

# PROGRAMAÇÃO BACK-END

The Django logo, featuring the word "django" in a white, lowercase, sans-serif font with a small dot over the 'i', set against a dark green rectangular background.

The web framework for  
perfectionists with deadlines.

**Referência de pesquisas  
de campo QuerySet**

# PROGRAMAÇÃO BACK-END

## Referência de pesquisas de campo QuerySet

Em Django, um **QuerySet** é uma coleção de objetos de banco de dados que correspondem a um determinado conjunto de parâmetros de consulta.

Em outras palavras, é a maneira de Django de representar e trabalhar com consultas de banco de dados, permitindo **filtrar**, **ordenar** e **manipular** os **dados** de forma eficiente e abstrata.

# PROGRAMAÇÃO BACK-END

## Referência de pesquisas de campo QuerySet

As pesquisas de campo em um **QuerySet** referem-se ao uso de **campos específicos** de modelos em consultas ao banco de dados.

Django permite realizar operações complexas de pesquisa, como filtrar registros por valores de campo, ordenar resultados por campos específicos, e até mesmo realizar **consultas** que **abrangem relacionamentos** entre diferentes **modelos**.

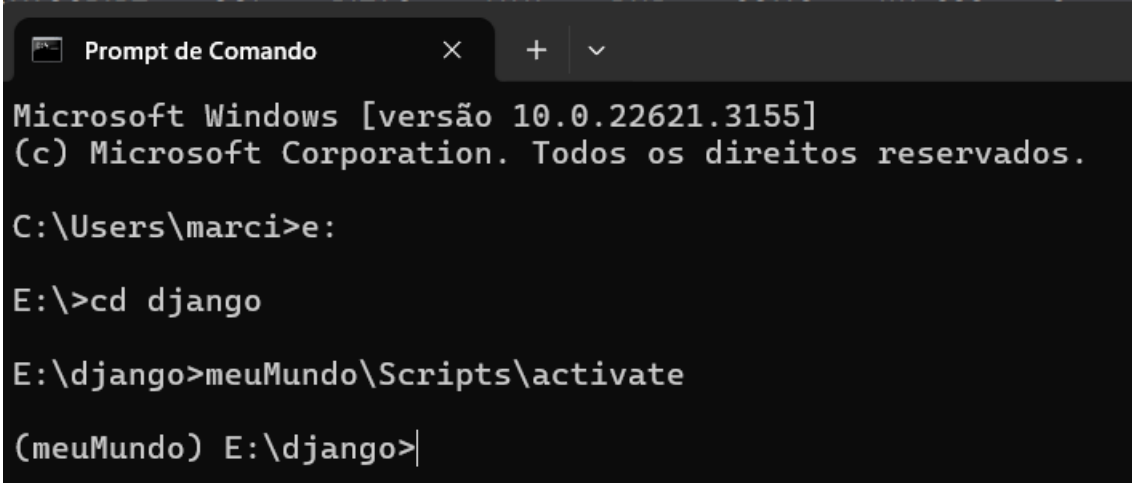
# PROGRAMAÇÃO BACK-END

---

**Preparando para os testes**

# PROGRAMAÇÃO BACK-END

## Rodar o script activate.bat



```
Prompt de Comando
Microsoft Windows [versão 10.0.22621.3155]
(c) Microsoft Corporation. Todos os direitos reservados.

C:\Users\marci>e:

E:\>cd django

E:\django>meuMundo\Scripts\activate

(meuMundo) E:\django>
```

**Este passo é importante para configurar as variáveis de ambiente.**


# PROGRAMAÇÃO BACK-END

---

**Iniciar Servidor Postgresql**

# PROGRAMAÇÃO BACK-END

Na pasta `../pgsql` execute o arquivo **PostgreSQL-Start.bat** para iniciar o servidor

