

## 2º Passo - To-Do List com Django

### Parte 1: Adicionar um Modelo **Tarefa** para Armazenar as Tarefas

#### 1. Definir o Modelo **Tarefa**:

- No arquivo **models.py** da aplicação **tarefas**, adicione um modelo para representar as tarefas:

```
from django.db import models
class Tarefa(models.Model):
    titulo = models.CharField(max_length=100)
    concluida = models.BooleanField(default=False)

    def __str__(self):
        return self.titulo
```

- Aqui, cada **Tarefa** tem um campo **titulo** para o nome da tarefa e um campo **concluida** que indica se a tarefa foi concluída.

#### 2. Criar e Aplicar as Migrações:

- No terminal, execute os comandos para criar e aplicar as migrações, o que criará a tabela **Tarefa** no banco de dados:

```
python manage.py makemigrations
python manage.py migrate
```

---

### Parte 2: Ajustar a View para Buscar Tarefas do Banco de Dados

#### 1. Modificar a View **lista\_tarefas**:

- No arquivo **views.py** da aplicação **tarefas**, ajuste a view para buscar as tarefas do banco de dados:

```
from django.shortcuts import render
from .models import Tarefa
```

```
def lista_tarefas(request):
    tarefas = Tarefa.objects.all()
    return render(request, 'lista_tarefas.html',
                  {'tarefas': tarefas})
```

- Agora, a view `lista_tarefas` busca todas as tarefas armazenadas no banco de dados e as passa para o template.

## 2. Atualizar o Template `lista_tarefas.html`:

- No arquivo `lista_tarefas.html`, mantenha o loop `for` para exibir as tarefas:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Lista de Tarefas</title>
</head>
<body>
    <h1>Minha Lista de Tarefas</h1>
    <ul>
        {% for tarefa in tarefas %}
            <li>{{ tarefa.titulo }} - {% if
tarefa.concluida %}Concluída{% else %}Pendente{% endif
%}</li>
        {% endfor %}
    </ul>
</body>
</html>
```

- O template agora exibe as tarefas e indica se cada uma está concluída ou pendente.

---

## Parte 3: Criar um Formulário para Adicionar Novas Tarefas

### 1. Adicionar um Formulário no Template `lista_tarefas.html`:

- Adicione um formulário simples para permitir que o usuário adicione novas tarefas diretamente pela interface:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Lista de Tarefas</title>
</head>
<body>
    <h1>Minha Lista de Tarefas</h1>
    <form method="POST">
        {% csrf_token %}
        <input type="text" name="titulo"
placeholder="Adicione uma nova tarefa" required>
        <button type="submit">Adicionar</button>
    </form>
    <ul>
        {% for tarefa in tarefas %}
            <li>{{ tarefa.titulo }} - {% if
tarefa.concluida %}Concluída{% else %}Pendente{% endif
%}</li>
        {% endfor %}
    </ul>
</body>
</html>
```

- O formulário utiliza o método **POST** para enviar o título da nova tarefa e inclui a tag `{% csrf_token %}` para proteger contra ataques CSRF.

## 2. Atualizar a View `lista_tarefas` para Processar o Formulário:

- Modifique a view `lista_tarefas` para lidar com a criação de novas tarefas:

```
from django.shortcuts import render, redirect
from .models import Tarefa
def lista_tarefas(request):
    if request.method == 'POST':
        titulo = request.POST.get('titulo')
        if titulo:
            Tarefa.objects.create(titulo=titulo)
```

```
        return redirect('lista_tarefas')
    tarefas = Tarefa.objects.all()
    return render(request, 'lista_tarefas.html',
                  {'tarefas': tarefas})
```

- Agora, a view verifica se a requisição é **POST** e, se um título for fornecido, cria uma nova tarefa e redireciona para a página de lista de tarefas, atualizando a lista.

### 3. Testar a Adição de Novas Tarefas:

- No terminal, inicie o servidor:

```
python manage.py runserver
```

- Acesse <http://127.0.0.1:8000/> e adicione algumas tarefas pelo formulário.
  - Verifique se as tarefas aparecem na lista abaixo do formulário.
-