

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.

Django Forms

PROGRAMAÇÃO BACK-END

Neste exemplo será listado a tabela Customer do banco dvdrental.

O formato será em tabela.

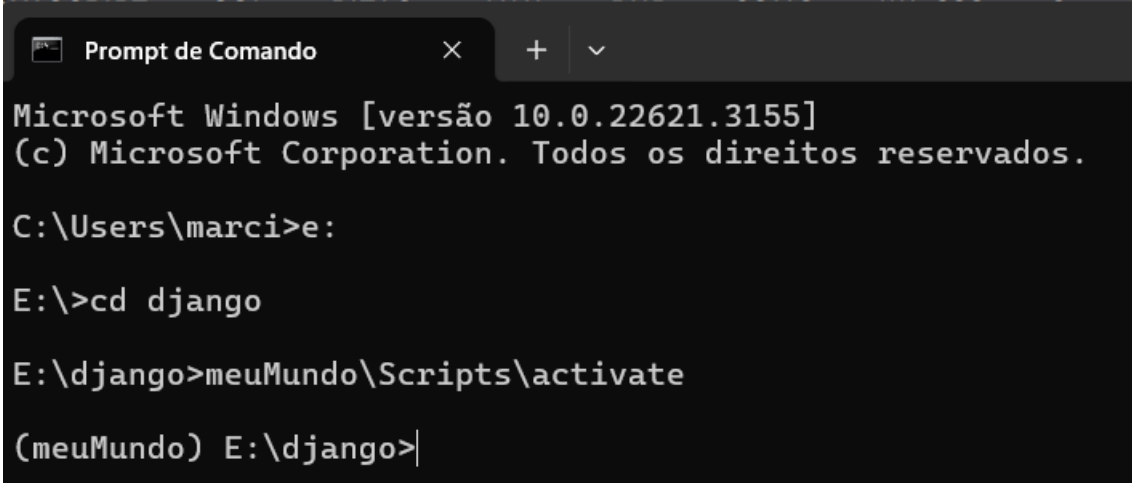
Tem-se duas ações (Edit e New) que chamara as templates relativo a edição e a inserção de novo Customer.

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

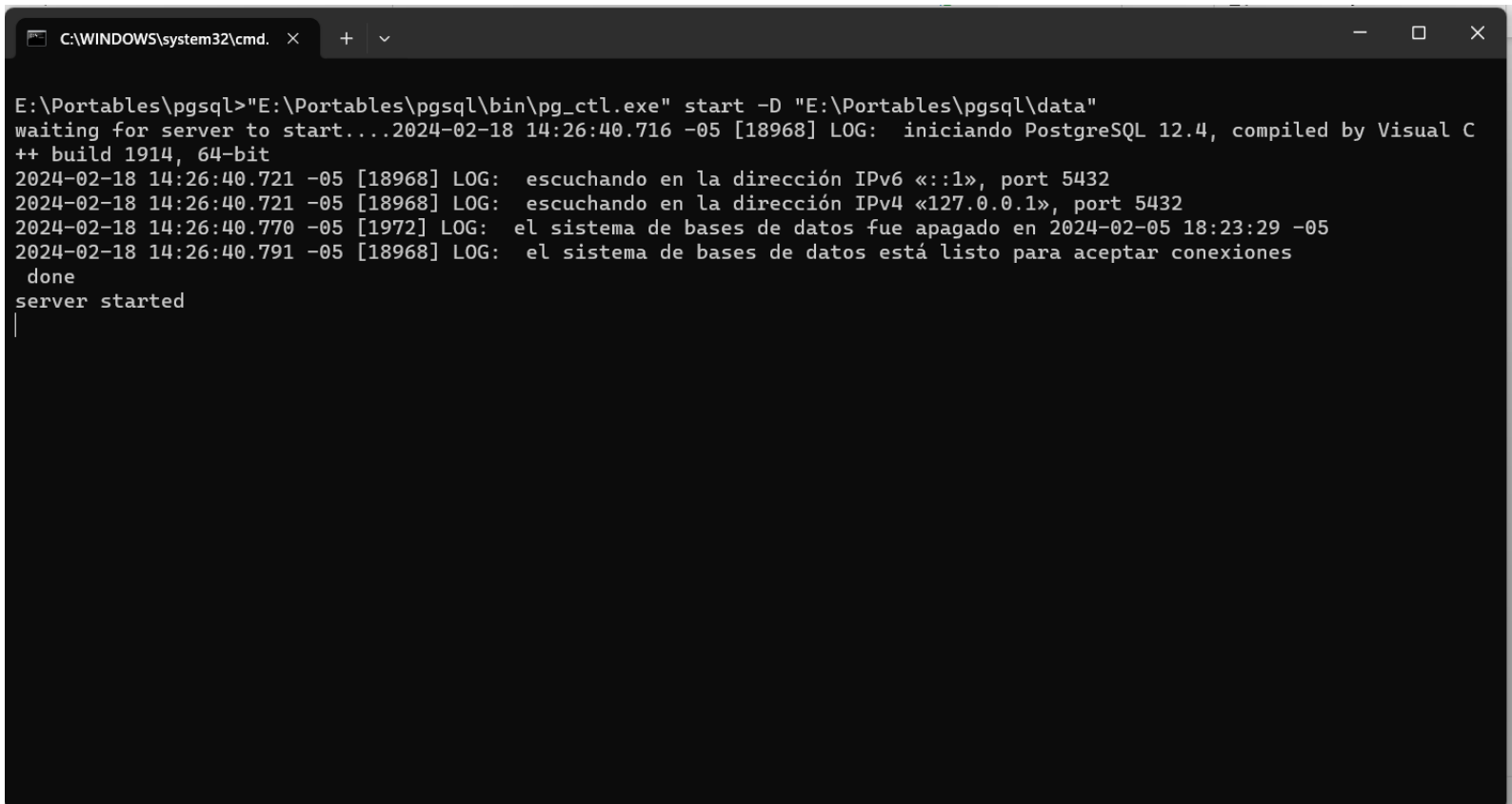
Na pasta `../pgsql` execute o arquivo **PostgreSQL-Start.bat** para iniciar o servidor



