✎ admin.py
    ✎ apps.py
    ✎ models.py
    ✎ tests.py
    ✎ views.py
  > venv
  ⚡ db.sqlite3
  ✎ manage.py
```

# Configurar o Django(settings.py)

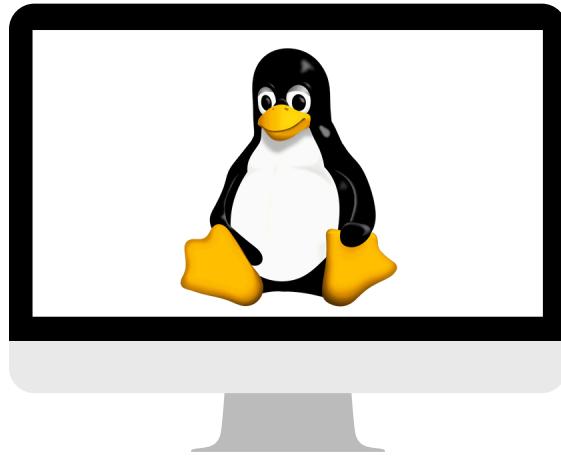


```
● ● ●  
1 LANGUAGE_CODE = 'pt-br'  
2  
3 TIME_ZONE = 'America/Manaus'  
4  
5 USE_I18N = True  
6  
7 USE_TZ = True
```

# Instalar o Crispy

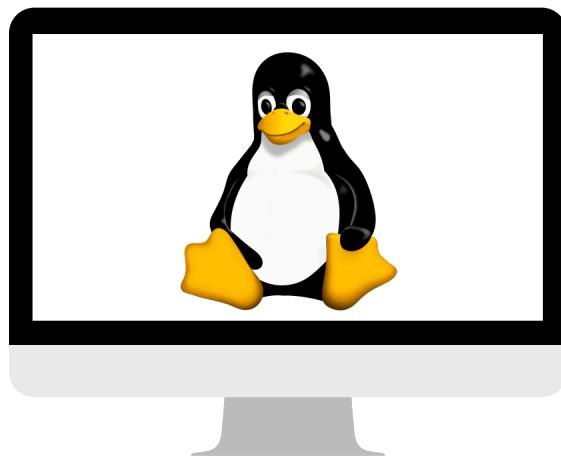


`pip install django-crispy-forms`



`pip install crispy-bootstrap5`

# Configurar o arquivo settings.py



```
1 INSTALLED_APPS = [
2     'django.contrib.admin',
3     'django.contrib.auth',
4     'django.contrib.contenttypes',
5     'django.contrib.sessions',
6     'django.contrib.messages',
7     'django.contrib.staticfiles',
8     'todo',
9     'crispy_forms',
10    'crispy_bootstrap5',
11 ]
```

# Configurar o Django



```
1 CRISPY_ALLOWED_TEMPLATE_PACKS = "bootstrap5"  
2  
3 CRISPY_TEMPLATE_PACK = "bootstrap5"
```

# Hello World View



```
1 from django.shortcuts import HttpResponse  
2  
3 def hello(request):  
4     return HttpResponse("Hello World")
```

# Hello World View



```
1 from .views import hello
2
3 urlpatterns = [
4     path("", hello, ),
5 ]
```

# Criar o model ToDo



```
1 class Todo(models.Model):
2     title = models.CharField(verbose_name="Título", max_length=100, null=False, blank=False)
3     created_at = models.DateTimeField(auto_now_add=True, null=False, blank=False)
4     deadline = models.DateField(verbose_name="Data de Entrega", null=False, blank=False)
5     finished_at = models.DateField(null=True)
```

# Criar o url ToDo

```
from django.contrib import admin
from django.urls import path, include
import todo.urls as todo_urls

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include(todo_urls))
]
```

# Criando a view



```
1 from .models import Todo  
2 from django.views.generic import ListView  
3  
4 class TodoListView(ListView):  
5     model = Todo
```

# Criando a rota

