



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento DJANGO WEB



1 - Relembrar é viver

1.1 A linguagem Python

Python é uma linguagem de alto nível, extremamente versátil, de sintaxe simples que facilita o aprendizado. Criado em 1989, teve sua primeira versão publicada em 1991, e ano após ano tem se tornado cada vez mais popular.

1.2 O que é um framework?

Um framework é um conjunto de ferramentas e regras que visam facilitar e agilizar o desenvolvimento, simplificando tarefas rotineiras e permitindo maior foco na lógica e nas funcionalidades de determinada aplicação.

1.3 O que é o Django?

Django é um framework para o desenvolvimento web, baseado em Python e que segue a estrutura MVT (*Model, View, Template*). Amplamente aceito no mercado, o Django é utilizado por sites como Disqus, Instagram, Spotify, YouTube e Dropbox.



Programa Campus Inteligente

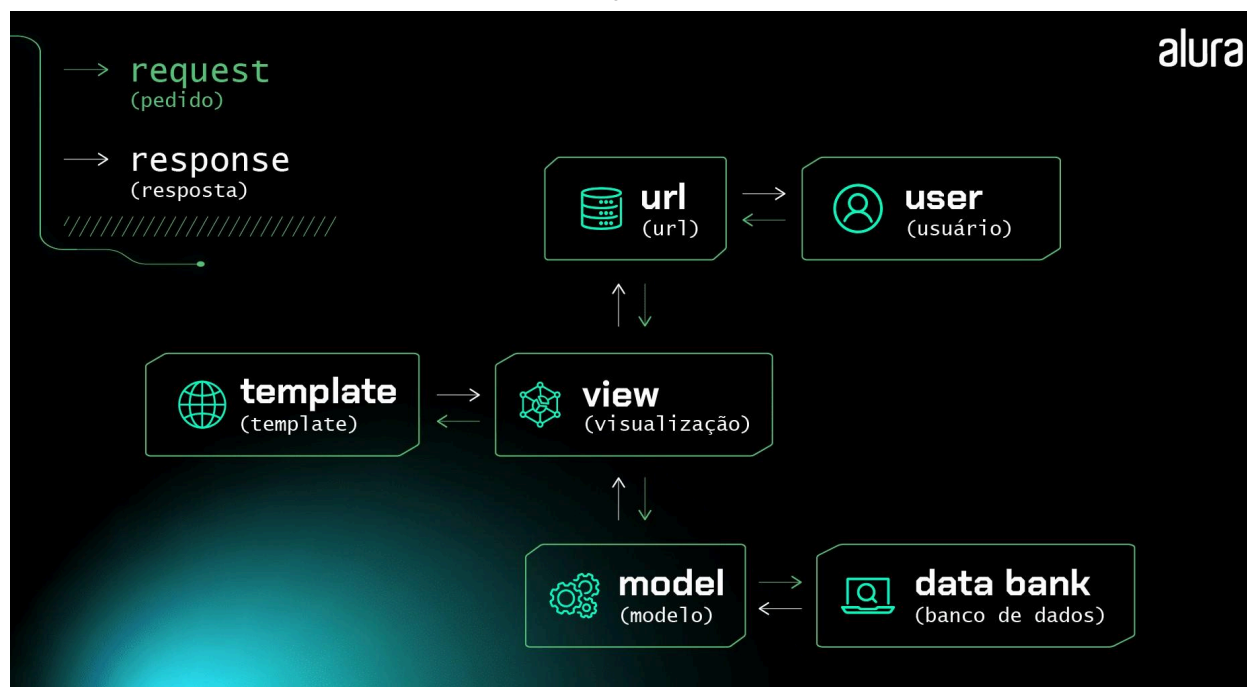
Iniciação Educacional no Desenvolvimento **DJANGO WEB**

1.4 Model, View, Template

Model: Responsável por definir a estrutura do banco de dados, criando tabelas, campos e relacionamentos.

View: Responsável pela lógica da aplicação, atuando como uma ponte entre o *model* e o *template*.

Template: Lida com a apresentação dos dados, é aqui que é construído o *front-end* da aplicação em Django.





Programa Campus Inteligente

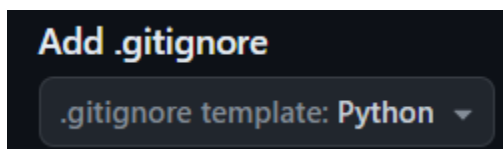
Iniciação Educacional no Desenvolvimento *DJANGO WEB*



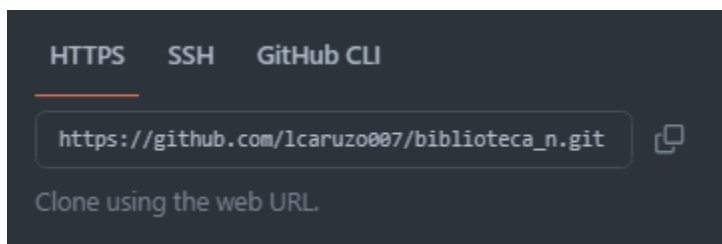
2 - Preparação de ambiente

Agora, vamos preparar nosso ambiente de desenvolvimento para iniciar nosso primeiro projeto com o framework Django.