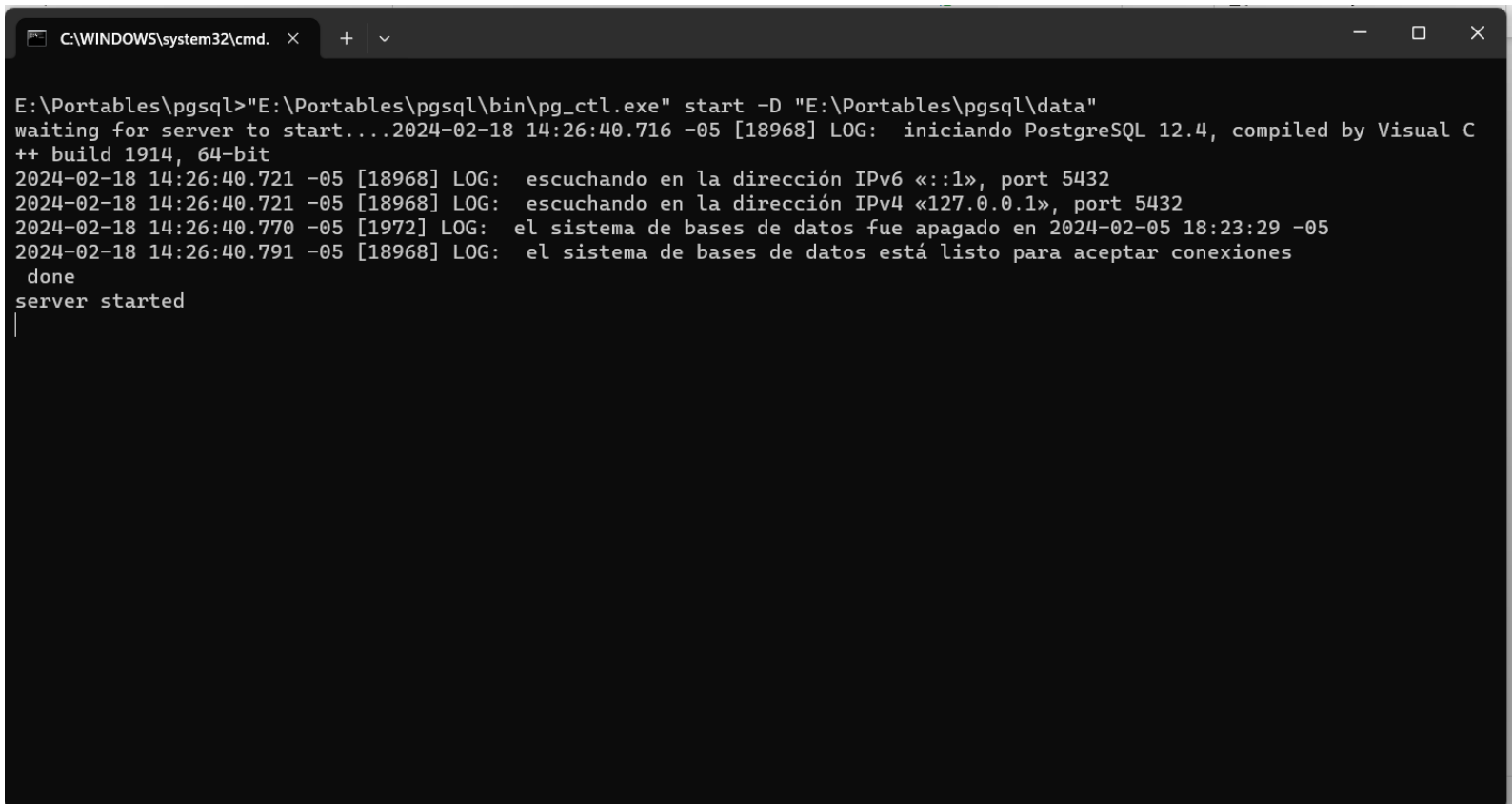


Nome	Data de modificação	Tipo	Tamanho
bin	11/08/2020 08:41	Pasta de arquivos	
data	18/02/2024 16:26	Pasta de arquivos	
doc	11/08/2020 08:41	Pasta de arquivos	
include	11/08/2020 08:41	Pasta de arquivos	
lib	11/08/2020 08:41	Pasta de arquivos	
pgAdmin 4	11/08/2020 08:41	Pasta de arquivos	
share	11/08/2020 08:41	Pasta de arquivos	
StackBuilder	11/08/2020 08:41	Pasta de arquivos	
Make Cluster.bat	09/01/2017 14:12	Arquivo em Lotes ...	1 KB
PgAdmin4.bat	28/04/2017 22:29	Arquivo em Lotes ...	1 KB
PostgreSQL-Restart.bat	06/04/2012 09:42	Arquivo em Lotes ...	1 KB
PostgreSQL-Start.bat	28/04/2017 21:46	Arquivo em Lotes ...	1 KB
PostgreSQL-Stop.bat	06/04/2012 09:43	Arquivo em Lotes ...	1 KB
Service - Delete.bat	29/08/2020 23:25	Arquivo em Lotes ...	1 KB
Service - Maker.bat	29/08/2020 23:26	Arquivo em Lotes ...	1 KB



# PROGRAMAÇÃO BACK-END

Será aberta uma janela em Texto informando que o servidor iniciou (**server started**).  
Não feche esta janela



```
C:\WINDOWS\system32\cmd. x + v

E:\Portables\pgsql>"E:\Portables\pgsql\bin\pg_ctl.exe" start -D "E:\Portables\pgsql\data"
waiting for server to start...2024-02-18 14:26:40.716 -05 [18968] LOG:  iniciando PostgreSQL 12.4, compiled by Visual C
++ build 1914, 64-bit
2024-02-18 14:26:40.721 -05 [18968] LOG:  escuchando en la dirección IPv6 «::1», port 5432
2024-02-18 14:26:40.721 -05 [18968] LOG:  escuchando en la dirección IPv4 «127.0.0.1», port 5432
2024-02-18 14:26:40.770 -05 [1972] LOG:  el sistema de bases de datos fue apagado en 2024-02-05 18:23:29 -05
2024-02-18 14:26:40.791 -05 [18968] LOG:  el sistema de bases de datos está listo para aceptar conexiones
done
server started
|
```

## Templates

# PROGRAMAÇÃO BACK-END

Podemos utilizar o App **dvdrentalapp**. Na pasta **templates** crie o arquivo **list\_customers.html** e digite o seguinte conteúdo:

```
<> list_customers.html X  urls.py  views.py
my_ads > dvdrentalapp > templates > <> list_customers.html > html
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h1>Customers</h1>
6
7  <ul>
8      {% for x in listcustomer %}
9          <li>{{ x.first_name }} {{ x.last_name }}</li>
10         {% endfor %}
11 </ul>
12
13 </body>
14 </html>
```

