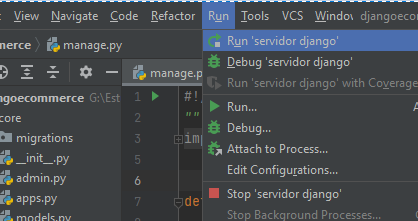
Python

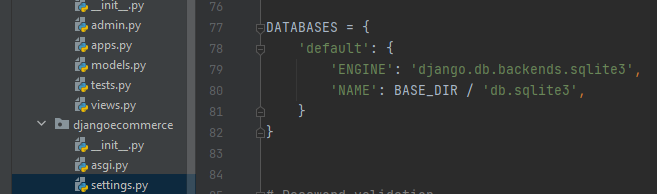
- Sempre iniciar o ambiente virtual do django executando o manage.py (servidor django).



**OU**

---------------------------------------------------------------------------------------------------

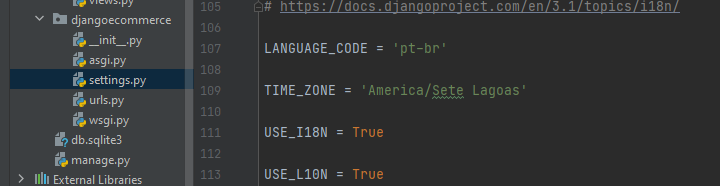
-arquivo settings.py= é o arquivo com as configurações gerais do django, ele é responsável onde eu vou dizer desde o banco de dados que vou escolher, templates, arquivos estáticos, configurações de e-mails.Ex:



\*LANGUAGE\_CODE= 'pt-br'

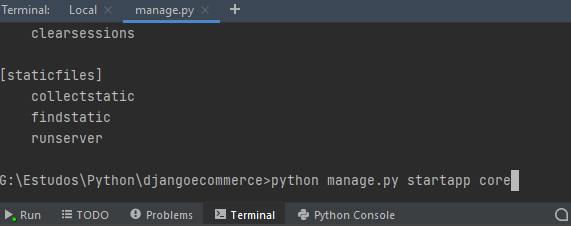
\*TIME\_ZONE = 'America-Minas Gerais'

-sqlite3 não necessitaria um sistema de gerenciamento de dados ativos.



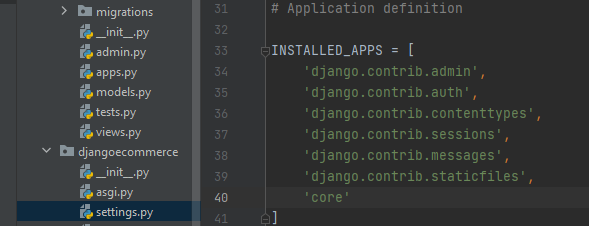
---------------------------------------------------------------------------------------------------

-Criar uma nova app, utilizando o terminal é só rodar o comando python manage.py startapp nomeDaNovaAplicação, no exemplo colocamos 'core' como nome



---------------------------------------------------------------------------------------------------

-DEpois de criar a aplicação core eu preciso dizer ao django que existe essa aplicação no sistema, para isso eu faço a configuração no arquivo sttings.py da pasta principal do projeto onde está o servidor, dentro do arquivo tem uma configuração chamada installed\_APPS, só incluir o 'nomeDaNovaAplicação', no nosso caso 'core'.



---------------------------------------------------------------------------------------------------

-Após isso no arquivo views.py da aplicação core eu devo fazer nossas configurações de views. Sempre que vou precisar criar uma página eu tenho que ter uma view pra ela.



Toda função recebe uma requisição e retorna uma resposta

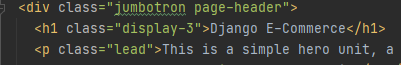
Importa o HttpResponse

---------------------------------------------------------------------------------------------------

Após isso, para inserir uma pagina válida para resposta a essa requisição do usuário eu devo primeiro informar todos templates no arquivo settings.py do meu projeto, na função templates. Essa função irá verificar os templates dentro da pasta template do meu app.

--------------------------------------------------------------------------------------------------

O template do django consite em um CONTEXTO e uma LINGUAGEM DO TEMPLATE, a linguagem do template significa que eu posso ter marcações especiais que irão me informar alguma coisa, exemplo, com a marcação abaixo (trocando o título ‘Django E-Commerce’ por ‘{{title}}’) indica ao django que eu vou renderizar uma variável, a marcação para variável é essa, {{}}.