- 2.1. Certifique-se de ter instalado Python e Postgres em seu computador. Para evitar erros de versão, certifique-se de que está utilizando a 3.11.5 do Python.
- 2.2. É importante que você possua uma conta do Github para realizar o versionamento dos seus códigos. Mais tarde trabalharemos com isto.
- 2.3. Na sua conta Github, crie um novo repositório selecionando Python como template do arquivo .gitignore:



- 2.4. Criado o repositório, vamos então cloná-lo em nossa máquina, para isso, copie o link https seguindo o caminho abaixo:





Programa Campus Inteligente

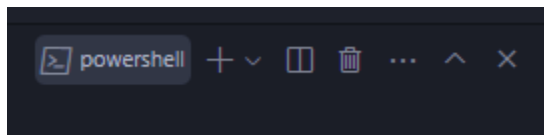
Iniciação Educacional no Desenvolvimento **DJANGO WEB**



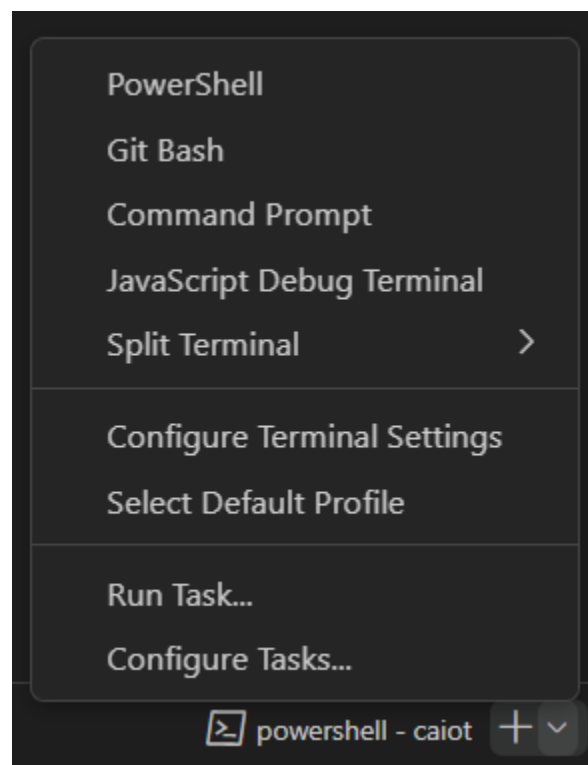
2.5. Com o link copiado, abra o editor de código Visual Studio Code. Inicie um novo terminal (Ctrl+Shift+').

```
C:\Users\Usuario> |
```

2.6. Note que o 'PS' indica que estamos utilizando o PowerShell. Queremos utilizar o Prompt de Comando. Para isso, siga o caminho:



2.6.1. Seta para baixo:





Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**



2.6.2. Selecione 'Command Prompt', e esta mensagem deve aparecer no terminal:

```
Microsoft Windows [versão 10.0.19045.4046]  
(c) Microsoft Corporation. Todos os direitos reservados.  
  
C:\Users\Usuario>
```

2.7. Clone seu repositório para a máquina com o comando 'git clone':

```
C:\Users\Usuario>git clone https://github.com/lcaruzo007/biblioteca_n.git
```

2.8. Abra a pasta do repositório que você acabou de clonar no Visual Studio Code (Ctrl+K+O), ela terá o mesmo nome que você inseriu na hora de criar o repositório no GitHub :

```
▼ BIBLIOTECA_N
```

2.9. Agora, uma sequência de comandos para inicializar se o projeto. Siga atentamente:

2.9.1. para iniciar um ambiente virtual python: **python -m venv .venv**

2.9.2. para acessar o ambiente virtual .venv criado: **cd .venv/scripts**

2.9.3. para ativar o ambiente virtual: **activate**

2.9.4. para retornar ao diretório principal: **cd ../..**

2.9.5. para instalar as dependências do projeto: **pip install django psycopg2 pillow**



Programa Campus Inteligente

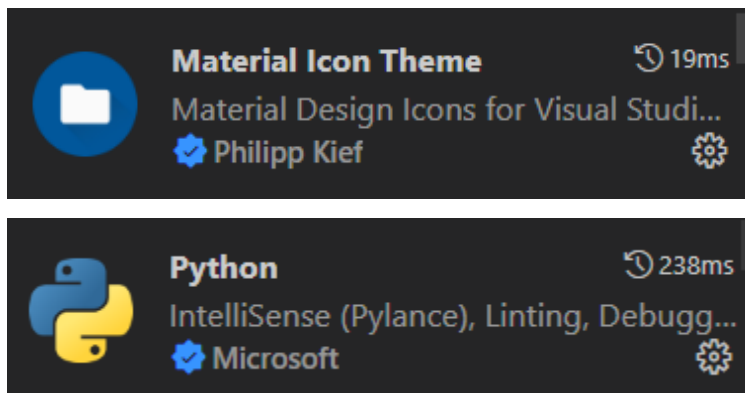
Iniciação Educacional no Desenvolvimento *DJANGO WEB*