```
● ● ●  
1 from django.contrib import admin  
2 from django.urls import path, include  
3 import todo.urls as todo_urls  
4  
5 urlpatterns = [  
6     path('admin/', admin.site.urls),  
7     path('', include(todo_urls))  
8 ]  
9
```

# Criando a rota

```
● ● ●  
1 from django.urls import path  
2  
3 from todo.views import (  
4     TodoListView,  
5 )  
6  
7  
8 urlpatterns = [  
9     path("", TodoListView.as_view(), name="todo_list"),  
10 ]
```

# Criando o HTML (todo\_list)



```
1  {% load static %}  
2  
3  <!DOCTYPE html>  
4  <html lang="pt-br">  
5    <head>  
6      <!-- Meta tags Obrigatórias -->  
7      <meta charset="utf-8">  
8      <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=0, height=100%">  
9      <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw3mqlq/BK3e0JI6BtguZchuNebqzCuEpeqcjCvEiwng65Wj7r8QsK1Gh5" crossorigin="anonymous">  
10     <title>To Do List</title>  
11     <link rel="stylesheet" href="{% static 'css/style.css' %}">  
12   </head>
```



```
1  
2  .  
3  .  
4  .  
5  <script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+9jwhV3D2O+tbhMbbA1h0AmwEzqeaVQz19u5mT+JNk" crossorigin="anonymous">  
6  <script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.3/dist/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV+2X1Bh4YJ4gD0CX3MISqYVXc1jEF8gWzq5sZUcu8c0LkX7zP9" crossorigin="anonymous">  
7  <script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCnJSK3+MXMIPGHN6+aP90/EEzdgndu46Dc6GW3F0XW1z0XQZ8nhWu" crossorigin="anonymous">  
8  </body>
```

# Importar o Bootstrap

Acesse o site: <https://getbootstrap.com.br/docs/4.1/getting-started/introduction/>

## CSS

Copie e cole o arquivo de estilo `<link>` dentro da sua `<head>` antes de todos os outros arquivos de estilo para carregar nosso CSS.

```
<link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3GXwEC2EjYH45D7F1o4zUXnZrgyQVZtmf27l76vqE+S1oJw" data-bbox="136 485 875 520"> Copy
```

## JS

Muitos dos nossos componentes precisam de JavaScript para funcionar. Mais especificamente, eles precisam do [jQuery](#), [Popper.js](#) e do nossos próprios plugins JavaScript. Coloque os seguintes `<script>`s perto do final da sua página, logo antes do fechamento da tag `</body>`. jQuery tem que vir antes, depois o Popper.js e só depois nossos plugins JavaScript.

Nós usamos a [build slim do jQuery](#), mas a versão completa também é suportada.

```
<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js" integrity="sha384-q8i/X+965DzO0rT7abK41JStQIAqVgRVzbzo5smXKp4YfRvh+8abtTE1Pi6" data-bbox="136 810 875 885"> Copy  
<script src="https://cdn.jsdelivr.net/npm/popper.js@1.14.3/dist/umd/popper.min.js" integrity="sha384-ZMP7rVo3mIykV4cFq�J02H45jfJEsrcq1ZkPdRlIGoqJZGK5h7WvO951qK" data-bbox="136 845 875 885"> Copy  
<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.1.3/js/bootstrap.min.js" integrity="sha384-ChfqqxuZUCnJSK3+MXmPNIyE6ZbwH2IMqE241r" data-bbox="136 875 875 885"> Copy
```

# Começando o corpo do HTML

```
1 <body>
2   <nav class="navbar navbar-light mb-4" id="navegar">
3     <a class="navbar-brand text-light fw-bold" href="{% url 'todo_list' %}"><strong>SNCT To Do</strong></a>
4   </nav>
5
6   <main class="container">
7     <h2>Lista de Tarefas</h2>
8
```

# Começando o corpo do HTML

