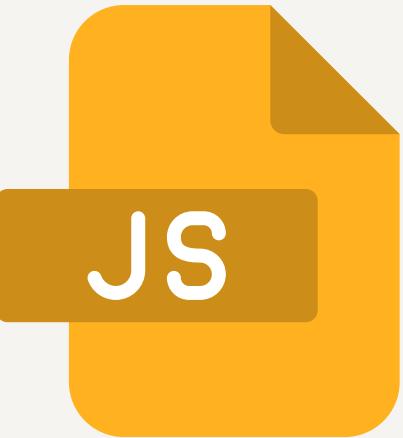
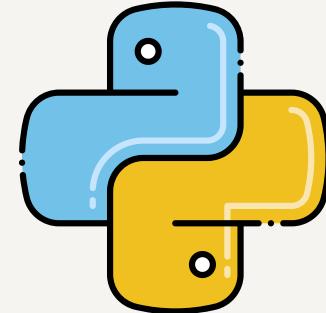


HTML



**STRAIGHT TO THE POINT
WITH**

django

Presented By

SUETONE CARNEIRO

VEJAMOS O QUE VAMOS APRENDER...



BREVE INTRODUÇÃO

Arquitetura básica do projeto Django

PREPARAÇÃO DO AMBIENTE E NOSSO PRIMEIRO PROJETO

Ambiente virtual

django admin startproject *project_name*

MOMENTO MODULARIZAÇÃO - CRIANDO APPS

python3 manage.py startapp *app_name*

CAMADA DE DADOS

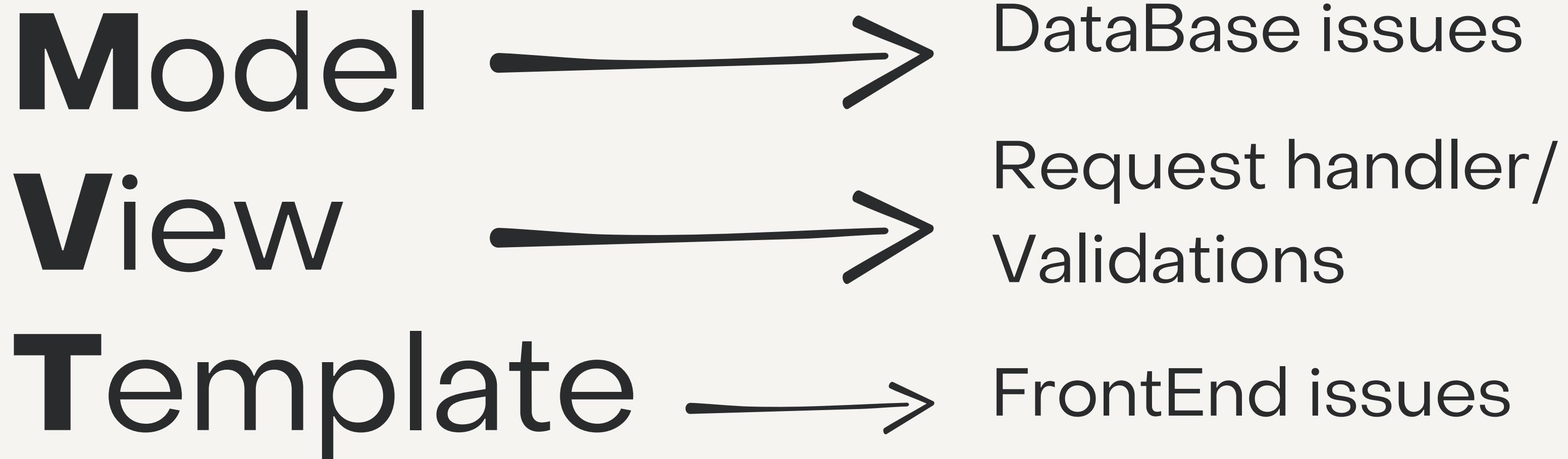
Models e migrations

DJANGO ADMIN

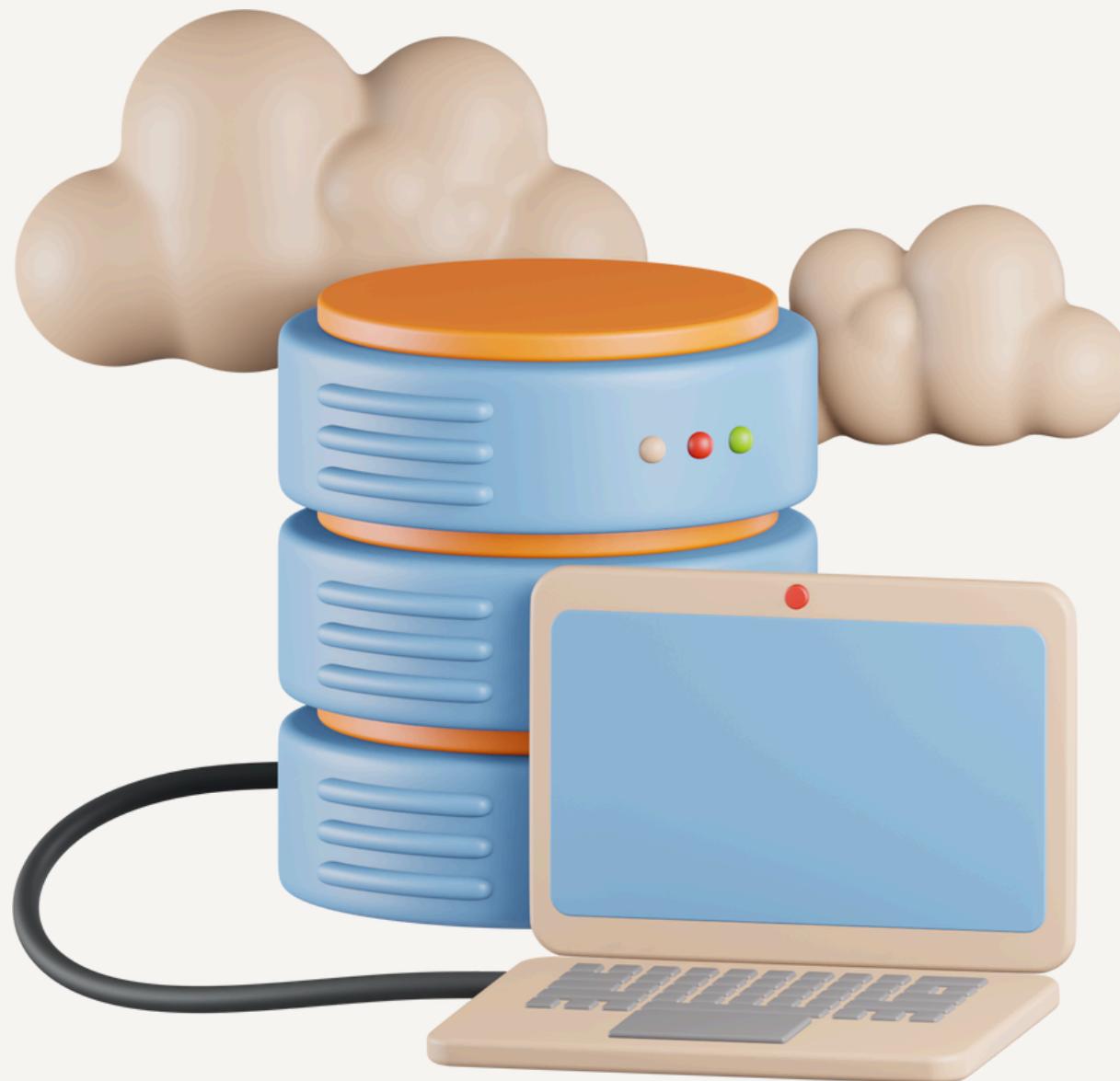
PASSO ZERO

INTRODUÇÃO

django



Model



SQL



ORM



(Object-Relational Mapping)

django



View

Lida com as requisições.
Organiza as rotas, ou seja, URLs

django



+ Algum tipo de validação, se necessário

Template

django



PREPARAÇÃO DE AMBIENTE

django

```
code@home:~$ python3 -m venv .venv  
code@home:~$ source .venv/bin/activate  
code@home:~$ pip install django  
code@home:~$ django-admin startproject [...]  
code@home:~$ python3 manage.py runserver
```

Lembrete: adicionar diretório para arquivos estáticos :)



Straight to the point with Django

MOMENTO MODULARIZAÇÃO

.....