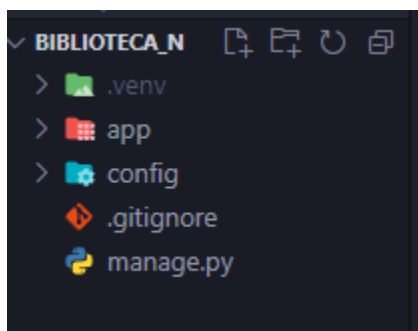
2.9.6. para gerar a pasta de configuração do projeto: **python -m django startproject config .**

2.9.7. para gerar a pasta do aplicativo: **python -m django startapp app**

2.10. É recomendado que você instale duas extensões (Ctrl+Shift+X) para facilitar o desenvolvimento:



2.11. Você deve se deparar com uma estrutura de pastas semelhante:



2.12. Por fim, vamos criar na interface do banco de dados PostgreSQL, o **pgAdmin**, o banco de dados que iremos trabalhar.



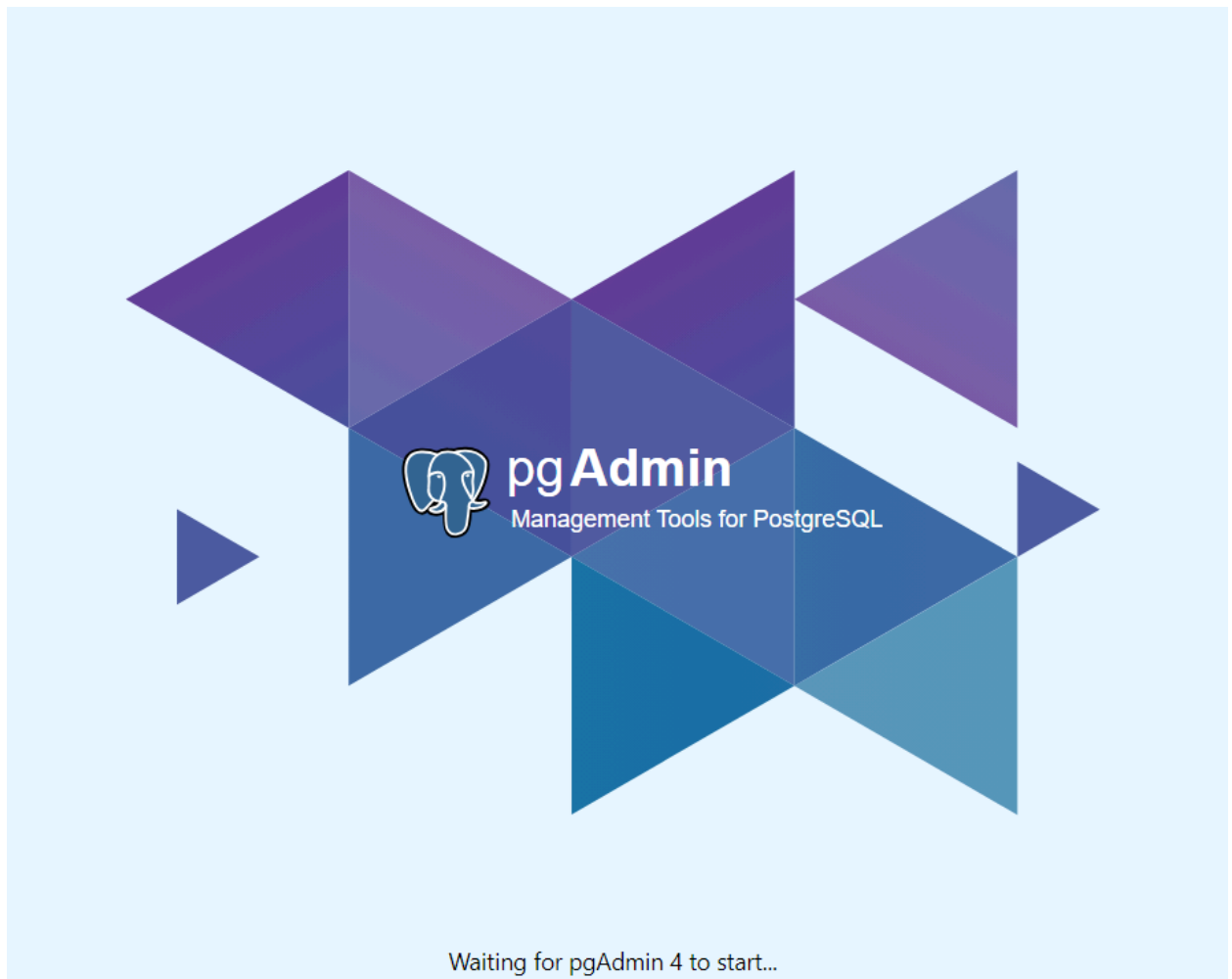
Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



2.13.