```
1 <main class="container">
2     <h2>Lista de Tarefas</h2>
3
4     {% if todo_list %}
5         <table class="table table-striped" id="table_list">
6             <thead id="head_table">
7                 <tr>
8                     <th scope="col">Título</th>
9                     <th scope="col">Criado em</th>
10                    <th scope="col">Data de entrega</th>
11                    <th scope="col">Data de finalização</th>
12                    <th scope="col">Ações</th>
13                </tr>
14            </thead>
```

# Começando o corpo do HTML

```
1 <tbody>
2     {% for todo in todo_list %}
3         <tr>
4
5             <td>{{todo.title}}</td>
6             <td>{{todo.created_at}}</td>
7             <td>{{todo.deadline}}</td>
8             <td>{{todo.finished_at|default:"-"}}</td>
9             <td id="func">
10                <a href="{%url 'todo_complete' pk=todo.pk%}"
11                   class="btn btn-primary btn-sm {%if todo.finished_at %}dis-
12                   abled{%endif%}">Concluir</a>
13                <a href="{%url 'todo_update' pk=todo.pk %}"
14                   class="btn btn-success btn-sm {%if todo.finished_at %}dis-
15                   abled{%endif%}">Editar</a>
16                <a href="{%url 'todo_delete' pk=todo.pk %}" class="btn bt-
17                   n-danger btn-sm">Excluir</a>
18            </td>
19        </tr>
20    {% endfor %}
21    </tbody>
22    </table>
23    {% else %}
24        <p>Não há tarefas cadastradas</p>
25    {% endif %}
```



```
1 <a href="{% url "todo_create"%}" class="btn btn-primary"> Nova Tarefa <
2 </a>
3 </main>
```

# Criando a view

```
● ● ●  
1 from .models import Todo  
2 from django.views.generic import ListView, CreateView  
3  
4 class TodoCreateView(CreateView):  
5     model = Todo  
6     fields = ["title", "deadline"]  
7     success_url = reverse_lazy("todo_list")
```

# Criando a view



```
1 from .models import Todo
2 from django.views.generic import ListView, CreateView, UpdateView
3
4 class TodoUpdateView(UpdateView):
5     model = Todo
6     fields = ["title", "deadline"]
7     success_url = reverse_lazy("todo_list")
```

# Criando a rota

```
● ● ●  
1  from django.urls import path  
2  
3  from todo.views import (  
4      TodoListView,  
5      TodoCreateView,  
6      TodoUpdateView,  
7  )  
8  
9  
10 urlpatterns = [  
11     path("", TodoListView.as_view(), name="todo_list"),  
12     path("create", TodoCreateView.as_view(), name="todo_create"),  
13     path("update/<int:pk>", TodoUpdateView.as_view(), name="todo_update"),  
14 ]
```

# Editando/cadastrando (todo\_form)

```
● ● ●  
1  {% load crispy_forms_tags%}  
2  {% load static %}  
3  
4  <!DOCTYPE html>  
5  <html lang="pt-br">  
6    <head>  
7      <!-- Meta tags Obrigatórias -->  
8      <meta charset="utf-8">  
9      <meta name="viewport" content="width=device-width, initial-scale=1,  
shrink-to-fit=no">  
10     <link rel="stylesheet" href="https://stackpath.bootstrapcdn.com/boo  
tstrap/4.1.3/css/bootstrap.min.css" integrity="sha384-MCw98/SFnGE8fJT3G  
XwEOngeV7Zt27NXFoaoApmYm81iuXoPkFOJwJ8ERdknLPMO" crossorigin="anonymou  
s">  
11     <link rel="stylesheet" href="{% static 'css/style.css' %}">  
12     {% if todo.pk %}  
13         <title>Editar Tarefa</title>  
14     {% else %}  
15         <title>Nova Tarefa</title>  
16     {% endif %}  
17   </head>
```

# Editando/cadastrando (todo\_form)