# PROGRAMAÇÃO BACK-END

**Arquivos `views.py` e `urls.py`**

# PROGRAMAÇÃO BACK-END


Na pasta **dvdrentalapp** adicione o seguinte conteúdo no arquivo **urls.py** (linha 10)

```
list_customers.html  urls.py  views.py

my_ads > dvdrentalapp > urls.py > ...
1  from django.urls import path
2  from . import views
3
4  urlpatterns = [
5      path('customer/', views.customer, name='mycustomer'),
6      path('detalhes/<int:id>', views.detalhes, name='myDetalhe'),
7      path('edit_customer/<int:customer_id>', views.edit_customer, name='edit_customer'),
8      path('add_category/', views.add_category, name='add_category'),
9      path('categories/', views.list_categories, name='list_categories'),
10     path('listacustomer/', views.listacustomer, name='listacustomer'),
11     # Adicione outras URLs conforme necessário
12 ]
```

# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** adicione o seguinte conteúdo no arquivo **views.py** (linhas 8 a 14)



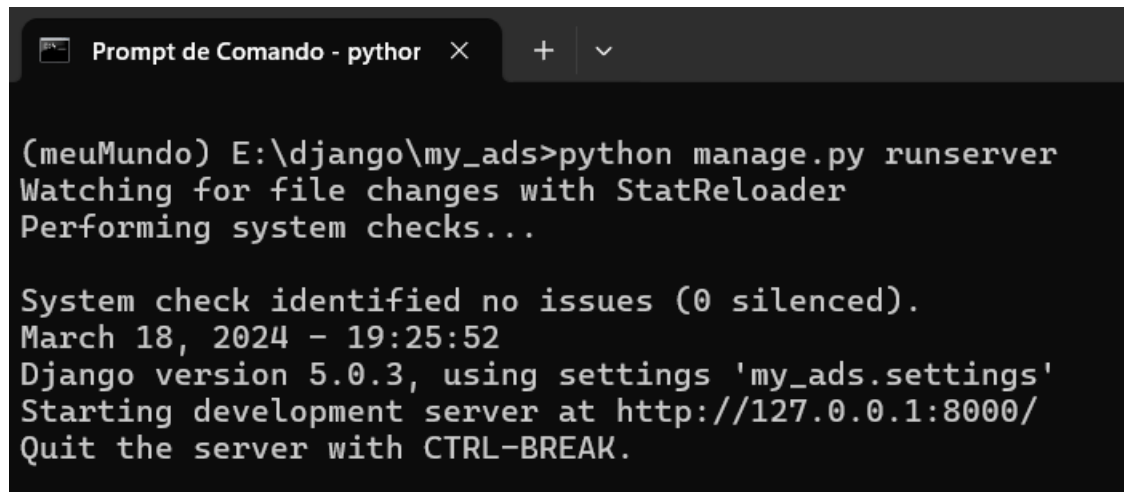
```
views.py  X  list_customers.html  urls.py
my_ads > dvdrentalapp > views.py > ...
1  # Create your views here.
2  from django.http import HttpResponse
3  from django.template import loader
4  from .models import Customer, Rental, Category
5  from django.utils import timezone
6  from django.shortcuts import render, get_object_or_404, redirect
7
8  def listcustomer(request):
9      mycustomers = Customer.objects.all().values()
10     template = loader.get_template('list_customers.html')
11     context = {
12         'listcustomer': mycustomers,
13     }
14     return HttpResponse(template.render(context, request))
15
```