2.12.1. Acesse o pgAdmin:

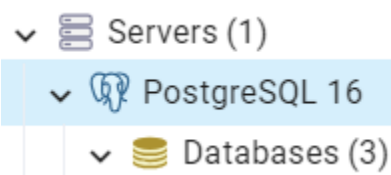


2.12.2. Aguarde até que o pgAdmin inicialize. Ao inicializar, clique em 'Servers', clicando em PostgreSQL 16 (ou PostgreSQL 15), será pedida uma senha de acesso, esta senha é por padrão **'123456'**:

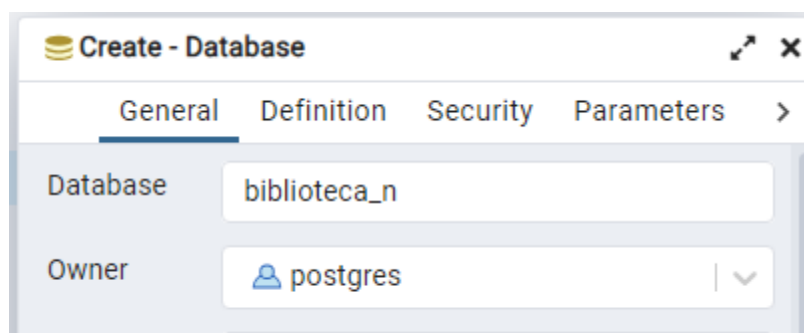


Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



2.12.3. Em 'Databases', clique com o botão direito do mouse, **Create > Database**. O campo 'Database' do formulário que for exibido é onde você irá inserir o nome do seu banco de dados.



2.12.4. Aperte em 'Save'. O banco de dados será criado com sucesso.



3 - Primeiros passos

3.1 Configuração do projeto

Agora, nós vamos configurar o nosso projeto Django para começarmos o desenvolvimento.



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



- 3.1.1. **Acesse a pasta 'config' e abra o arquivo 'settings.py'.** Siga as instruções e deixe o código como apresentado:

```
import os
from pathlib import Path

BASE_DIR = Path(__file__).resolve().parent.parent

SECRET_KEY =
'django-insecure-8l8o(-2su$6t%k424293i#x672q0$*^itfyi9r32$8j-igo7zf'

DEBUG = True

ALLOWED_HOSTS = []

INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'app',
]

#o template usado
MIDDLEWARE = [
    'django.middleware.security.SecurityMiddleware',
    'django.contrib.sessions.middleware.SessionMiddleware',
    'django.middleware.common.CommonMiddleware',
    'django.middleware.csrf.CsrfViewMiddleware',
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



```
'django.contrib.auth.middleware.AuthenticationMiddleware',
'django.contrib.messages.middleware.MessageMiddleware',
'django.middleware.clickjacking.XFrameOptionsMiddleware',
]

ROOT_URLCONF = 'config.urls'

TEMPLATES = [
    {
        'BACKEND': 'django.template.backends.django.DjangoTemplates',
        'DIRS': [
            os.path.join(BASE_DIR, 'app/templates'),
        ],
        'APP_DIRS': True,
        'OPTIONS': {
            'context_processors': [
                'django.template.context_processors.debug',
                'django.template.context_processors.request',
                'django.contrib.auth.context_processors.auth',
                'django.contrib.messages.context_processors.messages',
            ],
        },
    ],
]

WSGI_APPLICATION = 'config.wsgi.application'

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



```
'NAME': 'biblioteca_n',
'USER': 'postgres',
'PASSWORD': '123456',
'HOST': 'localhost',
'PORT': '5432',
}
}

AUTH_PASSWORD_VALIDATORS = [
    {
        'NAME':
'django.contrib.auth.password_validation.UserAttributeSimilarityVali
dator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.MinimumLengthValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.CommonPasswordValidator',
    },
    {
        'NAME':
'django.contrib.auth.password_validation.NumericPasswordValidator',
    },
]

LANGUAGE_CODE = 'pt-br'

TIME_ZONE = 'America/Sao_Paulo'
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



```
USE_I18N = True

USE_TZ = True

STATIC_URL = 'static/'
STATIC_ROOT = os.path.join(STATIC_URL, "static")
STATICFILES_DIRS = [
    os.path.join(BASE_DIR, "app/static/"),
]

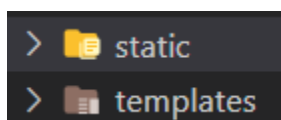
DEFAULT_AUTO_FIELD = 'django.db.models.BigAutoField'
```

- 3.1.2. Ainda na pasta **'config'**, **acesse** o arquivo **'urls.py'** e defina conforme apresentado:

