

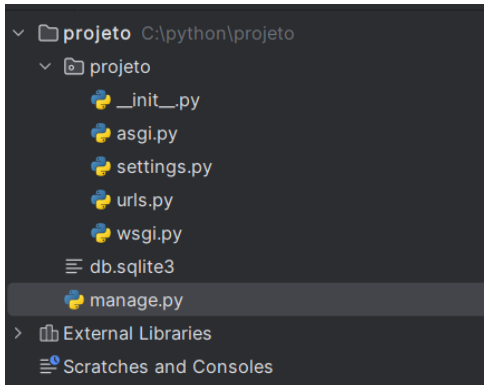
Elias Santos

Aula: Conhecendo pouco a pouco os recursos

Estamos nesta aula, continuando o conteúdo desenvolvido na aula anterior:

⇒ ambiente virtual: senac

⇒ nome do projeto: projeto



Embora estejamos usando a estrutura do MVT, vamos dar uma pequena olhada nos arquivos que mais a frente estaremos manipulando

1) movimentação no urls

No exemplo abaixo, estamos criando views (visualizações de página) com request (requisição), junto com a chamada (response) e os endereços específicos para o navegador, dentro do arquivo urls.py da pasta principal do projeto:

```
from django.contrib import admin
from django.urls import path
# a biblioteca abaixo chama pelo método HttpResponse
from django.http import HttpResponse

#funções para carregar views diretamente
def home(request):
    lista = '<ol><li>Saturnino</li><li>Eduardo</li><li>Kaua</li></ol>'
    return HttpResponse(lista)

def blog(request):
    print('Passei pelo blog')
    return HttpResponse('<body bgcolor="blue">Alô Mundo!</body>')

def contato(request):
    pagina = '<body bgcolor="#FF8DC"><h1><center>Elias Santos</center></h1><p>Email: eliasantos.senac@gmailcom<p>Telefone: 2667-1198'
    return HttpResponse(pagina)

urlpatterns = [
    path('', home),
    path('home/', home),
    path('blog/', blog),
    path('contato/', contato),
]
```

Após executar o `python manage.py runserver`, você pode ver o conteúdo das funções no navegador, inserindo após o endereço local a opção `/home` ou `/blog` ou `/contato`



Elias Santos

Aula: Conhecendo pouco a pouco os recursos

2) criação de apps

Vamos aprimorar nosso desenvolvimento com a criação de apps.

Será um app para cada página:

Sintaxe: python manage.py startapp nome_do_app

Vamos então criar um app para as 3 páginas utilizadas:

```
>> python manage.py startapp home  
>> python manage.py startapp contato  
>> python manage.py startapp blog
```

⇒ veja que foi criado uma pasta para cada um dos apps com vários arquivos dentro.

⇒ dentre estes arquivos, existe o arquivo views.py em cada app

⇒ transfira as funções para as views específicas dos apps, veja o modelo a seguir:

```
> blog > views.py > ...  
#from django.shortcuts import render  
from django.http import HttpResponseRedirect  
|  
# Create your views here.  
  
def blog(request):  
    print('Passei pelo blog')  
    return HttpResponseRedirect('<body bgcolor="blue">Alô Mundo!</body>')
```

```
projeto > contato > views.py > ...  
1  #from django.shortcuts import render  
2  from django.http import HttpResponseRedirect  
3  
4  # Create your views here.  
5  def contato(request):  
6      pagina = '<body bgcolor="#FFF8DC"><h1><center>Elias Santos</center></h1><p>Email: elias santos.senac@gmail.com<p>Telefone: 2667-1198'  
7      return HttpResponseRedirect(pagina)
```

```
projeto > home > views.py > ...  
1  #from django.shortcuts import render  
2  from django.http import HttpResponseRedirect  
3  
4  # Create your views here.  
5  def home(request):  
6      lista = '<ol><li>Saturnino</li><li>Eduardo</li><li>Kaua</li></ol>'  
7      return HttpResponseRedirect(lista)
```

Elias Santos

Aula: Conhecendo pouco a pouco os recursos

Agora iremos modificar o arquivo urls.py da pasta principal (projeto):

```
from django.contrib import admin
from django.urls import path
# importando as views dos apps
from blog import views as view_blog
from home import views as view_home
from contato import views as view_contato
💡
urlpatterns = [
    path('', view_home.home), #chamando as funções que estão na views
    path('home/', view_home.home),
    path('blog/', view_blog.blog),
    path('contato/', view_contato.contato),
]
```

Experimente e veja a aplicação rodando.

3) Aninhando URLs com path

a) Criamos o arquivo urls.py no app

b) Importação do include: `from django.urls import path, include`

As mudanças a seguir estão implementando arquivos html, e endereços aninhados.