```
● ● ●

1 <body>
2     <nav class="navbar navbar-light mb-4" id="navegar">
3         <a class="navbar-brand text-light fw-bold" href="{% url 'todo_1
ist' %}"><strong>SNCT To Do</strong></a>
4     </nav>
5
6     <main class="container">
7
8         {% if todo.pk %}
9             <h2>Editar Tarefa</h2>
10        {% else %}
11            <h2>Nova Tarefa</h2>
12        {% endif %}
13
14        <form method="POST">
15            {% csrf_token %}
16            {{form | crispy}}
17            <button type="submit" class="btn btn-primary"> Salvar </but
ton>
18        </form>
19    </main>
```

# Criando a view



```
1 from django.urls import reverse_lazy
2 from django.views.generic import ListView, CreateView, UpdateView, DeleteView
3
4 class TodoDeleteView(DeleteView):
5     model = Todo
6     success_url = reverse_lazy("todo_list")
```

# Excluir tarefa (todo\_confirm\_delete)



```
1 <main class="container">
2
3     <h2>Excluir Tarefa?</h2>
4
5         <p>Tem certeza que deseja excluir tarefa <strong>{{todo.title}}</strong> ?</p>
6
7         <form method="POST">
8             {% csrf_token %}
9
10            <button type="submit" class="btn btn-danger">Excluir</button>
11        <a href="{%url 'todo_list'%}" class ="btn btn-primary" >Cancelar</a>
12    </form>
13 </main>
```

# Criando a rota

```
● ● ●  
1  from django.urls import path  
2  
3  from todo.views import (  
4      TodoListView,  
5      TodoCreateView,  
6      TodoUpdateView,  
7      TodoDeleteView,  
8  )  
9  
10  
11 urlpatterns = [  
12     path("", TodoListView.as_view(), name="todo_list"),  
13     path("create", TodoCreateView.as_view(), name="todo_create"),  
14     path("update/<int:pk>", TodoUpdateView.as_view(), name="todo_update"),  
15     path("delete/<int:pk>", TodoDeleteView.as_view(), name="todo_delete"),  
16 ]
```

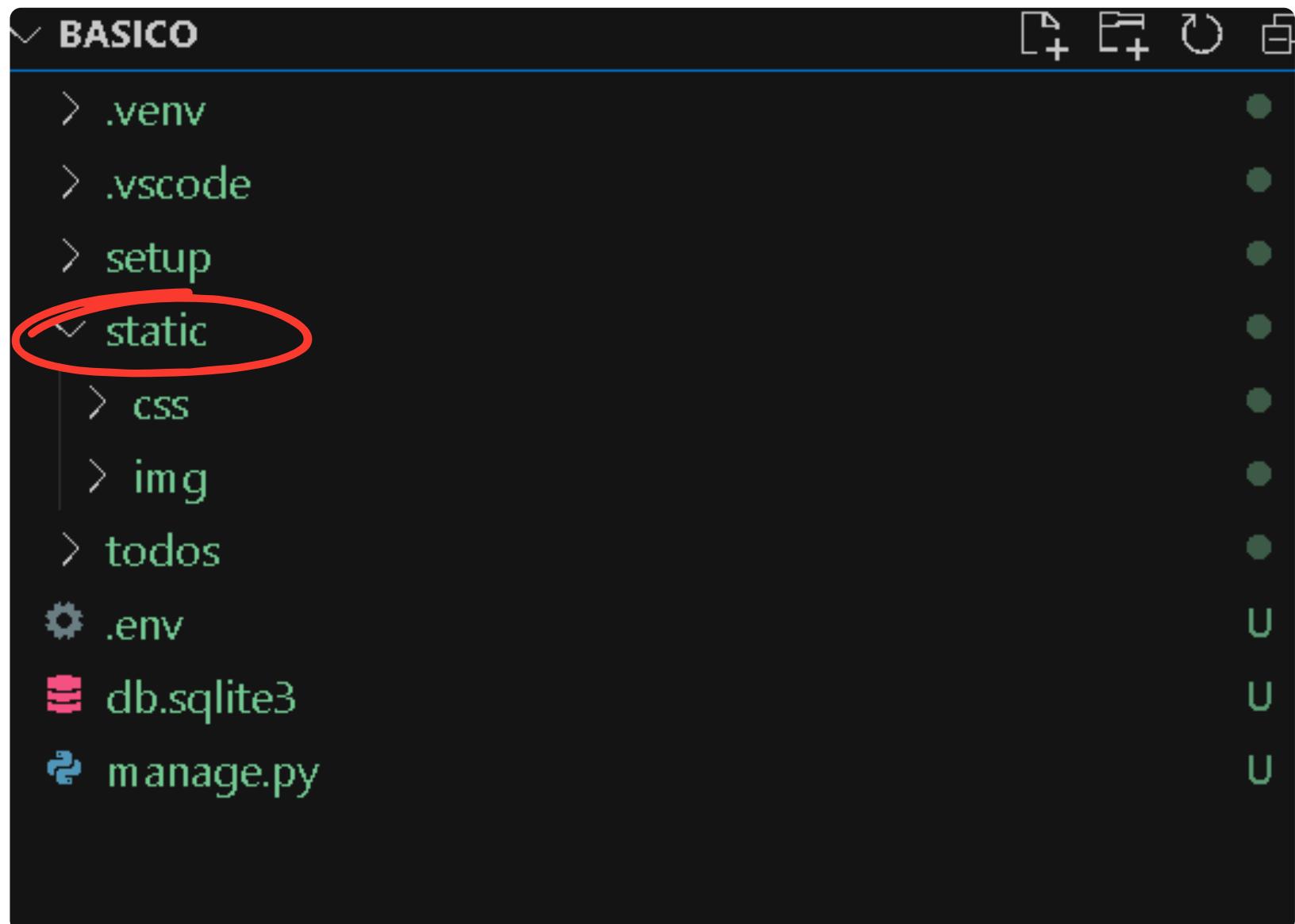
# Criando o model function