```
from django.contrib import admin
from django.urls import include, path
from app.views import *

urlpatterns = [
    path('admin/', admin.site.urls),
]
```

- 3.1.3. Na pasta 'app' você deve criar duas pastas, 'static' (onde deixaremos nossos arquivos estáticos como css, js e imagens) e 'templates' (onde trabalharemos nossas páginas html)





Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**



3.2 As models

Aqui, iremos definir nossos primeiros modelos, tabelas que serão criadas em banco de dados.

- 3.2.1. Dentro da pasta 'app', no arquivo 'models.py', definiremos as tabelas de Cidade, Autor, Gênero, Editora e Livro:

```
# Create your models here.
from django.db import models
class Cidade(models.Model):
    nome = models.CharField(max_length=100, verbose_name="Nome da cidade")
    uf = models.CharField(max_length=2, verbose_name="UF")
    def __str__(self):
        return f"{self.nome}, {self.uf}"
    class Meta:
        verbose_name = "Cidade"
        verbose_name_plural = "Cidades"

class Autor(models.Model):
    nome = models.CharField(max_length=100, verbose_name="Nome do autor")
    cidade = models.ForeignKey(Cidade, on_delete=models.CASCADE, verbose_name="Cidade do autor")
    def __str__(self):
        return self.nome
    class Meta:
        verbose_name = "Autor"
        verbose_name_plural = "Autores"
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



```
class Editora(models.Model):
    nome = models.CharField(max_length=100, verbose_name="Nome da
editora")
    site = models.CharField(max_length=100, verbose_name="Site da
editora")
    cidade = models.ForeignKey(Cidade, on_delete=models.CASCADE,
verbose_name="Cidade da editora")
    def __str__(self):
        return self.nome
    class Meta:
        verbose_name = "Editora"
        verbose_name_plural = "Editoras"

class Leitor(models.Model):
    nome = models.CharField(max_length=100, verbose_name="Nome do
leitor")
    email = models.CharField(max_length=100, verbose_name="Email do
leitor")
    cpf = models.CharField(max_length=11, unique=True,
verbose_name="CPF do leitor")
    def __str__(self):
        return self.nome
    class Meta:
        verbose_name = "Leitor"
        verbose_name_plural = "Leitores"

class Genero(models.Model):
    nome = models.CharField(max_length=100, verbose_name="Gênero")
    def __str__(self):
        return self.nome
    class Meta:
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



```
verbose_name = "Gênero"
verbose_name_plural = "Gêneros"

class Livro(models.Model):
    nome = models.CharField(max_length=100, verbose_name="Nome do livro")
    autor = models.ForeignKey(Autor, on_delete=models.CASCADE, verbose_name="Autor do livro")
    editora = models.ForeignKey(Editora, on_delete=models.CASCADE, verbose_name="Editora do livro")
    genero = models.ForeignKey(Genero, on_delete=models.CASCADE, verbose_name="Gênero do livro")
    preco = models.IntegerField(verbose_name="Preço do livro")
    data_plub = models.DateField(verbose_name="Data de publicação do livro")
    status = models.BooleanField(verbose_name="Status do livro")

    def __str__(self):
        return f'{self.nome}, {self.autor}'

    class Meta:
        verbose_name = "Livro"
        verbose_name_plural = "Livros"
```

3.2.2. Após **toda** model **criada ou modificada** é preciso que dê os seguintes comandos:

- **py manage.py makemigrations**
- **py manage.py migrate**



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



- 3.2.3. Vamos registrar estes **models** para serem exibidos no Ambiente de Administração do Django. Para tal, **acesse a pasta 'app' e o arquivo 'admin.py'**:

```
from django.contrib import admin
from .models import *
from django.contrib import admin

admin.site.register(Cidade)
admin.site.register(Autor)
admin.site.register(Editora)
admin.site.register(Leitor)
admin.site.register(Livro)
admin.site.register(Genero)
```

- 3.2.4. Agora com as models exibidas, lembre-se, **crie um superusuário** para si, relembre o comando: **py manage.py createsuperuser**.

```
Usuário (leave blank to use 'usuario'): admin
```

Pula o email.

LEBRANDO A SENHA NÃO APARECE CARACTERES, ENTÃO ATENTE-SE A SUA CRIAÇÃO.

```
Password:
Password (again):
Esta senha é muito curta. Ela precisa conter pelo menos 8 caracteres.
Esta senha é muito comum.
Esta senha é inteiramente numérica.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
```

Se a senha for muito curta, é só responder y.



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



3.2.5. Logue e acesse a existência das tabelas no **Django Admin**

| APP | | |
|----------|-------------|-------------|
| Autores | + Adicionar | ✎ Modificar |
| Cidades | + Adicionar | ✎ Modificar |
| Editoras | + Adicionar | ✎ Modificar |
| Gêneros | + Adicionar | ✎ Modificar |
| Leitores | + Adicionar | ✎ Modificar |
| Livros | + Adicionar | ✎ Modificar |

3.3 View de consulta

A nossa primeira view será a view 'Index', baseada em classe e responsável por exibir os livros cadastrados na nossa biblioteca:

3.3.1. Dentro da pasta '**app**', acesse o arquivo '**views.py**':

```
from django.shortcuts import render, redirect, get_object_or_404
from .models import *
from django.views import View
from django.contrib import messages

class IndexView(View):
    def get(self, request, *args, **kwargs):
        return render(request, 'index.html')
    def post(self, request):
        pass
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