COMO MANTER A ORGANIZAÇÃO?

django

“—

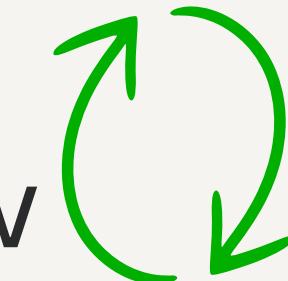
Django comes with a utility that automatically generates the basic directory structure of an app, so you can focus on writing code rather than creating directories.

[>> Django Documentation](#)

COMO MANTER A ORGANIZAÇÃO?

django

- Passo 1: `python3 manage.py startapp app_name`
- Passo 2: criar arquivo para as URLs do APP
- Passo 3: criar uma view
- Passo 4: criar a URL para essa view
- Passo 5: ‘aviser’ ao projeto que criamos novas rotas e um novo app (`settings.py -> installed apps`)
- Passo 6: desenvolver



*Depois que o app já
está criado...

Lembrete: adicionar
diretório para templates :)
`templates/app_name`

DJANGO E SUAS CLASS BASED VIEWS

django

• • • app_name/views.py

```
from django.views.generic import TemplateView

class IndexView(TemplateView):
    template_name = 'app_name/index.html'
```

DEPOIS DA VIEW... VEM AS ROTAS!

django



app_name/urls.py

```
from django.urls import path
from . import views

urlpatterns = [
    path('/', views.IndexView.as_view(
        template_name='app_name/index.html'
    ), name="index"),
]
```

CONFIGURE TAMBÉM NO PROJETO!

*Necessário apenas ao criar o app

● ● ● project_name/urls.py

```
from django.contrib import admin
from django.urls import include, path
```

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

CONFIGURE TAMBÉM NO PROJETO!

*Necessário apenas ao criar o app

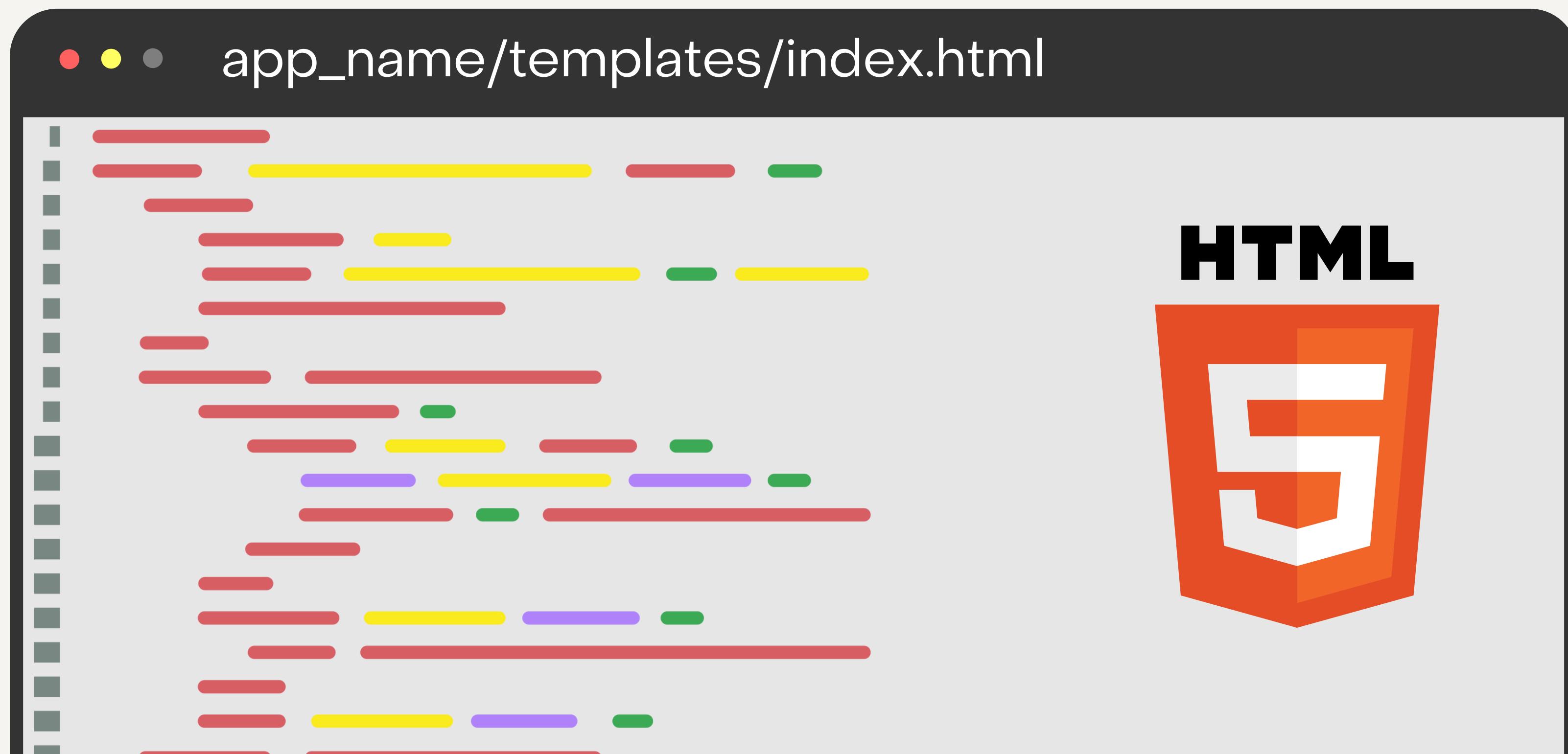
A smartphone icon with three colored dots (red, yellow, blue) at the top left. The screen displays the text "project_name/settings.py".

