Criação de página web com Django

Os ficheiros Javacscript, CSS, imagens, texto, são **ficheiros estáticos** que para serem utilizados/exibidos devem-se fazer algumas configurações do **settings.py**:

Em INSTALLED_APPS deve constar: 'django.contrib.staticfiles' e a linha para indicar o caminho da pasta **static** (STATIC_URL = "static/"), pasta que deve ser criada numa pasta de nome '**templates**' que , por sua vez, terá de estar na raiz do projeto.

Deverá também ser definido o local dos ficheiros estáticos e guardados na varável STATICFILES_DIRS.

Importação de um template bootstrap e algumas configurações do mesmo

▷ Criar uma nova pasta com o nome do projeto 'Ficha9' e criar um ambiente virtual nela.

▷ Instalar o django (>> pip install django).

▷ Criar o projeto F9 (>>django-admin startproject F9 .).

▷ Criar uma app de nome paginas(>>python manage.py startapp paginas).

▷ Registar a app 'paginas' nos ficheiros settings.py e urls.py do projeto 'F9'.

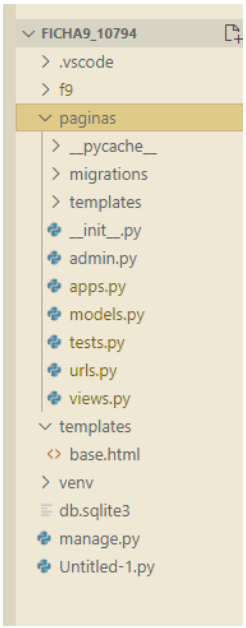
```
from django.contrib import admin
from django.urls import path, include

urlpatterns = [

    path("admin/", admin.site.urls),
    path('', include('paginas.urls')),

]
```

Repara que o caminho é vazio porque criámos apenas uma app, que contém os 2 ficheiros html a serem exibidos



▷ Na 'app', cria uma pasta de nome 'templates' e dentro dela outra pasta, agora, com o nome 'paginas'.

▷ Dentro da pasta 'paginas' cria 2 ficheiros html com o nome, respetivamente, sobre.html e index.html

▷ Cria, na raiz do projeto 'Ficha9' (a 1.ª pasta criada) uma pasta de nome ´templates', onde serão armazenadas as páginas html. Dentro dela o ficheiro '***base.html'.***

▷ Na app 'paginas', no ficheiro views.py, procede à definição dos métodos de resposta do servidor web ao pedido de exibição dos templates definidos em sobre.html e index.htm:

Settings | base.html | settings.py | views.py 1

paginas > views.py > ...

```python
from django.shortcuts import render

def index(request):
    return render(request, 'paginas/index.html')

def sobre(request):
    return render(request, 'paginas/sobre.html')
```

▷ Na app 'paginas', no ficheiro urls.py, indica o path de acesso aos métodos criados no passo anterior:

paginas > urls.py > ...

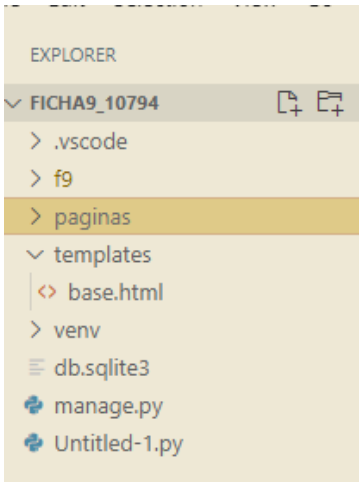
```python
from django.urls import path
from . import views

urlpatterns = [
    path('', views.index, name='index'),
    path('sobre/', views.sobre, name='sobre'),
]
```

Indicamos agora um nome para referir nos link do template

▷ Abre o ficheiro settings.py do projeto 'F9'.

Procura o dicionário de nome TEMPLATES e altera o valor da chave 'DIRS', indicado ao django onde procurar os ficheiros de templates. Adiciona o código seguinte no valor da chave: BASE_DIR / ´templates', que significa o path da raiz do projeto /caminho até 'templates'.



settings.py | Settings | urls.py f9 3 | urls.py paginas 1 | views.py 1

f9 > settings.py > ...

```python
]

ROOT_URLCONF = "f9.urls"

TEMPLATES = [
    {
        "BACKEND": "django.template.backends.django.DjangoTemplates",
        "DIRS": [BASE_DIR/'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",
            ],
        },
    },
]
```

▷ Faz download da pasta zipada bootstrap.zip, na Classroom, a partir do link para envio da ficha de trabalho nº9. Descompacta-a.