Em cada app criamos uma pasta denominada templates, e dentro desta, outra pasta com o nome do app

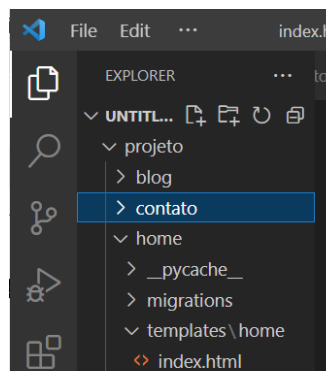
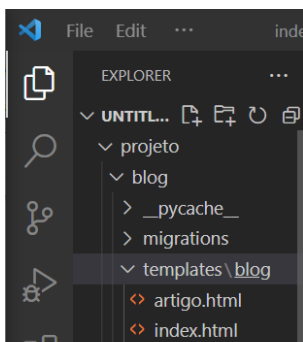
Exemplo:

blog

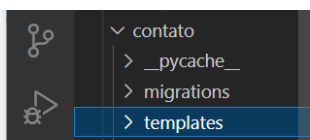
— templates

—blog

Veja a configuração das pastas:



Observe que dentro de blog >> templates >> blog há dois arquivos, a saber: index.html e artigo.html
E dentro de home >> templates >> home, o arquivo index.html

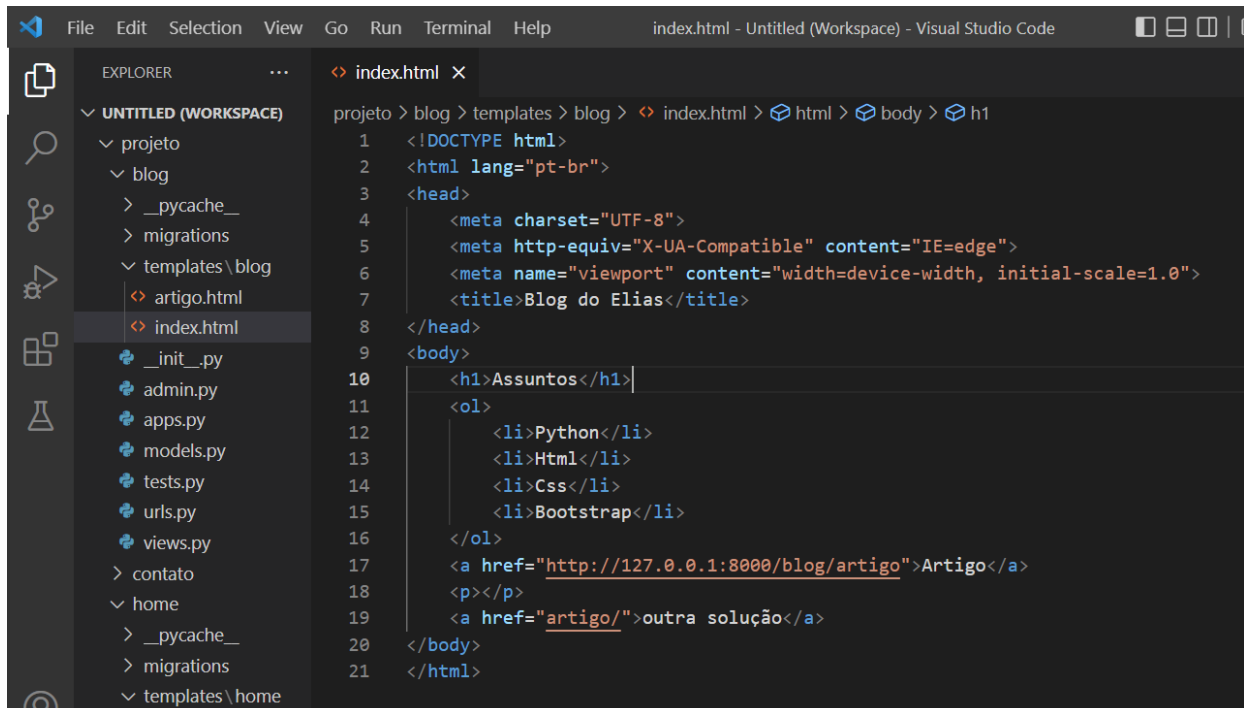


Elias Santos

Aula: Conhecendo pouco a pouco os recursos

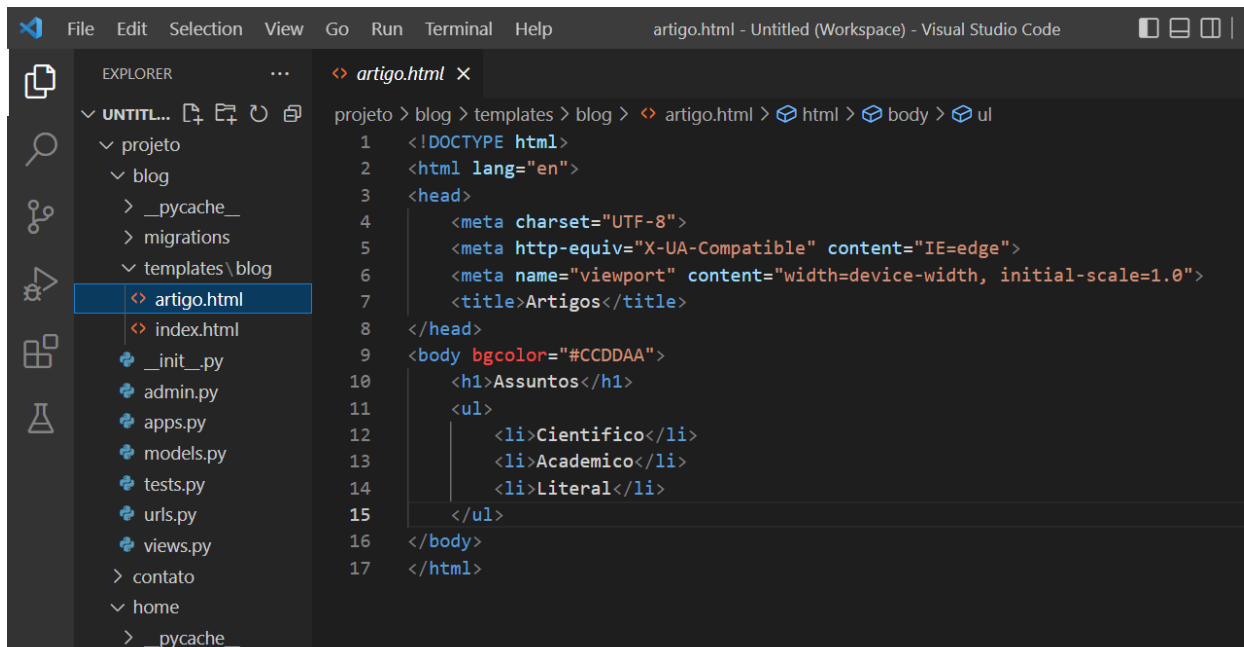
Segue o conteúdo destes arquivos:

Caminho: blog >> templates >> blog arquivo: index.html



