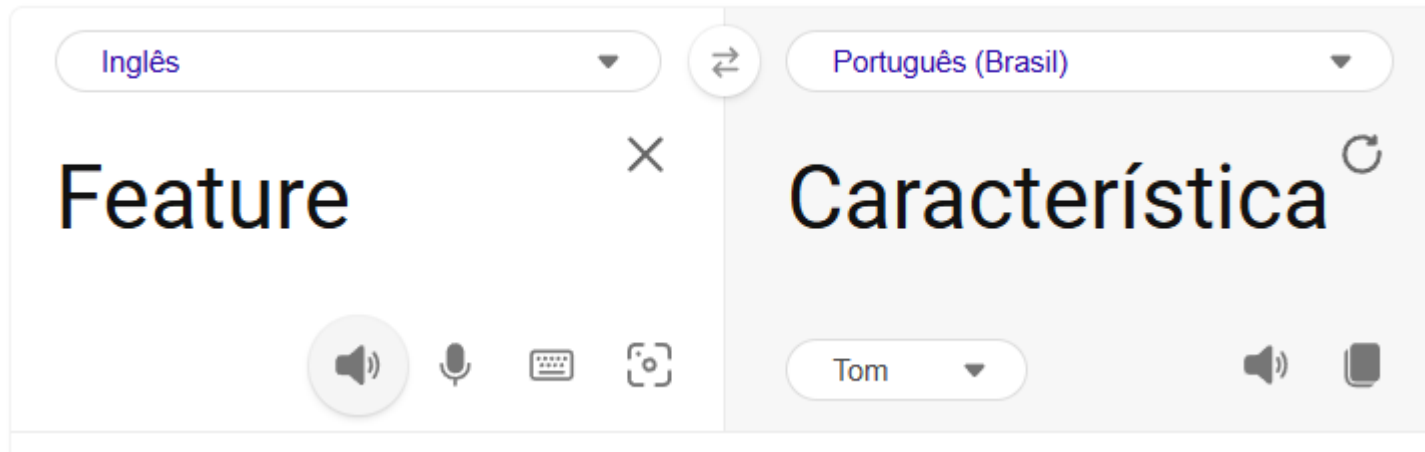


Buscar

# Feature



Uma nova característica (funcionalidade).

# Buscar

## Funcionalidade de busca

Na página principal, no nosso **"index.html"**,  
percebemos haver uma barra de busca no canto superior direito.

```
<div class="cabecalho_busca">
  <div class="busca_fundo">
    <form action="{% url 'buscar' %}">
      <input class="busca_input" type="text" name="buscar" placeholder="O que você procura?">
      <button type="submit">
        
      </button>
    </form>
  </div>
</div>
```

Contains a URL or a URL fragment that the hyperlink points to. A URL fragment is a name preceded by a hash mark ( # ), which specifies an internal target location (an `id` of an HTML element) within the current document. URLs are not restricted to Web

```
<div class="pagina-inicial">
  <header class="cabecalho">
    
    <div class="cabecalho_busca">
      <div class="busca_fundo">
        <input class="busca_input" type="text" placeholder="O que você procura?">
        
      </div>
    </div>
  </header>
  <main class="principal">
    <section class="menu-lateral">
      <nav class="menu-lateral navegacao">
```

**Nessa barra**, podemos escrever algo e, em teoria, deveria ser possível clicar no **ícone de lupa** para fazer a busca.

No entanto, isso não acontece, esse mecanismo não funciona.

O que você procura?



Então **faremos ele** funcionar!



Onde eu configuro as urls:

Vamos ao nosso código para começar a criar uma rota, uma nova URL. Quando falamos em URL, em qual arquivo devemos

ir? No "urls.py" da **nossa galeria** da nossa aplicação.

Queremos passar um **novo path chamado "buscar"**.  
Definimos que ele irá apontar para uma rota que  
criaremos no "views.py", também denominada "buscar".  
Como referência (name), também colocamos "buscar".

```
from django.urls import path
from galeria.views import index, imagen

urlpatterns = [
    path('', index, name='index'),
    path('imagen/<int:foto_id>', imagen,
name='imagen'),
    path("buscar", buscar, name="buscar"),
]
```

1. A segunda coisa que precisamos fazer é ir ao "views.py"
2. E criar essa rota que construímos.
3. Nós já temos index e imagem; agora, vamos criar buscar.
4. Na sequência, temos que receber o request e definir o que será retornado (return).
5. Vamos fazer o render, dentro do qual temos que passar o request novamente.