- 3.3.2. Dentro da pasta **“config”**, acesse o arquivo **“urls.py”** e coloque o caminho da página **index**:

```
from django.contrib import admin
from django.urls import include, path
from app.views import *

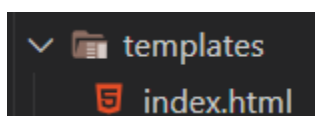
urlpatterns = [
    path('admin/', admin.site.urls),
    path('', TemplateView.as_view(template_name='index.html'),
    name='index'),
]
```

4 - Index

4.1 A página inicial

Agora nós iremos trabalhar com o index, primeira página exibida ao usuário acessar nosso sistema.

- 4.1.1. Ainda na pasta ‘app’, acesse a pasta ‘templates’ e crie o arquivo ‘index.html’:



- 4.1.2. Em ‘index.html’:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**

```
<meta name="viewport" content="width=device-width,  
initial-scale=1.0">  
<title>Página Inicial</title>  
<!-- Incluindo Bootstrap CSS -->  
<link rel="stylesheet"  
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/bootstrap.  
min.css">  
</head>  
<body>  
  <nav class="navbar navbar-expand-lg navbar-dark bg-success">  
    <a class="navbar-brand" href="/">Biblioteca</a>  
    <button class="navbar-toggler" type="button"  
data-toggle="collapse" data-target="#navbarNav"  
aria-controls="navbarNav" aria-expanded="false" aria-label="Toggle  
navigation">  
      <span class="navbar-toggler-icon"></span>  
    </button>  
    <div class="collapse navbar-collapse" id="navbarNav">  
      <ul class="navbar-nav">  
        <li class="nav-item">  
          <a class="nav-link text-dark" href="/">Home</a>  
        </li>  
        <li class="nav-item">  
          <a class="nav-link text-dark" href="/livros/">Livros</a>  
        </li>  
        <li class="nav-item">  
          <a class="nav-link text-dark" href="/reserva/">Reserva</a>  
        </li>  
        <li class="nav-item">  
          <a class="nav-link text-dark" href="/cidade/">Cidade</a>  
        </li>  
        <li class="nav-item">
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**



```
<a class="nav-link text-dark" href="/autor/">Autor</a>
</li>
<li class="nav-item">
  <a class="nav-link text-dark" href="/editor/">Editora</a>
</li>
<li class="nav-item">
  <a class="nav-link text-dark" href="/genero/">Genero</a>
</li>
<li class="nav-item">
  <a class="nav-link text-dark" href="/leitor/">Leitor</a>
</li>
<li class="nav-item">
  <a class="nav-link text-dark"
href="/admin/">Administração</a>
</li>
</ul>
</div>
</nav>

<div class="container mt-5">
  <div class="jumbotron text-center">
    <h1 class="display-4">Olá, é bom te ver por aqui</h1>
    <p class="lead">Esta é a página inicial da Biblioteca</p>
    <p class="lead">Seja Bem Vindo(A)</p>
  </div>
</div>

<!-- Incluindo jQuery e Bootstrap JS -->
<script
src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



```
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

4.2 O servidor de desenvolvimento

Nesta etapa, nós vamos conferir tudo que fizemos durante este tutorial inicial por meio de um servidor local.

- 4.2.1. Para gerar as migrações a serem aplicadas no banco de dados: **py manage.py makemigrations**
- 4.2.2. Para aplicar estas migrações no banco de dados: **py manage.py migrate**
- 4.2.3. Para criar um usuário administrador, capaz de acessar o Ambiente de Administração do Django: **py manage.py createsuperuser**
 - Usuário: admin
 - E-mail: <Enter>
 - Senha: 123 (não é exibida)
 - Senha (novamente): 123
 - Confirme com 'y'



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



4.2.4. Inicialize o servidor local: **py manage.py runserver**