```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Blog do Elias</title>
8 </head>
9 <body>
10  <h1>Assuntos</h1>
11  <ol>
12    <li>Python</li>
13    <li>Html</li>
14    <li>Css</li>
15    <li>Bootstrap</li>
16  </ol>
17  <a href="http://127.0.0.1:8000/blog/artigo">Artigo</a>
18  <p></p>
19  <a href="artigo/">outra solução</a>
20 </body>
21 </html>
```

Caminho: blog >> templates >> blog arquivo: artigo.html

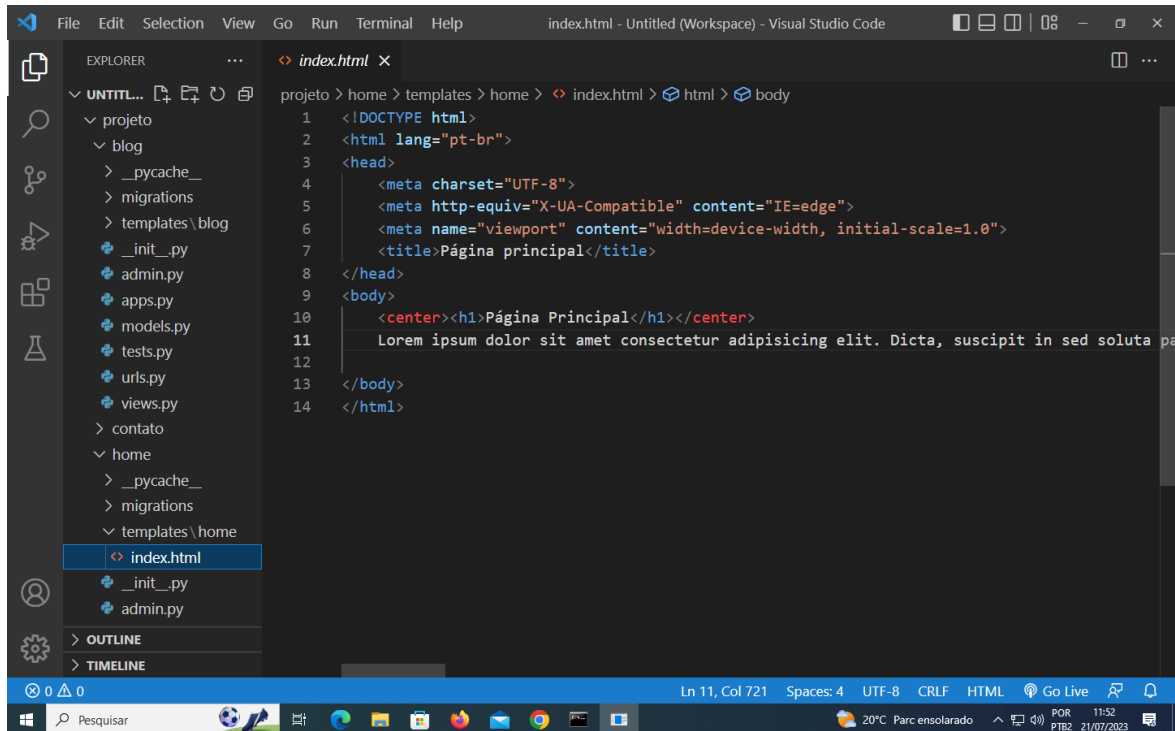


```
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Artigos</title>
8 </head>
9 <body bgcolor="#CCDDAA">
10  <h1>Assuntos</h1>
11  <ul>
12    <li>Cientifico</li>
13    <li>Academico</li>
14    <li>Literal</li>
15  </ul>
16 </body>
17 </html>
```