▷ Copia para o ficheiro de nome '**base.html** ', dentro da pasta 'paginas', o código do ficheiro ***index.html***, que se encontra na pasta bootstrap.
(Outros templates procura em: https://startbootstrap.com/)

▷ No código html, apagar o correspondente ao que se encontra selecionado na imagem seguinte, para aí se colocar html mais personalizado:

```
<!-- Page Content -->
<div class="container">
  <div class="row">
    <div class="col-lg-12 text-center">
      <h1 class="mt-5">Template básico bootstrap</h1>
      <p class="lead">
        Lorem ipsum dolor sit amet, consectetur adipiscing elit.
        Praesent egestas vulputate sollicitudin. Proin ac facilisis velit.
        Vestibulum tristique pharetra lorem eget tempor. Ut eu nunc auctor mi
        venenatis tristique a quis nibh. Sed accumsan congue est in convallis.
        Donec fermentum nisl id odio blandit volutpat. Curabitur purus leo,
        egestas a tortor vitae, venenatis ultrices elit.
        Suspendisse venenatis magna sed ornare pharetra.
        Fusce nec ipsum ultrices, placerat libero ac, varius orci.
        Nullam sollicitudin at ligula ut lacinia.
        Sed eu porttitor mi.
        Vestibulum placerat turpis non nisi aliquam cursus.
        Vestibulum a arcu a tortor viverra rhoncus.
        Nulla sagittis magna id eros congue, id dignissim odio semper.</p>
      <ul class="list-unstyled">
        <li>Bootstrap 4.3.1</li>
        <li>jQuery 3.4.1</li>
      </ul>
    </div>
  </div>
```

▷ No seu lugar, cria um `bloco` para poder inserir conteúdo de html provindo de outros ficheiros html que estão no projeto:

templates > base.html > html > body > div.container > div.row > div.col-lg-12.text-cent

```
49    <div class="container">
50      <div class="row">
51        <div class="col-lg-12 text-center">
52          {%block 'conteudo'%}{%endblock%}
53        </div>
54      </div>
55    </div>
56
57    <!-- Bootstrap core JavaScript -->
58    <script src="vendor/jquery/jquery.slim.min.js"></script>
59    <script src="vendor/bootstrap/js/bootstrap.bundle.min.js"></script>
60
61  </body>
```

▷ No ficheiro **sobre.html**, da pasta 'paginas', da app 'paginas', coloca o seguinte código:

Settings | urls.py f9 3 | urls.py paginas 1 | views.py 1

paginas > templates > paginas > sobre.html

```
1  {%extends 'base.html'%}
2  {% block 'conteudo'%}SOBRE{%endblock%}
```

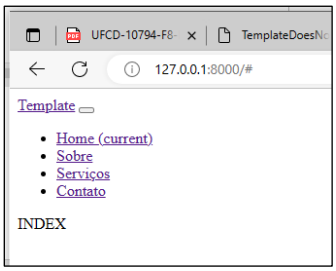
▷ No ficheiro **index.html**, da pasta 'paginas', da app 'paginas', coloca o seguinte código:

Settings | urls.py f9 3 | urls.py paginas 1 | views.py 1

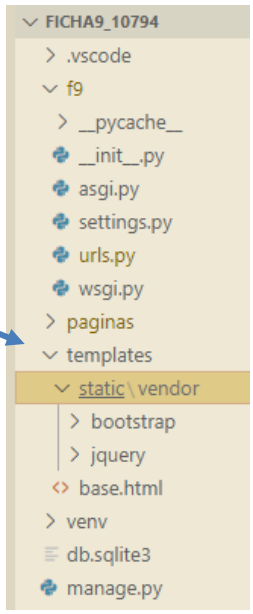
paginas > templates > paginas > index.html

```
1  {%extends 'base.html'%}
2  {% block 'conteudo'%}INDEX{%endblock%}
```

▷ Executa o servidor (deverás obter a página seguinte):



o CSS não foi carregado na página devido ao caminho não estar correto!



Alteração do código para que o css fique ativo:

Criar uma pasta de nome '**static**' na pasta '**templates**', criada na raiz do projeto. O Django irá procurar aí os ficheiros estáticos.

Dentro da pasta 'static' cria outra pasta de nome 'paginas' e coloca dentro dela a pasta 'vender', que contém os ficheiros bootstrap e jquery necessários ao projeto.

▷No ficheiro'base.html', procede às seguintes alterações:

- 1º ativar os elementos estáticos, da pasta static com a instrução: {% load static %}
- 2.ºMudar o endereço dos ficheiros css e jquery (uso de tags Django);

templates > base.html > html > head

```
1   {% load static %}
2   <!DOCTYPE html>
3   <html lang="pt">
4
5   <head>
6
7     <meta charset="utf-8">
8     <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">
9     <meta name="description" content="">
10    <meta name="author" content="">
11
12    <title>Template Bootstrap</title>
13
14    <!-- Bootstrap core CSS -->
15    <link href="{% static 'vendor/bootstrap/css/bootstrap.min.css'%}" rel="stylesheet">
16
17  </head>
18
19  <body>
```

```
    <div class="container">
      <div class="row">
        <div class="col-lg-12 text-center">
          {%block 'conteudo'%}{%endblock%}
        </div>
      </div>
    </div>

    <!-- Bootstrap core JavaScript -->
    <script src="{% static 'vendor/jquery/jquery.slim.min.js'%}"></script>
    <script src="{% static 'vendor/bootstrap/js/bootstrap.bundle.min.js'%}"></script>

  </body>
  </html>
```