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



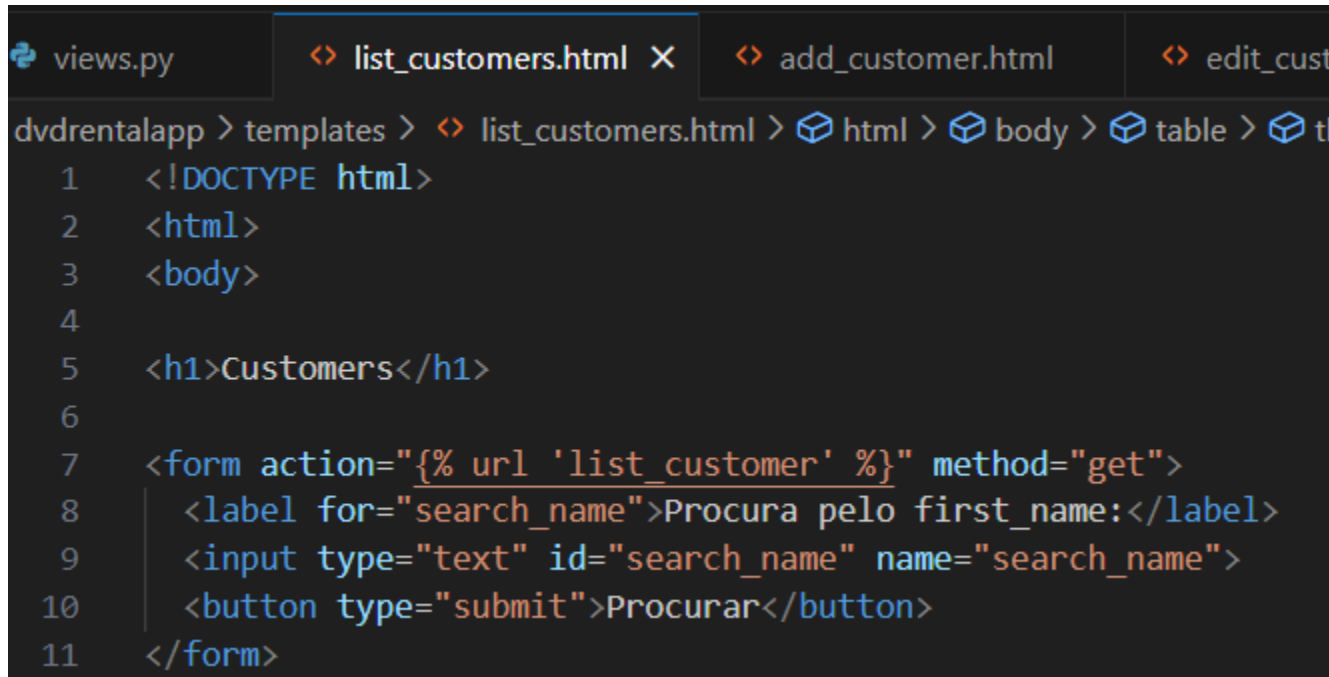
```
C:\WINDOWS\system32\cmd. x + v

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

Utilizando a App **dvdrentalapp**. Na pasta **templates** crie/altere o arquivo **list_customers.html** e digite o seguinte conteúdo:



```
views.py  list_customers.html X  add_customer.html  edit_cust
dvdrentalapp > templates > list_customers.html > html > body > table > th
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h1>Customers</h1>
6
7  <form action="{% url 'list_customer' %}" method="get">
8    <label for="search_name">Procura pelo first_name:</label>
9    <input type="text" id="search_name" name="search_name">
10   <button type="submit">Procurar</button>
11  </form>
```

PROGRAMAÇÃO BACK-END

Continuação do arquivo `list_customers.html`

```
12
13 <table>
14   <thead>
15     <tr>
16       <th>Customer id</th>
17       <th>First Name</th>
18       <th>Last Name</th>
19       <th>Actions</th>
20     </tr>
21   </thead>
22   <tbody>
23     {% for x in listcustomer %}
24     <tr>
25       <td data-label="Id">{{ x.customer_id }}</td>
26       <td data-label="first_name">{{ x.first_name }}</td>
27       <td data-label="last_name">{{ x.last_name }}</td>
28       <td><a href="{% url 'edit_customer' x.customer_id %}"><button>Edit</button></a>
29       <a href="{% url 'add_customer' %}"><button>New</button></a></td>
30     </tr>
31     {% endfor %}
32   </tbody>
33 </table>
34
35 </body>
36 </html>
```

PROGRAMAÇÃO BACK-END

Crie também o arquivo `edit_customer.html` e digite o seguinte conteúdo:

```
views.py  list_customers.html  add_customer.html
dvdrentalapp > templates > edit_customer.html > ...
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Editar Cliente</title>
5  </head>
6  <body>
7
8  <h2>Editar Cliente</h2>
9
10 <form method="post">
11 |   {% csrf_token %}
12 |   {{ form.as_p }}
13 |   <input type="submit" value="Salvar Alterações">
14 </form>
15
16 </body>
17 </html>
```

PROGRAMAÇÃO BACK-END

Crie também o arquivo **add_customer.html** e digite o seguinte conteúdo:

```
views.py  list_customers.html  add_customer.html X  e
dvdrentalapp > templates > add_customer.html > html > body > h2
1  <!DOCTYPE html>
2  <html>
3  <head>
4  |   <title>Adicionar Cliente</title>
5  </head>
6  <body>
7
8  <h2>Adicionar Cliente</h2>
9
10 <form method="post">
11 |   {% csrf_token %}
12 |   {{ form.as_p }}
13 |   <input type="submit" value="Salvar Alterações">
14 </form>
15
16 </body>
17 </html>
```

PROGRAMAÇÃO BACK-END

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

PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** adicione o seguinte conteúdo no arquivo **views.py**

```
views.py  x  list_customers.html  add_customer.html  edit_customer.html  urls.py  url

dvdrentalapp > views.py > edit_customer
1  # Create your views here.
2  from django.http import HttpResponse
3  from django.template import loader
4  from .models import Customer
5  from django.shortcuts import render, get_object_or_404, redirect
6  from .forms import CustomerForm
7
8  def list_customer(request):
9      search_name = request.GET.get( 'search_name' , '')
10     if search_name:
11         listacustomer = Customer.objects.filter(first_name__contains=search_name) . values()
12     else:
13         listacustomer = Customer.objects.all().values()
14
15     template = loader.get_template('list_customers.html')
16     context = {
17         'listcustomer': listacustomer,
18     }
19     return HttpResponse(template.render(context, request))
20
```


PROGRAMAÇÃO E

Continuação do arquivo views.py

```
20
21
22 def edit_customer(request, id):
23     customer = get_object_or_404(Customer, pk=id)
24     if request.method == 'POST':
25         form = CustomerForm(request.POST, instance=customer)
26         if form.is_valid():
27             form.save()
28             return redirect('list_customer')
29     else:
30         form = CustomerForm(instance=customer)
31
32     context = {
33         'form': form,
34     }
35     return render(request, 'edit_customer.html', context)
36
37 def add_customer(request):
38     if request.method == 'POST':
39         form = CustomerForm(request.POST)
40         if form.is_valid():
41             form.save()
42             return redirect('list_customer')
43     else:
44         form = CustomerForm()
45
46     context = {
47         'form': form,
48     }
49     return render(request, 'add_customer.html', context)
```


PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** edite o seguinte conteúdo no arquivo **urls.py**.