6. Após o request, o segundo parâmetro que precisamos passar é a página HTML que deve ser acessada.
7. Nós ainda não criamos uma, mas suponha que vamos criar uma chamada "buscar.html".

```
14 def buscar(request):  
15     return render(request, "buscar.html")
```

views.py M

urls.py M X

galeria > urls.py > ...

```
1  from django.urls import path
2  from galeria.views import index
3  from galeria.views import imagen, buscar ←
4
5  urlpatterns = [
6      path('', index, name='index'),
7      path('imagen/<int:foto_id>', imagen, name='imagen'),
8      path("buscar", buscar, name="buscar"),
9  ]
10
11
```

- Porém, ainda não criamos a página "buscar.html". Vamos criar então, em "**templates\galeria**", um arquivo chamado "**buscar.html**".
- Para visualizar no site, vamos colocar o seguinte trecho de código no arquivo:

▼ PLANETFINAL

- estudos
- galeria
- media
- setup
- static
- templates\galeria
  - partials
  - base.html
  - buscar.html U
  - imagem.html
  - index.html

templates > galeria > buscar.html

1



Conseguimos criar a rota de buscar, mas ela ainda está crua, pois colocamos o `<h1>` apenas para visualizar o processo. Vamos alterar o nosso "buscar.html" para conseguir receber as informações que precisamos.

# <h1>Buscar</h1>

Nós queremos mostrar a intenção, ter uma página muito parecida com a index, mas queremos que, ao pesquisar por "galáxia", por exemplo, apareçam como resultados somente os que de fato são galáxias, ou possuem essa palavra-chave. Porém, o restante da página seria igual à index.

Para fazer isso, iremos copiar algumas questões da página "index.html". Inicialmente, vamos copiar o código inteiro ("Ctrl + C"), e colar ("Ctrl + V") no lugar de Buscar.

Uma vez criada a nossa página "buscar.html", vamos ver o resultado no site. Ao buscar por "galaxia", você verá que somos encaminhados para a página de buscar sem exibir nenhum item.

Isso acontece porque não estamos recebendo itens do banco de dados, então nenhum irá aparecer. Inclusive, colocamos um condicional para verificar se existe algum card sendo passado (linha de código 22).

```
{% if cards %}
```



```
<!-- Comentário HTML-->
```

```
{% for fotografia in cards %}
```

```
<li class="card">
```

```
  <a href="{% url 'imagem' fotografia.id %}">
```

```
    <!--
```

```
    {% endif %}
```

```
  </a>
```

```
  <span class="card__tag">Estrelas</span>
```


```
  <div class="card__info">
```

```
    <p class="card__titulo">{{ fotografia.nome }}</p>
```

```
    <div class="card__texto">
```

```
      <p class="card_descricao">{{ fotografia.legenda }}</p>
```

```
        
      </span>
    </div>
  </div>
</li>
```



```
{% endfor %}
{% else %}
<div class="imagem__texto">
  <p>Fotografias não encontradas</p>
</div>
{% endif %}
```

A primeira coisa que precisamos fazer é puxar todos os objetos que temos no nosso banco de dados.

```
def buscar(request):
```

```
    fotografias = Fotografia.objects.order_by("data_fotografia").filter(publicada=True)
```





Agora, temos que filtrar os itens que puxamos para conferir se existe a palavra "galáxia" colocada na URL. Como devemos buscar essa palavra para fazer o filtro?

Primeiramente, temos que usar uma condicional para conferir se estamos de fato colocando o termo "buscar", ou seja, se o termo está presente nas informações passadas para a URL. Essas informações estão em request.GET.

A segunda coisa que vamos colocar é o nome que precisa ser buscado, o qual está dentro de `request.GET`, que é a representação da nossa URL. Em seguida, vamos abrir colchetes e buscá-lo.

```
if "buscar" in request.GET:  
    nome_a_buscar = request.GET['buscar']  
  
return render (request, "galeria/buscar.html")
```

```
git clone https://github.com/romulodf-cesar/planetfinal.git
```

Dever de Casa...