Elias Santos

Aula: Conhecendo pouco a pouco os recursos

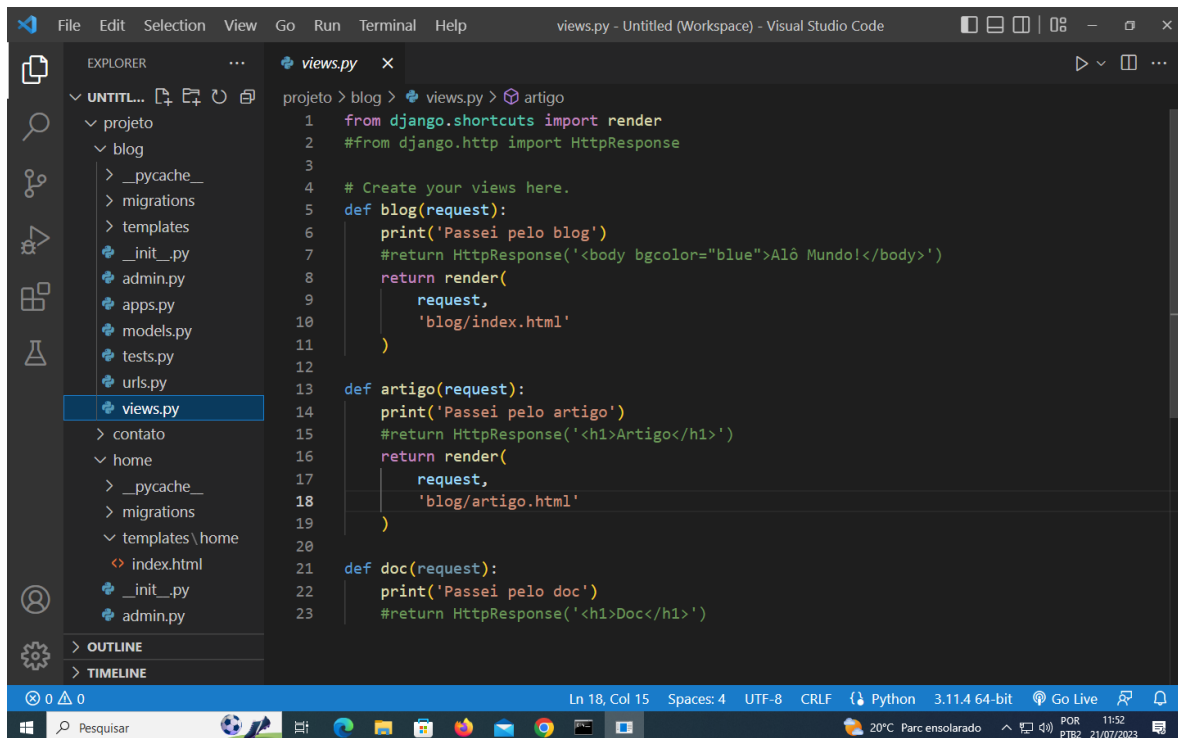
Caminho: home >> templates >> home arquivo: index.html



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The file explorer shows the project structure: projeto > blog > templates > home. The file index.html is selected. The main editor displays the content of index.html, which is an HTML template. The status bar at the bottom indicates the file is at line 11, column 721, with 4 spaces, UTF-8 encoding, and CRLF line endings.

```
1 <!DOCTYPE html>
2 <html lang="pt-br">
3 <head>
4   <meta charset="UTF-8">
5   <meta http-equiv="X-UA-Compatible" content="IE=edge">
6   <meta name="viewport" content="width=device-width, initial-scale=1.0">
7   <title>Página principal</title>
8 </head>
9 <body>
10   <center><h1>Página Principal</h1></center>
11   Lorem ipsum dolor sit amet consectetur adipisicing elit. Dicta, suscipit in sed soluta pa
12
13 </body>
14 </html>
```