**Iniciando o servidor  
e testando**

# PROGRAMAÇÃO BACK-END

**Execute na pasta my\_ads o comando:**

**python manage.py runserver**



```
Prompt de Comando - pythor  X  +  v

(meuMundo) E:\django\my_ads>python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
March 18, 2024 - 19:25:52
Django version 5.0.3, using settings 'my_ads.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

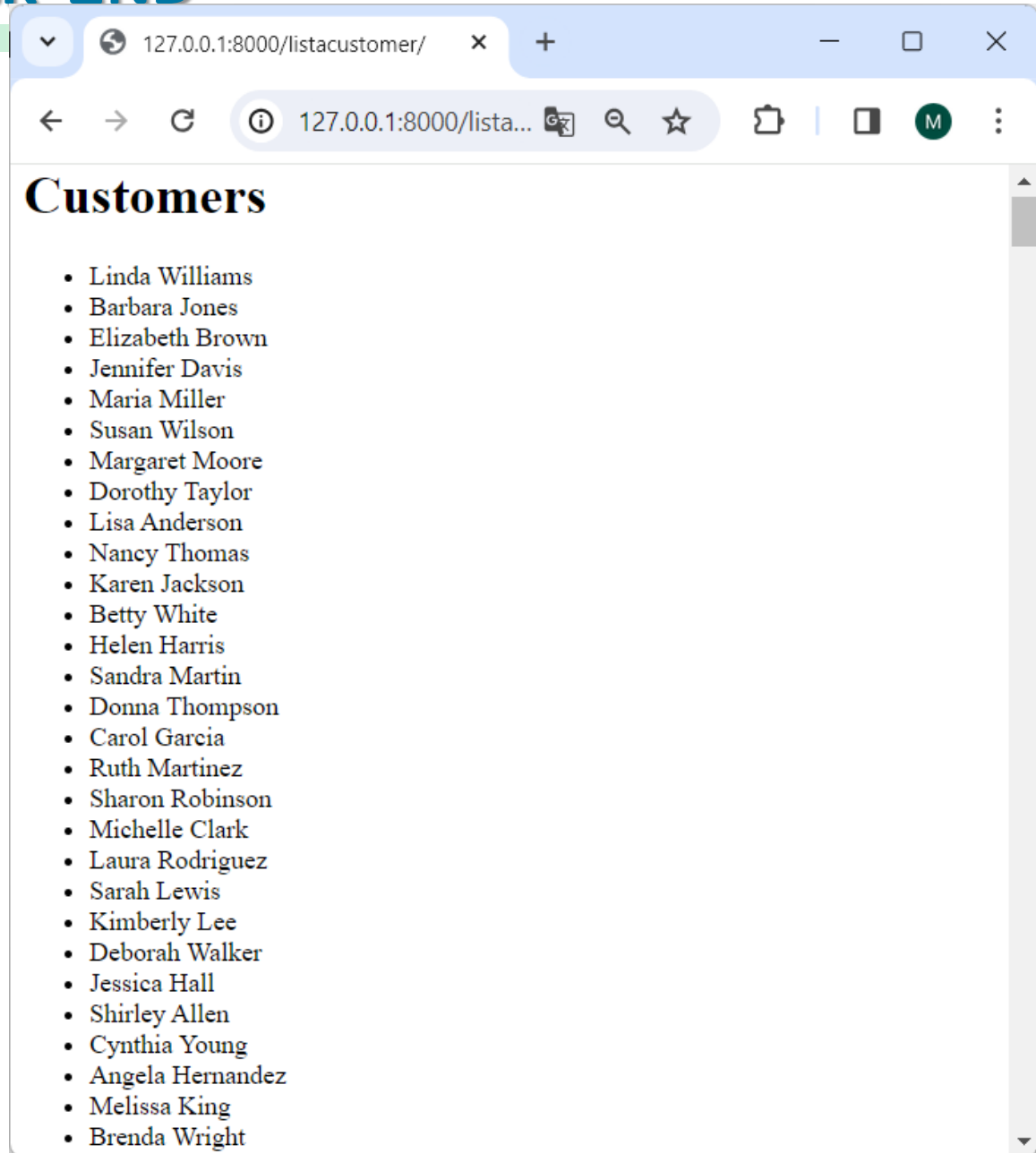
**Vá ao navegador e digite:**

**<http://127.0.0.1:8000/listacustomer/>**



# PROGRAMAÇÃO BACK-END

O resultado da página será a **listagem** de todos **customers**.



**Pesquisa de Campo  
contains**

# PROGRAMAÇÃO BACK-END

## Pesquisa de campo contains

### Definição e uso

A pesquisa **contains** é usada para obter registros que contenham um valor especificado.

Ela diferencia maiúsculas de minúsculas.

# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** altere o seguinte conteúdo no arquivo **views.py** (linha 9):



```
7
8  def listacustomer(request):
9      mycustomers = Customer.objects.filter(first_name__contains='Karen').values()
10     template = loader.get_template('list_customers.html')
11     context = {
12         'listcustomer': mycustomers,
13     }
14     return HttpResponse(template.render(context, request))
```

**Equivalente SQL**

**O SQL equivalente ao exemplo acima será:**

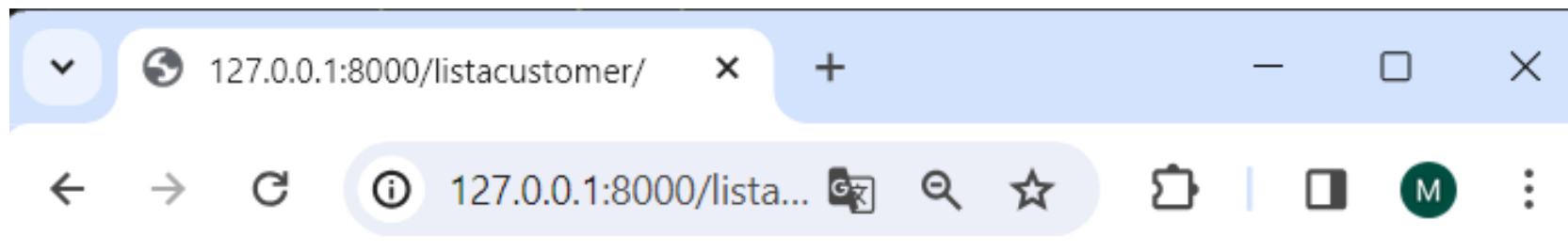
**WHERE first\_name LIKE 'Karem%';**

# PROGRAMAÇÃO BACK-END

Vá ao navegador e digite:

<http://127.0.0.1:8000/listacustomer/>

O resultado da página será a **listagem** a seguir.



## Customers

- Karen Jackson

**Pesquisa de Campo  
icontains**

# PROGRAMAÇÃO BACK-END

## Pesquisa de campo icontains


### Definição e uso

A pesquisa **icontains** é usada para obter registros que contenham um valor especificado.

Ela não diferencia maiúsculas de minúsculas.

# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** altere o seguinte conteúdo no arquivo **views.py** (linha 9):