project_name/settings.py

```
INSTALLED_APPS = [  
    'django.contrib.admin',  
    'django.contrib.auth',  
    'django.contrib.contenttypes',  
    'django.contrib.messages',  
    'django.contrib.staticfiles',  
    'app_name.apps.App_nameConfig',  
]
```

POR FIM, NOSSO QUERIDO
TEMPLATE:)

django



Straight to the point with Django

CAMADA DE DADOS

.....

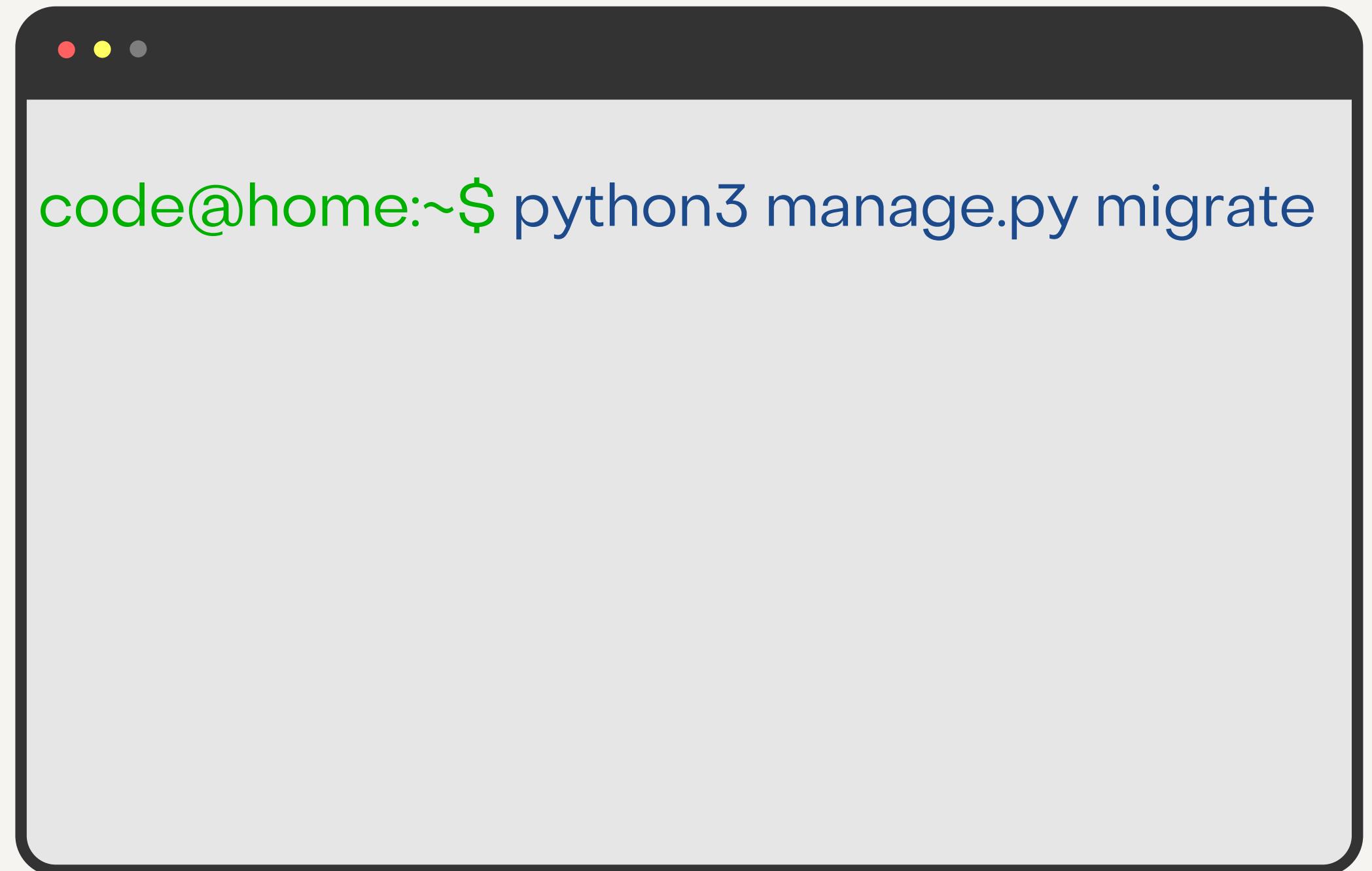


CAMADA DE DADOS

django

Vamos fazer o primeiro teste?

O objetivo é que o django crie as tabelas que já vêm por padrão com ele.



```
code@home:~$ python3 manage.py migrate
```

django

E QUANDO EU CRIAR MINHAS PRÓPRIAS TABELAS?

```
code@home:~$ python3 manage.py makemigrations  
code@home:~$ python3 manage.py migrate
```

-> Com esses dois comandos, o Django
vai criar suas tabelas no banco :)

Vejamos na prática.



E QUANDO EU CRIAR MINHAS PRÓPRIAS TABELAS?