```
● ● ●  
1 class Meta:  
2     ordering = ["deadline"]  
3  
4     def mark_has_complete(self):  
5         if not self.finished_at:  
6             self.finished_at = date.today()  
7             self.save()
```

# Criando a view

```
● ● ●  
1 from .models import Todo  
2 from django.shortcuts import get_object_or_404, redirect  
3 from django.views.generic import ListView, CreateView, UpdateView, DeleteView, View  
4  
5 class TodoCompleteView(View):  
6     def get(self, request, pk):  
7         todo = get_object_or_404(Todo, pk=pk)  
8         todo.mark_has_complete()  
9         return redirect("todo_list")  
10
```

# Criando arquivos estáticos



- Crie uma pasta chamada static
- Crie uma pasta para o CSS

# No arquivo settings.py



```
1 STATIC_URL = "static/"  
2 STATICFILES_DIRS = [BASE_DIR / 'static']  
3 STATIC_ROOT = BASE_DIR / 'staticfiles'
```

# Estilização (CSS)



```
1 html, body {  
2     height: 100%;  
3     display: flex;  
4     flex-direction: column;  
5 }  
6  
7 #navegar{  
8     background: rgb(5, 126, 225);  
9 }
```



```
1 #table_list{  
2     overflow: hidden;  
3     border: 1px solid #ddd;  
4     border-radius: 10px;  
5 }  
6  
7 #head_table{  
8     background-color: rgb(5, 126, 225);  
9 }  
10 #head_table th{  
11     color: white;  
12 }  
13  
14 #func a{  
15     margin-right: 2px;  
16     margin-bottom: 2px;  
17 }
```