```
8 def listcustomer(request):
9     mycustomers = Customer.objects.filter(last_name__icontains='Mill').values()
10    template = loader.get_template('list_customers.html')
11    context = {
12        'listcustomer': mycustomers,
13    }
14    return HttpResponse(template.render(context, request))
```

**Equivalente SQL**

**O SQL equivalente ao exemplo acima será:**

**WHERE last\_name LIKE '%Mill%';**

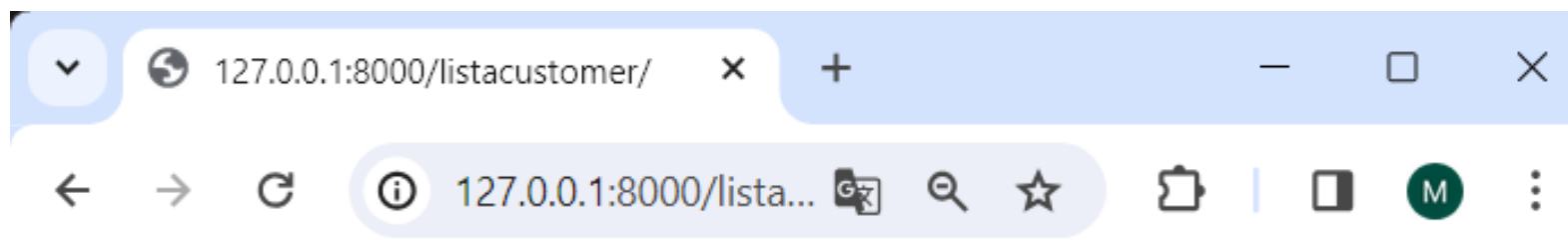


# PROGRAMAÇÃO BACK-END

Vá ao navegador e digite:

<http://127.0.0.1:8000/listacustomer/>

O resultado da página será a **listagem** a seguir.



## Customers

- Maria Miller
- Alicia Mills
- Shane Millard

**Pesquisa de Campo  
endswitch**

# PROGRAMAÇÃO BACK-END

## Pesquisa de campo endswith


### Definição e uso

A pesquisa **endswith** é usada para obter registros que terminam com um valor especificado.

A pesquisa **endswith** diferencia maiúsculas de minúsculas.

# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** altere o seguinte conteúdo no arquivo **views.py** (linha 9):



```
8 def listcustomer(request):
9     mycustomers = Customer.objects.filter(first_name__endswith='s').values()
10    template = loader.get_template('list_customers.html')
11    context = {
12        'listcustomer': mycustomers,
13    }
14    return HttpResponse(template.render(context, request))
```

**Equivalente SQL**

**O SQL equivalente ao exemplo acima será:**

**WHERE first\_name LIKE '%s';**

# PROGRAMAÇÃO BACK-END

Vá ao navegador e digite:

**<http://127.0.0.1:8000/listacustomer/>**

O resultado da página será a **listagem** a seguir.



## Customers

- Frances Parker
- Doris Reed
- Lois Butler
- Phyllis Foster
- Gladys Hamilton
- Dolores Wagner
- Agnes Bishop
- Delores Hansen
- James Gannon
- Charles Kowalski
- Thomas Grigsby
- Dennis Gilman
- Douglas Graf
- Nicholas Barfield
- Louis Leone
- Carlos Coughlin
- Chris Brothers

**Pesquisa de Campo  
iendswith**

# PROGRAMAÇÃO BACK-END

## Pesquisa de campo iendswith


### Definição e uso

A pesquisa **iendswith** é usada para obter registros que terminam com um valor especificado.

A pesquisa **iendswith** não diferencia maiúsculas de minúsculas.

# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** altere o seguinte conteúdo no arquivo **views.py** (linha 9):



```
8 def listcustomer(request):
9     mycustomers = Customer.objects.filter(first_name__endswith='s').values()
10    template = loader.get_template('list_customers.html')
11    context = {
12        'listcustomer': mycustomers,
13    }
14    return HttpResponse(template.render(context, request))
```

**Equivalente SQL**

**O SQL equivalente ao exemplo acima será:**

**WHERE first\_name LIKE '%s';**

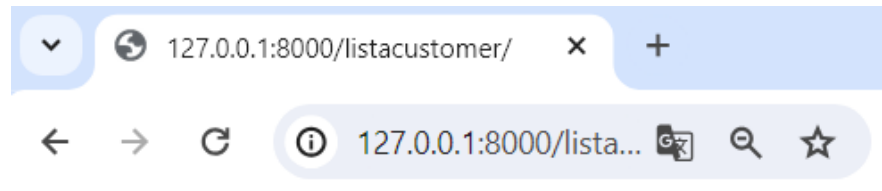


# PROGRAMAÇÃO BACK-END

Vá ao navegador e digite:

**<http://127.0.0.1:8000/listacustomer/>**

O resultado da página será a **listagem** a seguir.



## Customers

- Frances Parker
- Doris Reed
- Lois Butler
- Phyllis Foster
- Gladys Hamilton
- Dolores Wagner
- Agnes Bishop
- Delores Hansen
- James Gannon
- Charles Kowalski
- Thomas Grigsby
- Dennis Gilman
- Douglas Graf
- Nicholas Barfield
- Louis Leone
- Carlos Coughlin
- Chris Brothers

**Pesquisa de Campo  
exact**

# PROGRAMAÇÃO BACK-END

## Pesquisa de campo exact

### Definição e uso

A pesquisa **exact** é usada para obter registros com um valor especificado.

A pesquisa **exact** diferencia maiúsculas de minúsculas.

# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** altere o seguinte conteúdo no arquivo **views.py** (linha 9):