django

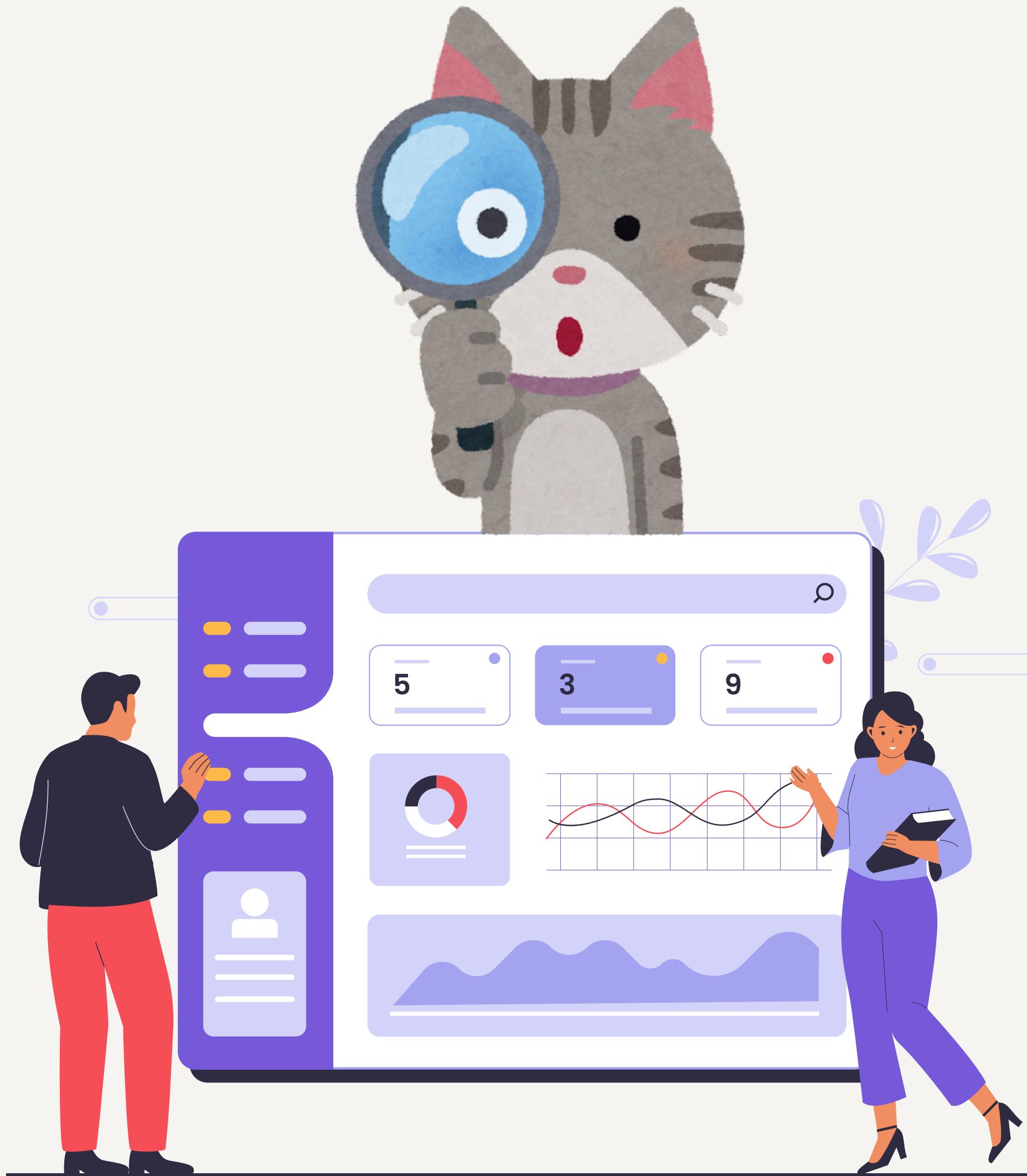
```
from django.db import models
from django.contrib.auth.models import User } ←

class Usuario(models.Model):
    usuario = models.OneToOneField(User, on_delete=models.CASCADE)
    email = models.EmailField(unique=True, null=False)
    nome_completo = models.CharField(
        max_length=255, verbose_name='Nome Completo')
    endereco = models.CharField(max_length=255, verbose_name='Endereço')
    telefone = models.CharField(max_length=22, verbose_name='Telefone')

    def __str__(self):
        return f'{self.nome_completo} - {self.email}'
```

Straight to the point with Django

DJANGO ADMIN



DJANGO ADMIN

The Django logo, consisting of the word "django" in a lowercase, sans-serif font, with the "d" being dark green and the rest of the letters being white.

O django conta com um painel administrativo built in e que pode ser personalizado (uau).

Okay. Mas... como acessar?

Existe uma rota que provavelmente já vimos...

A rota é: **admin/**

DJANGO ADMIN

Creating a super user

Usemos a
documentação :)



django

Creating an admin user

First we'll need to create a user who can login to the admin site. Run the following command:

 \$ python manage.py createsuperuser

Enter your desired username and press enter.

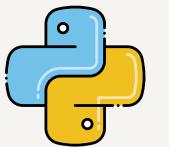
Username: admin

You will then be prompted for your desired email address:

Email address: admin@example.com

The final step is to enter your password. You will be asked to enter your password twice, the second time as a confirmation of the first.

Password: *****
Password (again): *****
Superuser created successfully.



STRAIGHT TO THE POINT
WITH

django

Presented By
SUETONE CARNEIRO



THE
END