Caminho: blog arquivo: views.py



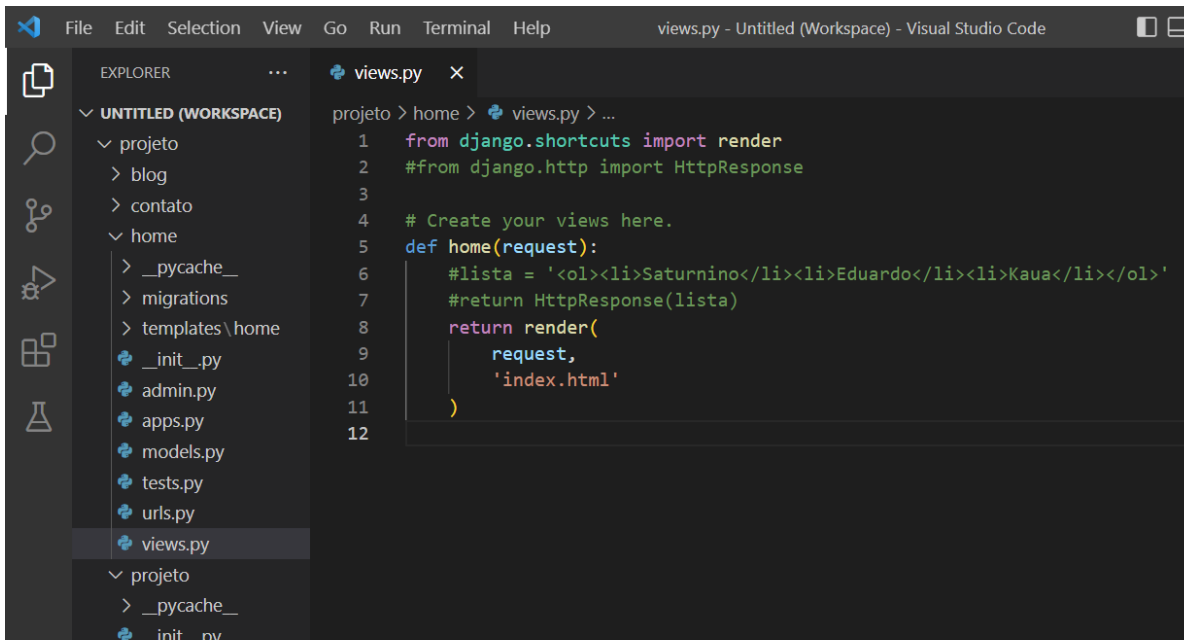
The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The file explorer shows the project structure: projeto > blog > views.py. The file views.py is selected. The main editor displays the content of views.py, which is a Python file for Django views. The status bar at the bottom indicates the file is at line 18, column 15, with 4 spaces, UTF-8 encoding, and CRLF line endings.

```
1 from django.shortcuts import render
2 #from django.http import HttpResponse
3
4 # Create your views here.
5 def blog(request):
6     print('Passei pelo blog')
7     #return HttpResponse('<body bgcolor="blue">Alô Mundo!</body>')
8     return render(
9         request,
10        'blog/index.html'
11    )
12
13 def artigo(request):
14     print('Passei pelo artigo')
15     #return HttpResponse('<h1>Artigo</h1>')
16     return render(
17         request,
18        'blog/artigo.html'
19    )
20
21 def doc(request):
22     print('Passei pelo doc')
23     #return HttpResponse('<h1>Doc</h1>')
```