```
8 def listcustomer(request):
9     mycustomers = Customer.objects.filter(first_name__exact='Phyllis').values()
10    template = loader.get_template('list_customers.html')
11    context = {
12        'listcustomer': mycustomers,
13    }
14    return HttpResponse(template.render(context, request))
15
```

**Equivalente SQL**

**O SQL equivalente ao exemplo acima será:**

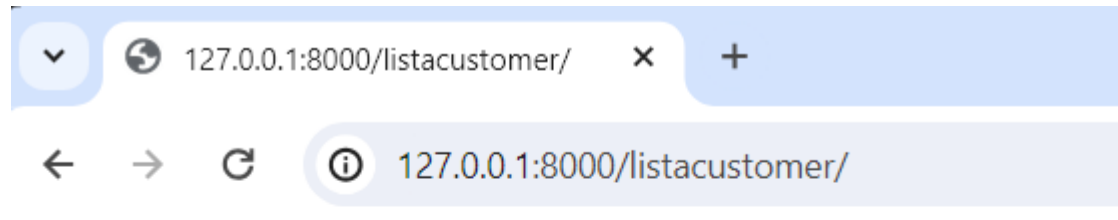
**WHERE first\_name = 'Phyllis';**

# PROGRAMAÇÃO BACK-END

Vá ao navegador e digite:

<http://127.0.0.1:8000/listacustomer/>

O resultado da  
página será a  
**listagem** a seguir.



## Customers

- Phyllis Foster

**Pesquisa de Campo  
iexact**

# PROGRAMAÇÃO BACK-END

## Pesquisa de campo **iexact**


### Definição e uso

A pesquisa **iexact** é usada para obter registros com um valor especificado.

A pesquisa **iexact** não diferencia maiúsculas de minúsculas.

# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** altere o seguinte conteúdo no arquivo **views.py** (linha 9):



```
8 def listcustomer(request):
9     mycustomers = Customer.objects.filter(first_name__iexact='phyllis').values()
10    template = loader.get_template('list_customers.html')
11    context = {
12        'listcustomer': mycustomers,
13    }
14    return HttpResponse(template.render(context, request))
```

## Equivalente SQL

O SQL equivalente ao exemplo acima será:

**WHERE first\_name = 'phyllis';**

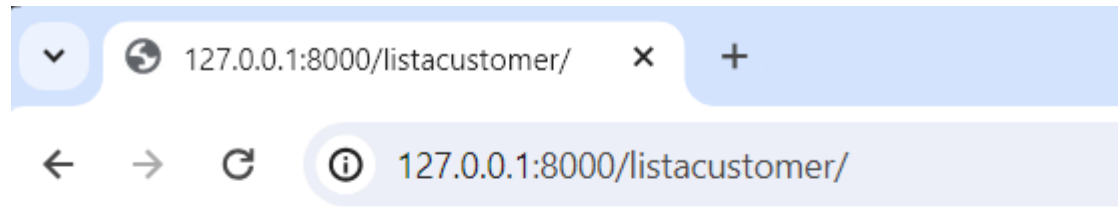


# PROGRAMAÇÃO BACK-END

Vá ao navegador e digite:

<http://127.0.0.1:8000/listacustomer/>

O resultado da  
página será a  
**listagem** a seguir.



## Customers

- Phyllis Foster

**Pesquisa de Campo  
in**

# PROGRAMAÇÃO BACK-END

## Pesquisa de campo in


### Definição e uso

A pesquisa **in** é usada para obter registros onde o valor é um dos valores em um iterável (lista, tupla, string, conjunto de consultas).

A pesquisa **in** diferencia maiúsculas de minúsculas.

# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** altere o seguinte conteúdo no arquivo **views.py** (linha 9):



