Para criar a tabela de mesas e exibi-las ao clicar no menu de mesas, você precisará adicionar um modelo para as mesas, criar uma view para lidar com a exibição das mesas e ajustar as URLs e os templates conforme necessário. Vamos seguir passo a passo:

1. **Criação do modelo para as mesas**:

No seu arquivo models.py, adicione o seguinte código para criar o modelo de mesas:

python

from django.db import models

class Mesa(models.Model):

nome = models.CharField(max\_length=10)

def \_\_str\_\_(self):

return self.nome

1. **Inicialização das mesas no banco de dados**:

Você pode criar uma migração inicial e um script de inicialização para adicionar as mesas automaticamente ao banco de dados. Para isso, crie um arquivo signals.py dentro do seu aplicativo e adicione o seguinte código:

python

from django.db.models.signals import post\_migrate

from django.dispatch import receiver

from .models import Mesa

@receiver(post\_migrate)

def create\_initial\_mesas(sender, \*\*kwargs):

if sender.name == 'app\_chop':

if Mesa.objects.count() == 0:

for i in range(1, 11):

Mesa.objects.create(nome=f'Mesa {i}')

No seu arquivo apps.py, adicione o seguinte código para garantir que os sinais sejam registrados:

python

from django.apps import AppConfig

class AppChopConfig(AppConfig):

name = 'app\_chop'

def ready(self):

import app\_chop.signals

1. **Criação da view para exibir as mesas**:

No seu arquivo views.py, adicione uma view para listar as mesas:

from .models import Mesa

def lista\_mesas(request):

mesas = Mesa.objects.all()

context = {

'mesas': mesas

}

return render(request, 'app\_chop/lista\_mesas.html', context)

1. **Adição da URL para a view de mesas**:

No seu arquivo urls.py, adicione o caminho para a nova view:

urlpatterns = [

# ... outras URLs ...

path('mesas/', views.lista\_mesas, name='lista\_mesas'),

]

1. **Criação do template para exibir as mesas**:

Crie um arquivo lista\_mesas.html na pasta templates/app\_chop/ com o seguinte conteúdo:

{% extends "base.html" %}

{% block conteudo %}

<div class="container">

<h2>Mesas Disponíveis</h2>

<ul class="list-group">

{% for mesa in mesas %}

<li class="list-group-item">{{ mesa.nome }}</li>

{% endfor %}

</ul>

</div>

{% endblock %}

1. **Atualização do menu de navegação**:

Certifique-se de que o link do menu para mesas está apontando para a nova URL. No seu arquivo base.html, ajuste o link do menu de mesas para:

<li class="sidebar-item">

<a href="{% url 'lista\_mesas' %}" class="sidebar-link">

<i class="lni lni-dinner"></i>

<span>Mesas</span>

</a>

</li>

1. **Aplicação das migrações e testes**:

Execute as migrações para criar a tabela de mesas no banco de dados:

python manage.py makemigrations

python manage.py migrate

Após seguir esses passos, você deverá ter uma tabela de mesas criada no banco de dados, inicializada com 10 mesas. Essas mesas serão exibidas na página de mesas quando você clicar no link "Mesas" no menu lateral.

Se ocorreu erro

O erro ocorre porque o sinal post\_migrate é registrado antes da definição do modelo Mesa. Para resolver esse problema, é necessário garantir que o modelo seja definido antes de importar e registrar os sinais.

Vamos seguir um método alternativo sem precisar criar sinais customizados, aproveitando o comando post\_migrate diretamente na configuração do aplicativo. Aqui está uma abordagem revisada:

1. **Criação do modelo Mesa**:

No seu arquivo models.py, adicione o modelo Mesa:

from django.db import models

class Mesa(models.Model):

nome = models.CharField(max\_length=10)

def \_\_str\_\_(self):

return self.nome

1. **Adicionar a lógica de inicialização das mesas no arquivo apps.py**:

No seu arquivo apps.py, registre um hook para post\_migrate que criará as mesas após as migrações:

from django.apps import AppConfig

from django.db.models.signals import post\_migrate

def create\_initial\_mesas(sender, \*\*kwargs):

from .models import Mesa

if Mesa.objects.count() == 0:

for i in range(1, 11):

Mesa.objects.create(nome=f'Mesa {i}')

class AppChopConfig(AppConfig):

name = 'app\_chop'

def ready(self):

post\_migrate.connect(create\_initial\_mesas, sender=self)

1. **Criação da view para exibir as mesas**:

No seu arquivo views.py, adicione uma view para listar as mesas:

from django.shortcuts import render

from .models import Mesa

def lista\_mesas(request):

mesas = Mesa.objects.all()

context = {

'mesas': mesas

}

return render(request, 'app\_chop/lista\_mesas.html', context)

1. **Adicionar a URL para a view de mesas**:

No seu arquivo urls.py, adicione o caminho para a nova view:

from django.urls import path

from . import views

urlpatterns = [

# ... outras URLs ...

path('mesas/', views.lista\_mesas, name='lista\_mesas'),

]

1. **Criação do template para exibir as mesas**:

Crie um arquivo lista\_mesas.html na pasta templates/app\_chop/ com o seguinte conteúdo:

{% extends "base.html" %}

{% block conteudo %}

<div class="container">

<h2>Mesas Disponíveis</h2>

<ul class="list-group">

{% for mesa in mesas %}

<li class="list-group-item">{{ mesa.nome }}</li>

{% endfor %}

</ul>

</div>

{% endblock %}

1. **Atualização do menu de navegação**:

Certifique-se de que o link do menu para mesas está apontando para a nova URL. No seu arquivo base.html, ajuste o link do menu de mesas para:

<li class="sidebar-item">

<a href="{% url 'lista\_mesas' %}" class="sidebar-link">

<i class="lni lni-dinner"></i>

<span>Mesas</span>

</a>

</li>

1. **Aplicação das migrações e testes**:

Execute as migrações para criar a tabela de mesas no banco de dados:

python manage.py makemigrations

python manage.py migrate

Seguindo esses passos, você deve resolver o problema de importação e garantir que as mesas sejam criadas e exibidas corretamente.

**Apagar o banco de dados**

rm db.sqlite3

**Apagar as migrações:**

rm -rf app\_chop/migrations