Elias Santos

Aula: Conhecendo pouco a pouco os recursos

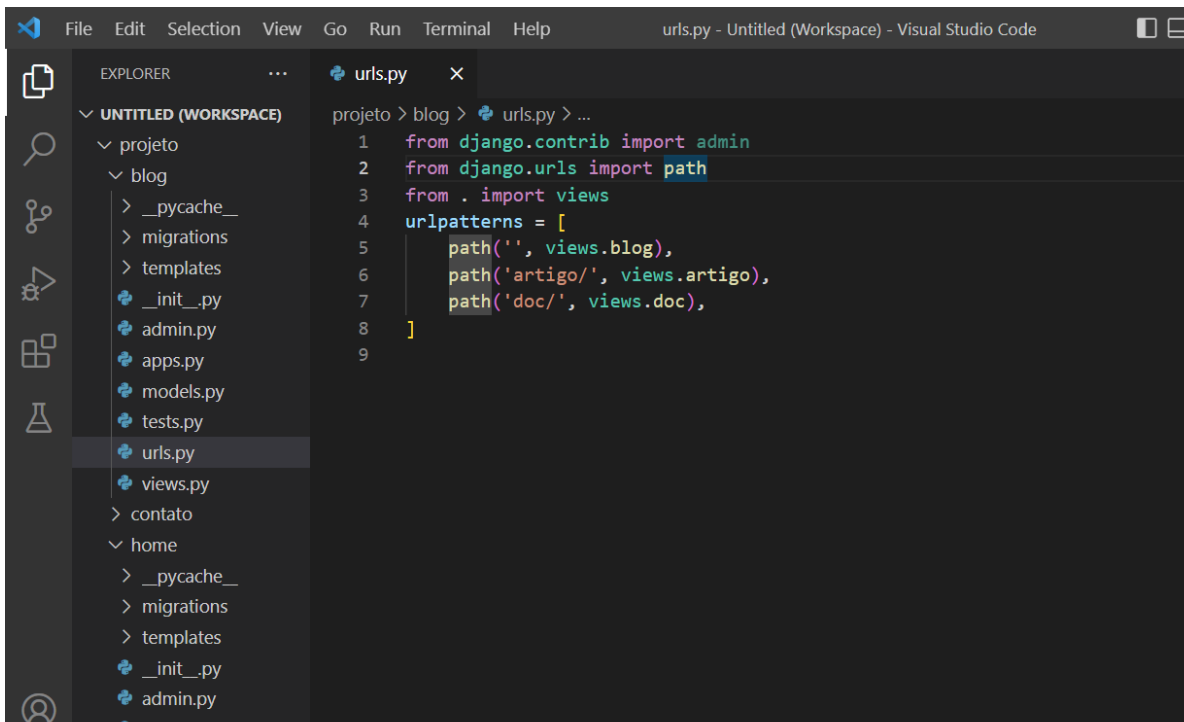
Caminho: home arquivo: views.py



The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The 'home' directory is expanded, showing files like __pycache__, migrations, templates, __init__.py, admin.py, apps.py, models.py, tests.py, urls.py, and views.py. The 'views.py' file is selected and its content is displayed in the main editor. The code defines a 'home' view function that uses 'render' from 'django.shortcuts' to return an 'HttpResponse' with a list of names: Saturnino, Eduardo, and Kaua.

```
1 from django.shortcuts import render
2 #from django.http import HttpResponse
3
4 # Create your views here.
5 def home(request):
6     #lista = '<ol><li>Saturnino</li><li>Eduardo</li><li>Kaua</li></ol>'
7     #return HttpResponse(lista)
8     return render(
9         request,
10        'index.html'
11    )
12
```

Caminho: blog arquivo: urls.py



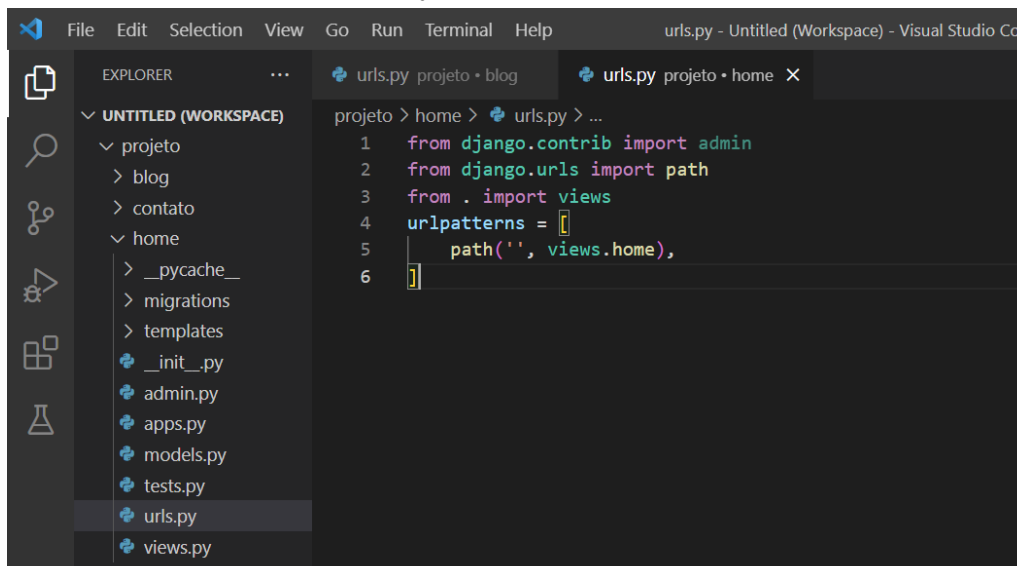
The screenshot shows the Visual Studio Code interface with the Explorer sidebar on the left. The 'blog' directory is expanded, showing files like __pycache__, migrations, templates, __init__.py, admin.py, apps.py, models.py, tests.py, urls.py, and views.py. The 'urls.py' file is selected and its content is displayed in the main editor. The code imports 'admin' from 'django.contrib', 'path' from 'django.urls', and 'views' from the current directory. It then defines a list of 'urlpatterns' with three entries: a root path for 'views.blog', a path for 'artigo/' for 'views.artigo', and a path for 'doc/' for 'views.doc'.

```
1 from django.contrib import admin
2 from django.urls import path
3 from . import views
4 urlpatterns = [
5     path('', views.blog),
6     path('artigo/', views.artigo),
7     path('doc/', views.doc),
8 ]
9
```