https://www.deepl.com/pt-BR/translator



DeepL

Tradutor ▾



Entr



Traduzir texto

33 idiomas



Traduzir arquivos

.pdf, .docx, .pptx



DeepL Write

Assistente de escrita baseado em IA

Detectar idioma ▾



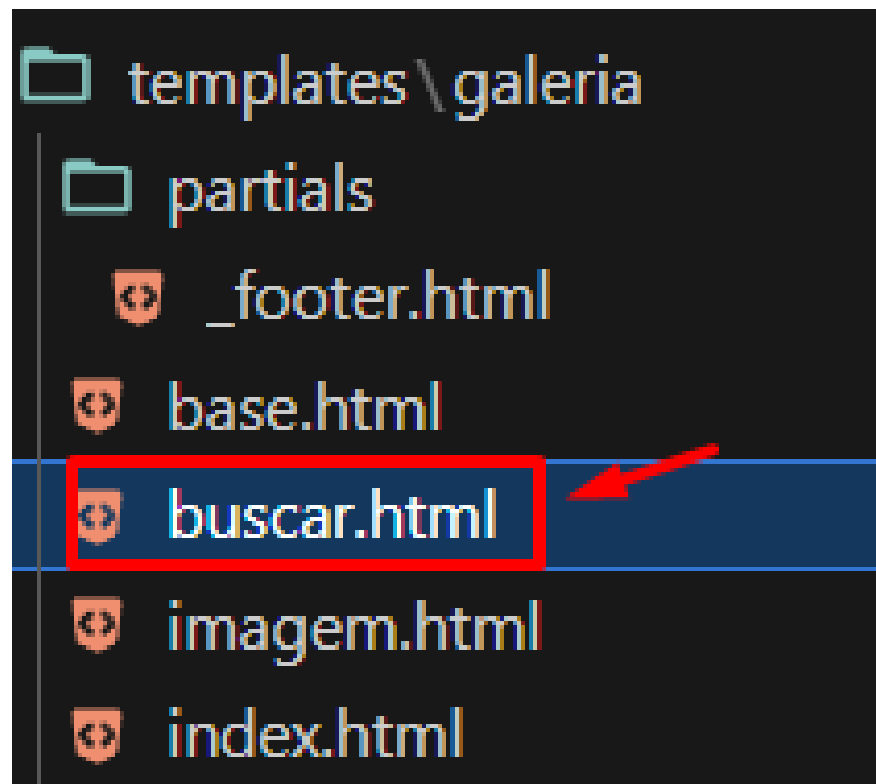
português ▾

Escreva ou cole o texto aqui.

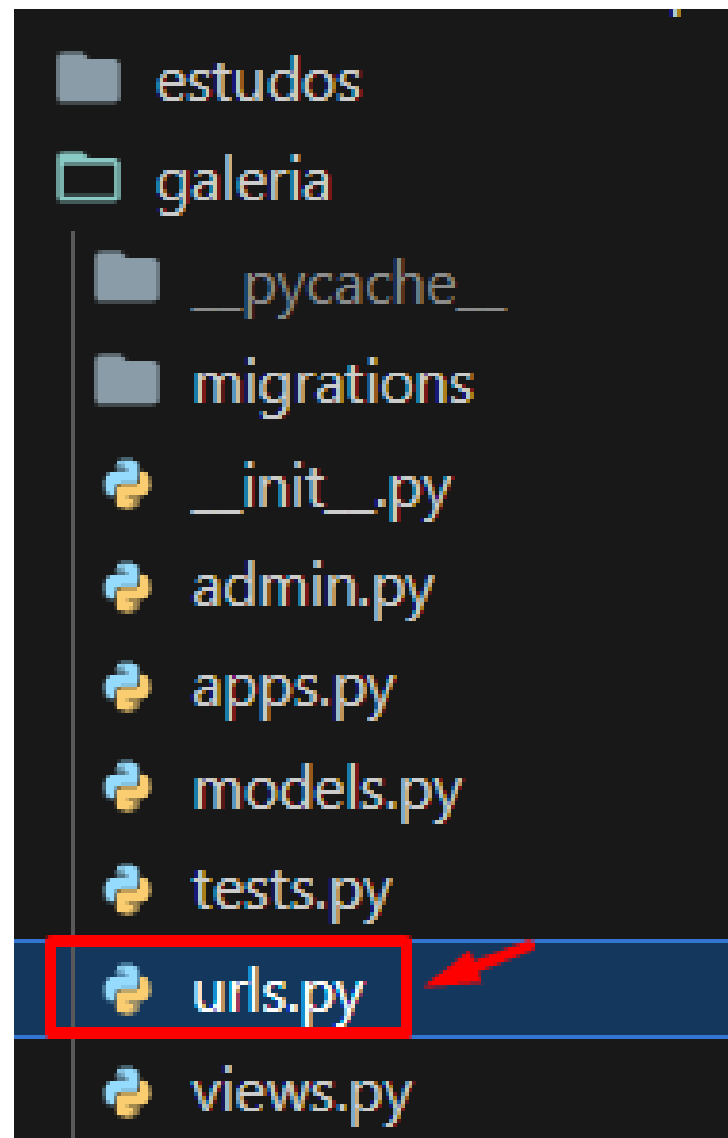
Arraste um documento PDF, Word (.docx) ou PowerPoint (.pptx) até aqui para traduzi-lo.

O que você procura?









```
1  from django.urls import path
2  from galeria.views import index
3  from galeria.views import imagen, buscar
4
5  urlpatterns = [
6      path('', index, name='index'),
7      path('imagen/<int:foto_id>', imagen, name='imagen'),
8      path("buscar", buscar, name="buscar"),
9  ]
10
11
```

```
<form action="{% url 'buscar' %}">
  <input class="busca__input" type="text" name="buscar" placeholder="O que
  <button type="submit">
    
    ←  
    <input class="busca__input" type="text"  
    <button type="submit">  
        <img class="busca__icone" src="{% st  
    </button>  
</form>
```