```
System check identified no issues (0 silenced).  
February 25, 2024 - 13:24:38  
Django version 5.0.2, using settings 'config.settings'  
Starting development server at http://127.0.0.1:8000/  
Quit the server with CTRL-BREAK.
```

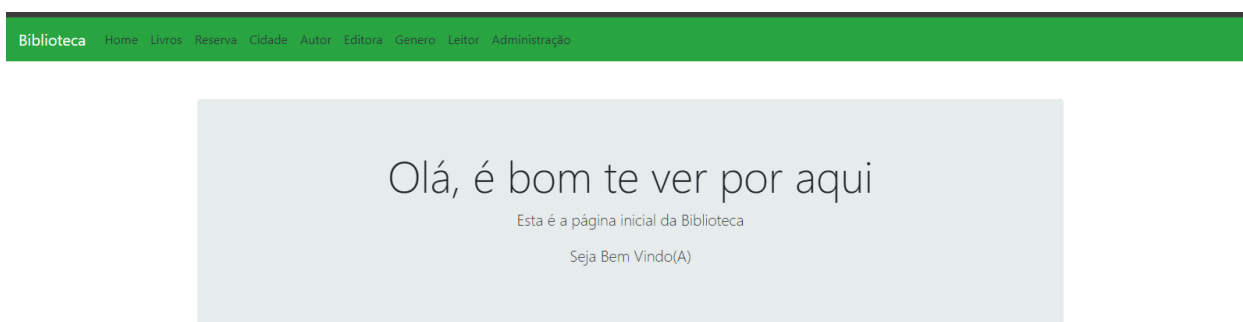


Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



- 4.2.5. Acessando a URL '<http://127.0.0.1:8000>' em seu navegador, você irá se deparar com o index :



- 4.2.6. Acesse o Ambiente de Administração do Django pela URL '<http://127.0.0.1:8000/admin/>' para registrarmos alguns objetos nas tabelas, adicione ao menos 5 de cada em todas as tabelas:

| APP | | |
|----------|-------------|-------------|
| Autores | + Adicionar | ✎ Modificar |
| Cidades | + Adicionar | ✎ Modificar |
| Editoras | + Adicionar | ✎ Modificar |
| Gêneros | + Adicionar | ✎ Modificar |
| Leitores | + Adicionar | ✎ Modificar |
| Livros | + Adicionar | ✎ Modificar |



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



5 Exibições Front End

5.1 Views

5.1.1. Vamos para **“app/views.py”** e vamos criar nossas views para as visualizações:

```
from django.shortcuts import
render, redirect, get_object_or_404
from .models import *
from django.views import View

class IndexView(View):
    def get(self, request, *args, **kwargs):
        return render(request, 'index.html')

class LivrosView(View):
    def get(self, request, *args, **kwargs):
        livros = Livro.objects.all()
        return render(request, 'livros.html', {'livros':
livros})
# def post(self, request, *args, **kwargs):

class EmprestimoView(View):
    def get(self, request, *args, **kwargs):
        reservas = Emprestimo.objects.all()
        return render(request, 'reserva.html',
{'reservas': reservas})
```




Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



```
class CidadesView(View):
    def get(self, request, *args, **kwargs):
        cidades = Cidade.objects.all()
        return render(request, 'cidade.html', {'cidades':
cidades})

class AutoresView(View):
    def get(self, request, *args, **kwargs):
        autores = Autor.objects.all()
        return render(request, 'autor.html', {'autores':
autores})

class EditorasView(View):
    def get(self, request, *args, **kwargs):
        editoras = Editora.objects.all()
        return render(request, 'editora.html',
{'editoras': editoras})

class LeitoresView(View):
    def get(self, request, *args, **kwargs):
        leitores = Leitor.objects.all()
        return render(request, 'leitor.html',
{'leitores': leitores})

class GenerosView(View):
    def get(self, request, *args, **kwargs):
        generos = Genero.objects.all()
        return render(request, 'genero.html', {'generos':
generos})
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



5.2 URLS

1. Vá para **“config/urls.py”**, vamos adicionar todas as urls para nossas páginas:

```
from django.contrib import admin
from django.urls import path
from django.views.generic import TemplateView
from app.views import *

urlpatterns = [
    path('admin/', admin.site.urls),
    path('',
TemplateView.as_view(template_name='index.html'),
name='index'),
    path('livros/', LivrosView.as_view(), name='livros'),
    path('reserva/', EmprestimoView.as_view(),
name='reserva'),
    path('cidade/', CidadesView.as_view(),
name='cidade'),
    path('autor/', AutoresView.as_view(), name='autor'),
    path('editor/', EditorasView.as_view(),
name='editora'),
    path('leitor/', LeitoresView.as_view(),
name='leitor'),
    path('genero/', GenerosView.as_view(),
name='genero'),
]
```

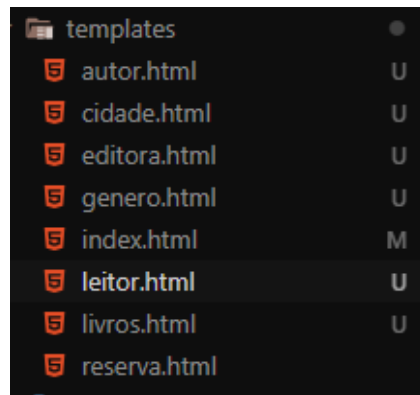
5.3 HTMLS

- 5.3.1. Dentro de **“app/templates”**, crie os arquivos



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**



5.3.2. Como base utilize **livros.html**:

```
<!DOCTYPE html>
<html lang="pt-BR">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width,
initial-scale=1.0">
  <title>Página Inicial</title>
  <!-- Incluindo Bootstrap CSS -->
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css
/bootstrap.min.css">
</head>
<body>
  <nav class="navbar navbar-expand-lg navbar-dark
bg-success">
    <a class="navbar-brand" href="/">Biblioteca</a>
    <button class="navbar-toggler" type="button"
data-toggle="collapse" data-target="#navbarNav"
aria-controls="navbarNav" aria-expanded="false"
aria-label="Toggle navigation">
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**