Elias Santos

Aula: Conhecendo pouco a pouco os recursos

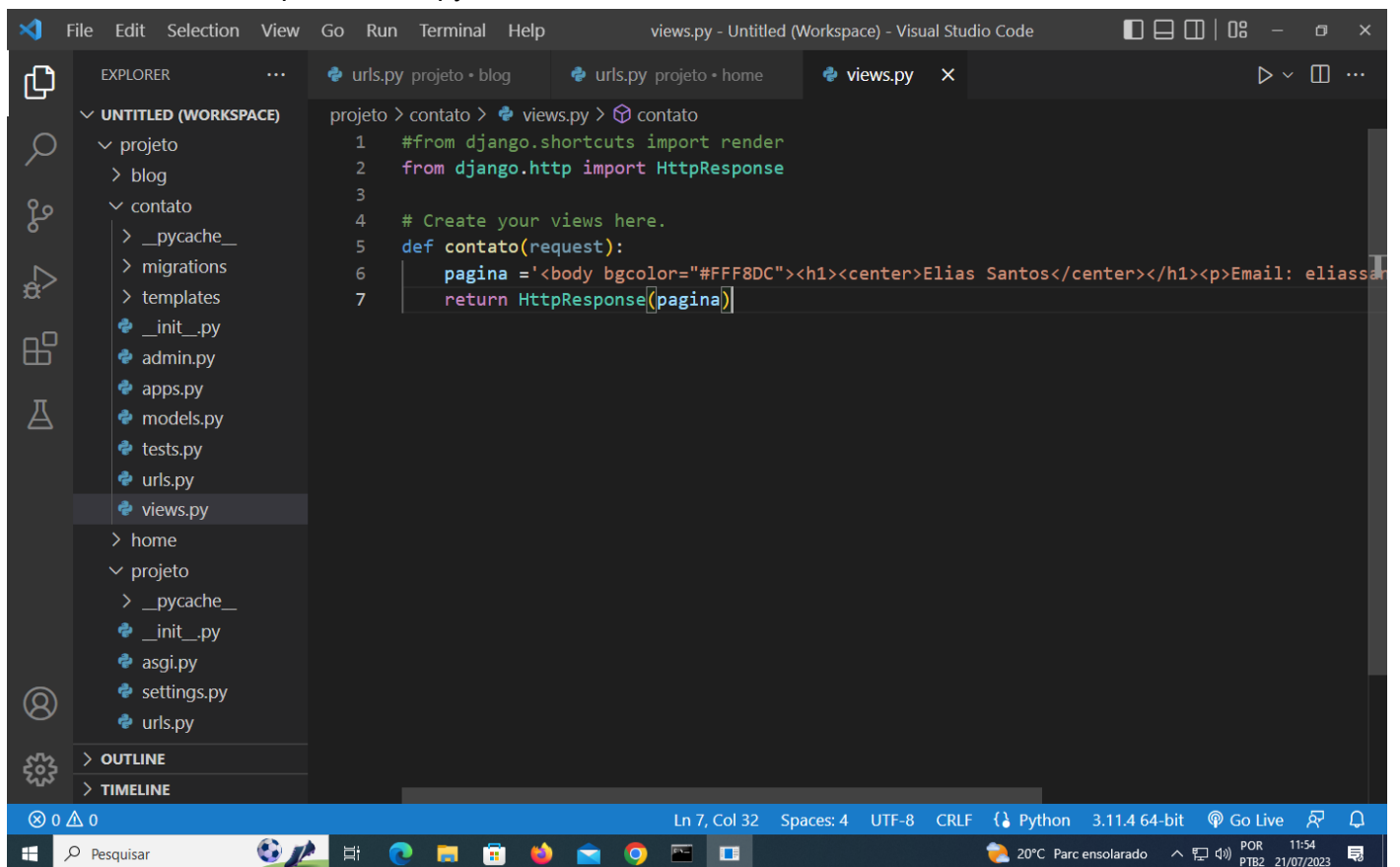
Caminho: home arquivo: urls.py



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the project structure under 'UNTITLED (WORKSPACE)'. The 'home' directory is expanded, showing files like __pycache__, migrations, templates, __init__.py, admin.py, apps.py, models.py, tests.py, urls.py (selected), and views.py. The main editor window shows the contents of 'urls.py' for the 'home' project. The code defines a list of URL patterns, including a path for the home view.

```
projeto > home > urls.py > ...  
1 from django.contrib import admin  
2 from django.urls import path  
3 from . import views  
4 urlpatterns = [  
5     path('', views.home),  
6 ]
```

Caminho: contato arquivo: views.py



The screenshot shows the Visual Studio Code interface. The Explorer sidebar on the left displays the project structure. The 'contato' directory is expanded, showing files like __pycache__, migrations, templates, __init__.py, admin.py, apps.py, models.py, tests.py, urls.py, and views.py (selected). The main editor window shows the contents of 'views.py' for the 'contato' project. The code defines a view function 'contato' that renders a response with HTML content.