▷ Executa o servidor (>> python mange.py runserver)

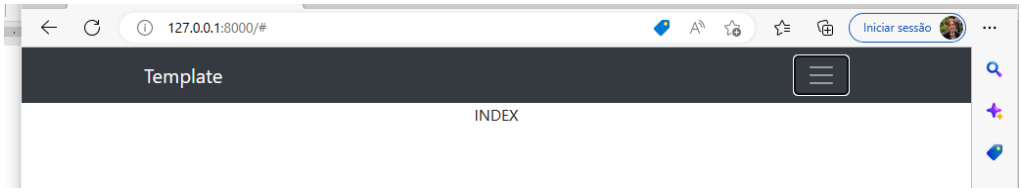
Ainda não deves conseguir visualizar o CSS, porque é necessário indicar onde o Django deve procurar os ficheiros estáticos!

Abre, para isso, o ficheiro settings.py do projeto. No código, após a variável STATIC_URL indica o caminho para o diretório dos ficheiros estáticos:

```
# Static files (CSS, JavaScript, Images)
# https://docs.djangoproject.com/en/4.1/howto/static-files/

STATIC_URL = "static/"
STATICFILES_DIRS=[BASE_DIR/ 'templates/static']
```

▷ Executa o servidor para visualizares o resultado



▷ Apaga as tag relativas às listas (
) com os link de 'contactos' e 'serviços' da página **base.html**:

```
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#">Sobre</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#">Serviços</a>
            </li>
            <li class="nav-item">
              <a class="nav-link" href="#">Contato</a>
            </li>
          </ul>
        </div>
      </div>
    </nav>
```

▷ Configurar o caminho dos links '*home*' e '*sobre*' para apontarem para os ficheiros respetivos (index.html e sobre.html):

```
28        <div class="collapse navbar-collapse" id="navbarResponsive">
29          <ul class="navbar-nav ml-auto">
30            <li class="nav-item active">
31              <a class="nav-link" href="#">Home
32                <span class="sr-only">(current)</span>
33              </a>
34            </li>
35            <li class="nav-item">
36              <a class="nav-link" href="#">Sobre</a>
37            </li>
38
39          </ul>
```

```
<li class="nav-item active">
  <a class="nav-link" href="/">Home
    <span class="sr-only">(current)</span>
  </a>
</li>
<li class="nav-item">
  <a class="nav-link" href="/sobre">Sobre</a>
</li>
```

▷ Executa o servidor e observa o resultado

- Poderás alternar entre as páginas de html index e sobre clicando nos link da barra de menus do template, **Home** e **Sobre**.

  Poder-se-ia usar uma tag Django para o efeito:
  Em vez de "/" e "/sobre" poderíamos ter escrito

  "{%url 'index'%}" e "{%url 'sobre'%}"

- Substitui um dos caminhos por uma das tag Django referidas a cima. Deverá ser este o código a adotar no futuro, para referir nos link de ficheiros html.

- Inserir uma imagem
  - Copia o ficheiro de imagem 'a.jpg' para dentro da pasta 'static' do projeto (criada na pasta 'templates 'que se encontra na raiz do projeto).

  - Abre o ficheiro sobre.html e adiciona o seguinte código:

Settings | base.html | settings.py | sobre.html ● | a.jpg

paginas > templates > paginas > sobre.html > ...

```
1  {%extends 'base.html'%}
2  {%load static %}
3  {%block 'conteudo'%}
4
5  <h1>SOBRE</h1>
6  <img src="{% static 'a.jpg/' %}"style="max-width: 350px; height: auto;>
7
8  {%endblock%}
```

A tag Django extends deve ser a 1.ª a escrever nos templates 'filhos'

- Executa o servidor e observa o resultado: ao clicares no link 'Sobre' a página é exibida com a imagem.

- Exibir data em português.
  - Em settings.py altera a língua para poutuguês:

```
LANGUAGE_CODE = "pt-PT"
```

  - Escreve no ficheiro index.html o código:

ettings.py | urls.py ficha8 | views.py home | __ir

home > templates > home > index.html > ...

```
1   {%extends 'base.html'%}
2   {%block 'título'%}HOME{%endblock%}
3   {%block 'conteúdo'%}>
4   <h1>eu sou o index!</h1>
5   <h3> hoje é: {%now "D d b y H:i"%}</h3>
6   {%endblock%}
```

- Executa o servidor e observa o resultado

Bibliografia

https://www.w3schools.com/django/index.php
https://stackoverflow.com/questions/63259560/extends-base-html-doesnt-loaded-in-django
https://www.w3schools.com/django/django_master_template.php
https://docs.djangoproject.com/en/4.1/howto/static-files/
Curso de Python 3 do básico ou Avançado – Udemy, -de Luiz Miranda
https://docs.djangoproject.com/en/4.1/ref/settings/
Tags e filtros de template embutidos (Built-in) — Django 1.0 documentation (django-portuguese.readthedocs.io)
https://www.javatpoint.com/django-tutorial
https://docs.python.org/3.12/library/datetime.html?highlight=datetime#strftime-and-strptime-format-codes