```
dvdrentalapp >  urls.py > ...  
1  from django.contrib.auth import views as auth_views  
2  from django.urls import path  
3  from . import views  
4  
5  urlpatterns = [  
6      path('list_customer/', views.list_customer, name = 'list_customer'),  
7      path('edit_customer/<int:id>/', views.edit_customer, name = 'edit_customer'),  
8      path('add_customer/', views.add_customer, name = 'add_customer'),  
9  
10 ]
```

PROGRAMAÇÃO BACK-END

Na pasta **dvdrentalapp** edite o arquivo **forms.py** adicione o seguinte conteúdo no arquivo:

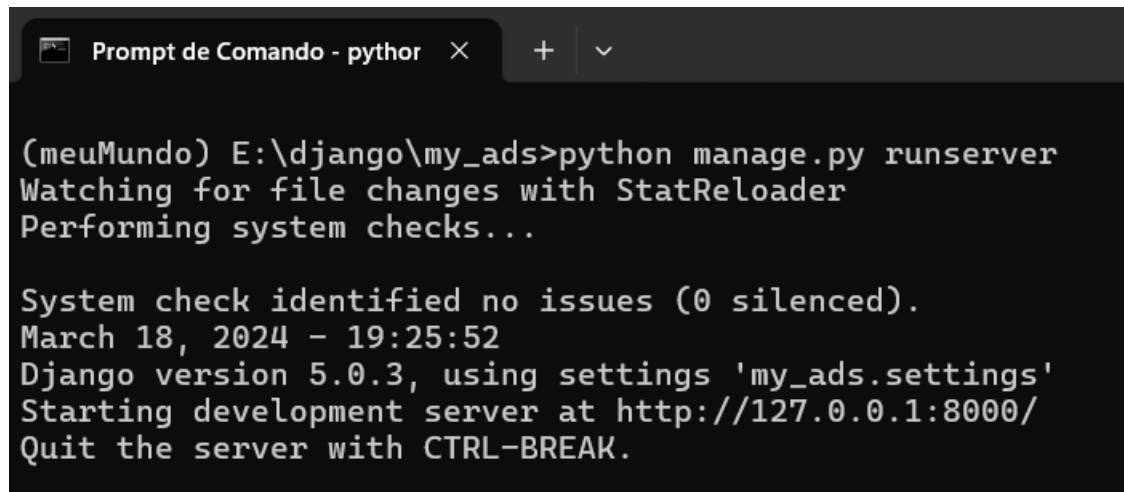
```
dvdrentalapp > forms.py > ...
1  from django.forms import ModelForm
2  from django import forms
3  from dvdrentalapp.models import *
4  from .models import Customer
5
6  class CustomerForm(ModelForm):
7      class Meta:
8          model = Customer
9          #fields = '__all__'
10         fields = ['first_name', 'last_name', 'email', 'address', 'activebool', 'store_id', 'create_date']
11
```

**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/list_customer/

PROGRAMAÇÃO BACK-END

O resultado da página será:

127.0.0.1:8000/list_customer/?search_name=

Customers

Procura pelo first_name:

Customer id	First Name	Last Name	Actions	
3	Linda	Williams	<input type="button" value="Edit"/>	<input type="button" value="New"/>
4	Barbara	Jones	<input type="button" value="Edit"/>	<input type="button" value="New"/>
5	Elizabeth	Brown	<input type="button" value="Edit"/>	<input type="button" value="New"/>
6	Jennifer	Davis	<input type="button" value="Edit"/>	<input type="button" value="New"/>
7	Maria	Miller	<input type="button" value="Edit"/>	<input type="button" value="New"/>
8	Susan	Wilson	<input type="button" value="Edit"/>	<input type="button" value="New"/>
9	Margaret	Moore	<input type="button" value="Edit"/>	<input type="button" value="New"/>
10	Dorothy	Taylor	<input type="button" value="Edit"/>	<input type="button" value="New"/>
11	Lisa	Anderson	<input type="button" value="Edit"/>	<input type="button" value="New"/>
12	Nancy	Thomas	<input type="button" value="Edit"/>	<input type="button" value="New"/>
13	Karen	Jackson	<input type="button" value="Edit"/>	<input type="button" value="New"/>
14	Betty	White	<input type="button" value="Edit"/>	<input type="button" value="New"/>
15	Helen	Harris	<input type="button" value="Edit"/>	<input type="button" value="New"/>
16	Sandra	Martin	<input type="button" value="Edit"/>	<input type="button" value="New"/>

Editar Cliente

127.0.0.1:8000/edit_customer/3/

Editar Cliente

First name:

Last name:

Email:

Address:

Activebool: ☒

Store id:

Create date:

Adicionar Cliente

127.0.0.1:8000/add_customer,

Adicionar Cliente

First name:

Last name:

Email:

Address:

Activebool: ☐

Store id:

Create date:

PROGRAMAÇÃO BACK-END

Tarefa

PROGRAMAÇÃO BACK-END

Faça as alterações para a tabela dos **Category**