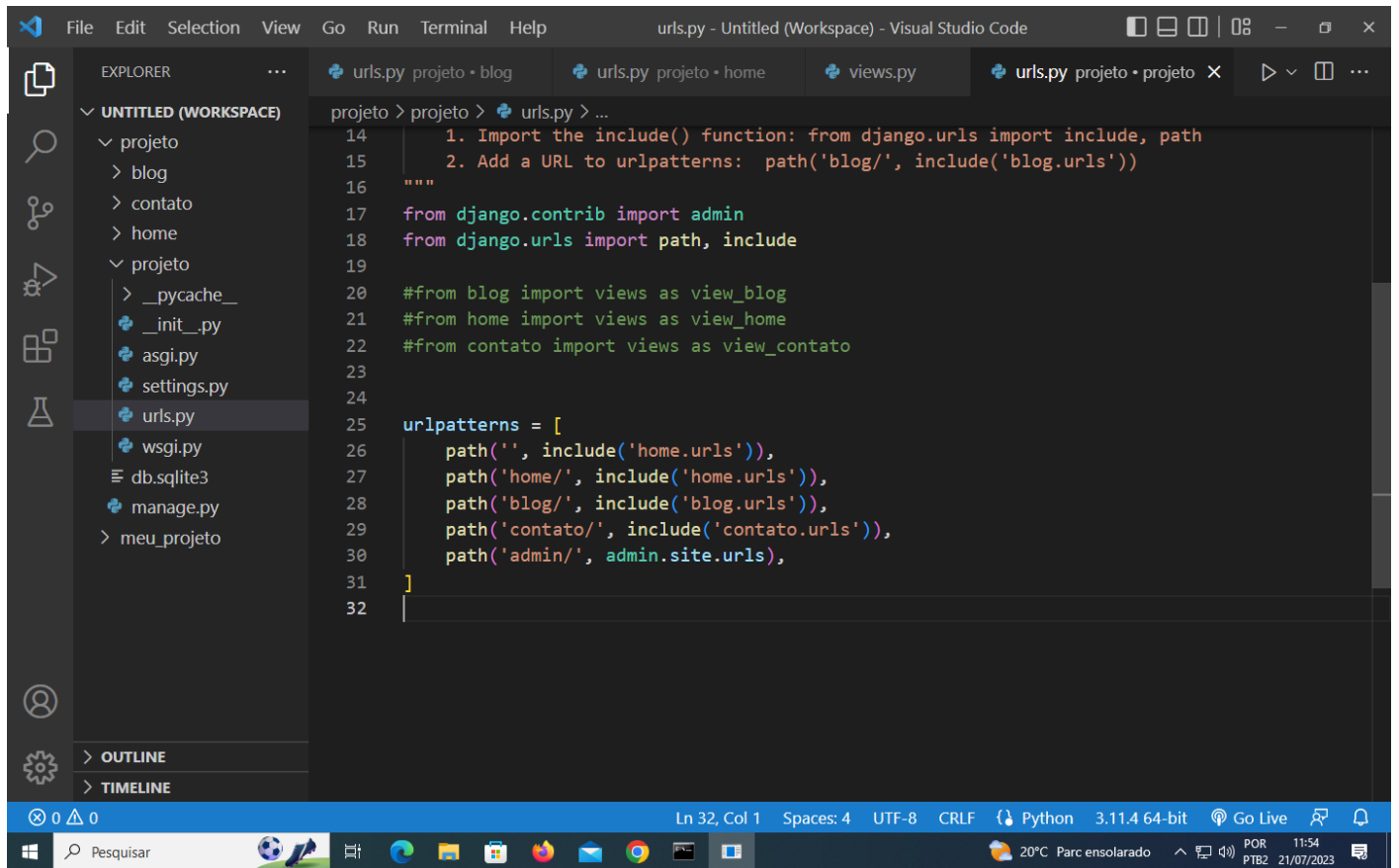
```
projeto > contato > views.py > contato  
1 #from django.shortcuts import render  
2 from django.http import HttpResponse  
3  
4 # Create your views here.  
5 def contato(request):  
6     pagina = '<body bgcolor="#FFF8DC"><h1><center>Elias Santos</center></h1><p>Email: eliasar  
7     return HttpResponse(pagina)
```

The status bar at the bottom indicates the current position (Ln 7, Col 32), encoding (UTF-8), line endings (CRLF), and the Python environment (Python 3.11.4 64-bit).

Elias Santos

Aula: Conhecendo pouco a pouco os recursos

Caminho: projeto arquivo: urls.py



The screenshot shows the Visual Studio Code interface with a Django project. The Explorer sidebar on the left displays the project structure under 'UNTITLED (WORKSPACE)'. The project is named 'projeto' and contains subdirectories 'blog', 'contato', 'home', and 'projeto'. The 'projeto' directory contains files: '__pycache__', '__init__.py', 'asgi.py', 'settings.py', 'urls.py' (selected), 'wsgi.py', 'db.sqlite3', 'manage.py', and 'meu_projeto'. The main editor window shows the content of 'urls.py' with the following code:

```
14     1. Import the include() function: from django.urls import include, path
15     2. Add a URL to urlpatterns: path('blog/', include('blog.urls'))
16     """
17     from django.contrib import admin
18     from django.urls import path, include
19
20     #from blog import views as view_blog
21     #from home import views as view_home
22     #from contato import views as view_contato
23
24
25     urlpatterns = [
26         path('', include('home.urls')),
27         path('home/', include('home.urls')),
28         path('blog/', include('blog.urls')),
29         path('contato/', include('contato.urls')),
30         path('admin/', admin.site.urls),
31     ]
32
```

The status bar at the bottom indicates the current file is 'urls.py' at line 32, column 1. It also shows the Python version (3.11.4 64-bit) and the Go Live extension. The system tray at the bottom right shows the temperature (20°C), location (Parc ensolarado), and the date and time (21/07/2023, 11:54).