```
8 def listacustomer(request):
9     mycustomers = Customer.objects.filter(first_name__in=['Phyllis','Dennis','Nicholas']).values()
10    template = loader.get_template('list_customers.html')
11    context = {
12        'listcustomer': mycustomers,
13    }
14    return HttpResponse(template.render(context, request))
```

## Equivalente SQL

O SQL equivalente ao exemplo acima será:

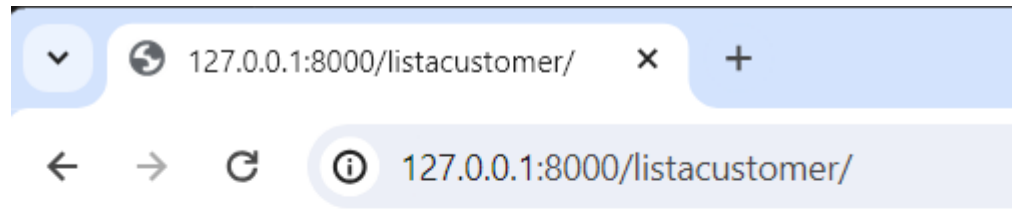
**WHERE first\_name IN ( 'Phyllis', 'Dennis', 'Nicholas');**

# PROGRAMAÇÃO BACK-END

Vá ao navegador e digite:

<http://127.0.0.1:8000/listacustomer/>

O resultado da página será a **listagem** a seguir.



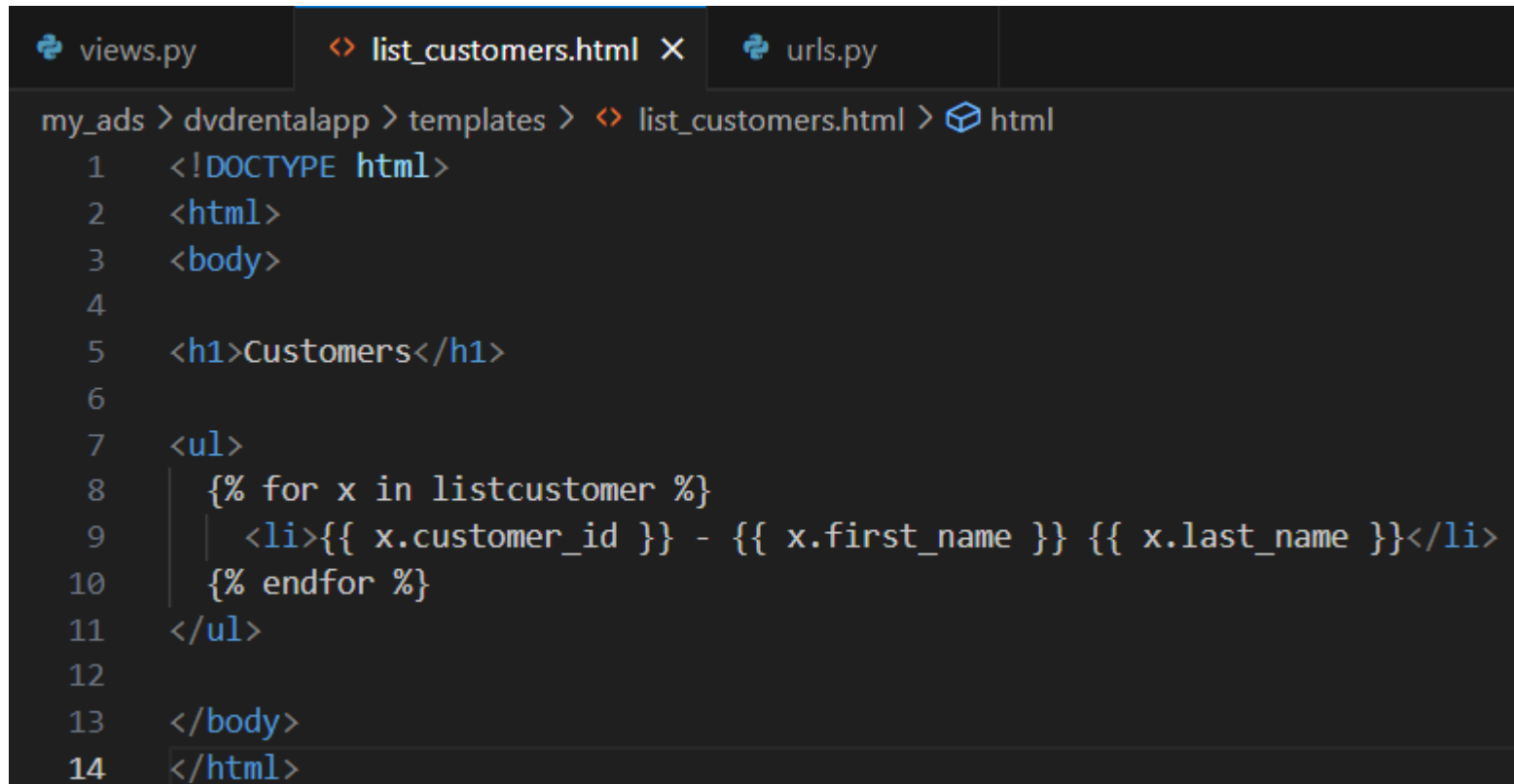
## Customers

- Phyllis Foster
- Dennis Gilman
- Nicholas Barfield

**Pesquisa de Campo**  
**gt**

# PROGRAMAÇÃO BACK-END

Vamos alterar o arquivo `list_customers.html` para aparecer o campo `customer_id`. Na pasta `templates` alterar o arquivo `list_customers.html` com o seguinte conteúdo:

A screenshot of a code editor with a dark theme. The top bar shows three tabs: 'views.py', 'list\_customers.html' (which is active and has a close button), and 'urls.py'. Below the tabs, the breadcrumb path 'my\_ads > dvdrentalapp > templates > list\_customers.html' is displayed, followed by a file icon and the text 'html'. The main area contains 14 lines of HTML code. Line 1: <!DOCTYPE html>. Line 2: <html>. Line 3: <body>. Line 4: (empty line). Line 5: <h1>Customers</h1>. Line 6: (empty line). Line 7: <ul>. Line 8: { % for x in listcustomer %}. Line 9: | <li>{{ x.customer\_id }} - {{ x.first\_name }} {{ x.last\_name }}</li>. Line 10: { % endfor %}. Line 11: </ul>. Line 12: (empty line). Line 13: </body>. Line 14: </html>.

```
views.py list_customers.html X urls.py
my_ads > dvdrentalapp > templates > list_customers.html > html
1 <!DOCTYPE html>
2 <html>
3 <body>
4
5 <h1>Customers</h1>
6
7 <ul>
8   { % for x in listcustomer %}
9   | <li>{{ x.customer_id }} - {{ x.first_name }} {{ x.last_name }}</li>
10  { % endfor %}
11 </ul>
12
13 </body>
14 </html>
```

# PROGRAMAÇÃO BACK-END

## Pesquisa de campo gt

### Definição e uso

A pesquisa **gt** é usada para obter registros maiores que um valor especificado.



# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** altere o seguinte conteúdo no arquivo **views.py** (linha 9):