Para isso eu tenho que criar um contexto para essa variável la na minha view da seguinte forma: criando o contexto e depois passando como parâmetro no meu template.



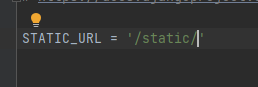
---------------------------------------------------------------------------------------------------

Temos também os filtros do django, para indicar um filtro eu uso o “|”, a barra na vertical mais o nome do filtro, exemplo o capfirst que deixa a primeira letra maiuscula

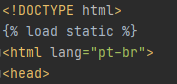


---------------------------------------------------------------------------------------------------

Para arquivos estáticos normalmente configuramos um servidor como o apache para rodar esse tipo de arquivo pois são mais específicos para isso. Porém neste exemplo vamos rodar no próprio servidor Django, para isso temos que ter uma pasta chamada static dentro de cada aplicação, é nela que ficará os arquivos estáticos. Para o django encontrar esses arquivos static temos que definir no arquivo settings.py o STATIC\_URL=’’ passando o caminho, exemplo:



Para que fazer com que a url que chamei no stylesheet do arquivo index.html eu posso utilizar a biblioteca static de template tag, é só iniciar ela no começo do arquivo {% load static %} e depois chamar essa tag nas urls do css e java script “{% static ‘’ %}”, exemplo:

 e depois



Fazendo isso no css lá embaixo da página também

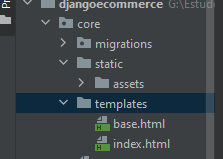


---------------------------------------------------------------------------------------------------

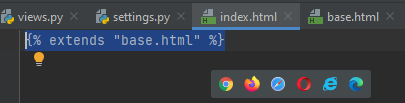
**Herança com Django:**

Se o site usa o mesmo corpo tanto na página inicial quando nas outras páginas, mudando apenas uma parte do corpo da pág podemos usar a herança, para isso vamos:

- Criar um arquivo base na pasta template com o nome de base.html, todo o conteúdo base do projeto irá ficar nesse arquivo:



-Em todas as páginas que vão herdar o conteúdo dessa página base eu vou colocar a tag django {% extends “base.html” %}, colocando o nome do arquivo que quero herdar dele.



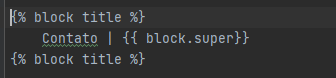
-No arquivo base eu devo indicar através de blocos, os blocos de código em que eu terei alterações, que no lugar é um possível local onde eu posso criar alterações, possíveis códigos html nos arquivos filhos: Veja abaixo:



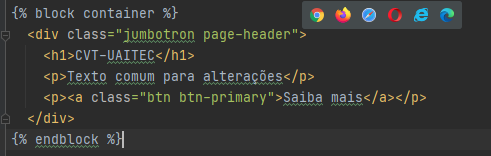
-Podemos colocar também essa tag block no título, para que cada pág tenha seu título default, que demonstre q posso ter códigos html para aquela parte do código também la no documento filho:



Ainda sobre esse block title, nas outras páginas podemos usar essa tag e alterar o título da pag filho, só colocarmos o código abaixo: Nesse exemplo resultado sairá “Contato | CVT-UAITEC” pois o {{block.super}} pega o título super(do html base).

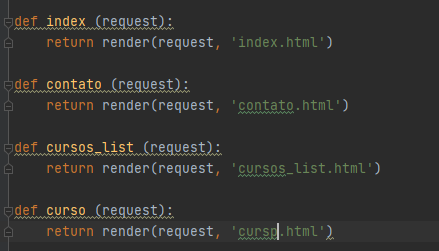


-Após isso, na pág que estamos herdando vamos colocar os trechos de código que querendo substituir essas partes em específico do nosso código base, no exemplo na pag index.html q estamos herdando da base.html , colocamos o bloco de código que será diferente da seguinte forma:



--------------------------------------------------------------------------------------------

-Feito todos os templates de cada página, tenho que identificar eles no arquivo de views, views.py. no nosso exemplo temos 4 templates, produtos\_list, produto, index, contato e ficará na seguinte forma dentro do views.py



---------------------------------------------------------------------------------------------------

-Temos também que declara-los como urlpatterns, para isso vamos no arquivo de urls, urls.py e declaramos da seguinte forma:

---------------------------------------------------------------------------------------------------

Observações:

Python sempre utiliza o padrão de variáveis Snake\_Case, exemplo **nome\_Esposa** e não o camelCase exemplo **nomeEsposa**.

pesquisar expressão regulares do python.

Getbootstrap site com templates

BootStrap