```
8 def listcustomer(request):
9     mycustomers = Customer.objects.filter(customer_id__gt=500).values()
10    template = loader.get_template('list_customers.html')
11    context = {
12        'listcustomer': mycustomers,
13    }
14    return HttpResponse(template.render(context, request))
```

**Equivalente SQL**

**O SQL equivalente ao exemplo acima será:**

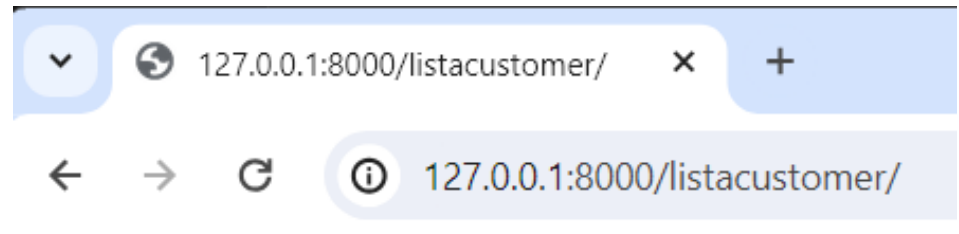
**WHERE customer\_id > 500**

# PROGRAMAÇÃO BACK-END

Vá ao navegador e digite:

<http://127.0.0.1:8000/listacustomer/>

O resultado da página será a **listagem** a seguir.



## Customers

- 501 - Ruben Geary
- 502 - Brett Cornwell
- 503 - Angel Barclay
- 504 - Nathaniel Adam
- 505 - Rafael Abney
- 506 - Leslie Seward
- 507 - Edgar Rhoads

**Pesquisa de Campo  
gte**

# PROGRAMAÇÃO BACK-END

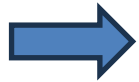
## Pesquisa de campo gte

### Definição e uso

A pesquisa **gte** é usada para obter registros maiores ou iguais que um valor especificado.

# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** altere o seguinte conteúdo no arquivo **views.py** (linha 9):



```
8 def listcustomer(request):
9     mycustomers = Customer.objects.filter(customer_id__gte=500).values()
10    template = loader.get_template('list_customers.html')
11    context = {
12        'listcustomer': mycustomers,
13    }
14    return HttpResponse(template.render(context, request))
```

**Equivalente SQL**

**O SQL equivalente ao exemplo acima será:**

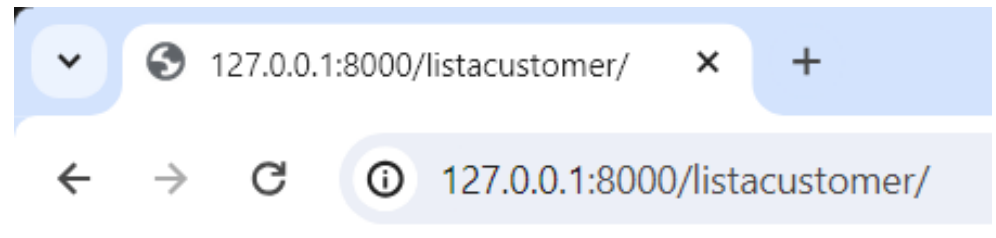
**WHERE customer\_id >= 500**

# PROGRAMAÇÃO BACK-END

Vá ao navegador e digite:

<http://127.0.0.1:8000/listacustomer/>

O resultado da página será a **listagem** a seguir.



## Customers

- 500 - Reginald Kinder
- 501 - Ruben Geary
- 502 - Brett Cornwell
- 503 - Angel Barclay

**Pesquisa de Campo  
It**

# PROGRAMAÇÃO BACK-END

## Pesquisa de campo **lt**

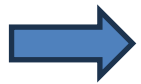
### Definição e uso

A pesquisa **lt** é usada para obter registros menores que um valor especificado.



# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** altere o seguinte conteúdo no arquivo **views.py** (linha 9):



```
8 def listacustomer(request):
9     mycustomers = Customer.objects.filter(customer_id__lt=15).values()
10    template = loader.get_template('list_customers.html')
11    context = {
12        'listcustomer': mycustomers,
13    }
14    return HttpResponse(template.render(context, request))
15
```

**Equivalente SQL**

**O SQL equivalente ao exemplo acima será:**

**WHERE customer\_id < 15**

# PROGRAMAÇÃO BACK-END

Vá ao navegador e digite:

**<http://127.0.0.1:8000/listacustomer/>**

O resultado da página será a **listagem** a seguir.



## Customers

- 1 - Mary Smith1
- 2 - Patricia Johnson1
- 3 - Linda Williams
- 4 - Barbara Jones
- 5 - Elizabeth Brown
- 6 - Jennifer Davis
- 7 - Maria Miller
- 8 - Susan Wilson
- 9 - Margaret Moore
- 10 - Dorothy Taylor
- 11 - Lisa Anderson
- 12 - Nancy Thomas
- 13 - Karen Jackson
- 14 - Betty White

**Pesquisa de Campo  
Ite**

# PROGRAMAÇÃO BACK-END


## Pesquisa de campo lte

### Definição e uso

A pesquisa **lte** é usada para obter registros menores ou iguais que um valor especificado.

# PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** altere o seguinte conteúdo no arquivo **views.py** (linha 9):



```
8 def listacustomer(request):
9     mycustomers = Customer.objects.filter(customer_id__lte=15).values()
10    template = loader.get_template('list_customers.html')
11    context = {
12        'listcustomer': mycustomers,
13    }
14    return HttpResponse(template.render(context, request))
```

**Equivalente SQL**

**O SQL equivalente ao exemplo acima será:**

**WHERE customer\_id <= 15**

# PROGRAMAÇÃO BACK-END

Vá ao navegador e digite:

<http://127.0.0.1:8000/listacustomer/>

O resultado da página será a **listagem** a seguir.



## Customers

- 1 - Mary Smith1
- 2 - Patricia Johnson1
- 3 - Linda Williams
- 4 - Barbara Jones
- 5 - Elizabeth Brown
- 6 - Jennifer Davis
- 7 - Maria Miller
- 8 - Susan Wilson
- 9 - Margaret Moore
- 10 - Dorothy Taylor
- 11 - Lisa Anderson
- 12 - Nancy Thomas
- 13 - Karen Jackson
- 14 - Betty White
- 15 - Helen Harris