```
<span class="navbar-toggler-icon"></span>
</button>
    <div class="collapse navbar-collapse"
id="navbarNav">
    <ul class="navbar-nav">
        <li class="nav-item">
            <a class="nav-link text-dark"
href="/">Home</a>
        </li>
        <li class="nav-item">
            <a class="nav-link text-dark"
href="/livros/">Livros</a>
        </li>
        <li class="nav-item">
            <a class="nav-link text-dark"
href="/reserva/">Reserva</a>
        </li>
        <li class="nav-item">
            <a class="nav-link text-dark"
href="/cidade/">Cidade</a>
        </li>
        <li class="nav-item">
            <a class="nav-link text-dark"
href="/autor/">Autor</a>
        </li>
        <li class="nav-item">
            <a class="nav-link text-dark"
href="/editor/">Editora</a>
        </li>
        <li class="nav-item">
            <a class="nav-link text-dark"
href="/genero/">Genero</a>
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**



INSTITUTO
FEDERAL
Sul de Minas Gerais
Campus
Muzambinho

```
</li>
<li class="nav-item">
    <a class="nav-link text-dark"
href="/leitor/">Leitor</a>
</li>
<li class="nav-item">
    <a class="nav-link text-dark"
href="/admin/">Administração</a>
</li>
</ul>
</div>
</nav>

<div class="container mt-5">
<h2 class="mb-4">Tabela de Livros</h2>
<table class="table">
    <thead>
        <tr>
            <th scope="col">Nome</th>
            <th scope="col">Autor</th>
            <th scope="col">Editora</th>
            <th scope="col">Gênero</th>
            <th scope="col">Preço</th>
            <th scope="col">Data Publicação</th>
            <th scope="col">Status</th>
        </tr>
    </thead>
    <tbody>
        {% for livros in livros %}
        <tr>
            <td>{{ livros.nome }}</td>
            <td>{{ livros.autor }}</td>
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**



```
<td>{{ livros.editora }}</td>
<td>{{ livros.genero }} </td>
<td>{{ livros.preco }}</td>
<td>{{ livros.data_plub }}</td>
<td>{{ livros.status }}</td>
</tr>

{% endfor %}
</tbody>
</table>
</div>
<!-- Incluindo jQuery e Bootstrap JS -->
<script
src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></
script>
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"></script>
<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>
</body>
</html>
```

Utilizando esse html como base, coloque em todas as outras páginas, modificando o necessário para aparecer os dados das respectivas tabelas.



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento DJANGO WEB



INSTITUTO
FEDERAL
Sul de Minas Gerais
Campus
Muzambinho

6 Curiosidades(Não Obrigatório)

6.1 Delete

Já temos a função para visualizar, para cadastrar e então agora vamos criar a função de deletar.

View Delete

6.1.1. Dentro de “**views.py**”, vamos criar a view:

```
class DeleteLivroView(View):  
    def get(self, request, id, *args, **kwargs):  
        livro = Livro.objects.get(id=id)  
        livro.delete()  
        messages.success(request, 'Livro excluído com  
sucesso!') # Success message  
        return redirect('livros')
```

6.1.2. Agora vamos adicionar essa views dentro das urls:

```
from django.contrib import admin  
from django.urls import path  
from django.views.generic import TemplateView  
from app.views import *  
urlpatterns = [  
    path('admin/', admin.site.urls),  
    path('', ConsultaView.as_view(), name='livros'),  
    path('reserva/', ReservaView.as_view(),  
name='reserva'),  
    path('delete/<int:id>/', DeleteLivroView.as_view(),  
name='delete'),
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



```
]
```

6.2 Template Delete

Agora vamos adicionar o botão na tabela para podermos excluir o livro.

6.2.1. Já com o caminho feito, vá para **“index.html”** vai na tabela e como primeiro passo adicione uma tag

```
<table class="table">
  <thead>
    <tr>
      <th scope="col">Nome</th>
      <th scope="col">Autor</th>
      <th scope="col">Editora</th>
      <th scope="col">Gênero</th>
      <th scope="col">Preço</th>
      <th scope="col">Data Publicação</th>
      <th scope="col">Status</th>
      <th scope="col">Excluir</th>
    </tr>
  </thead>
```

6.2.2. e por fim antes da tabela adicione um forms para exclusão:

```
<tbody>
  {% for livros in livros %}
  <tr>
    <td>{{ livros.nome }}</td>
    <td>{{ livros.autor }}</td>
    <td>{{ livros.editora }}</td>
```




Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



```
        <td>
            {% for genero in livros.genero.all %}
            {{ genero.nome }}{% if not forloop.last %},
{% endif %}

            {% endfor %}
        </td>
        <td>{{ livros.preco }}</td>
        <td>{{ livros.data_plub }}</td>
        <td>{{ livros.status }}</td>
        <td><a href="delete/{{ livros.id }}"
class="btn btn-danger">Excluir</a></td>
    </tr>
    {% endfor %}
</tbody>
```

6.2.3. Recarregue a página **“index.html”**, exclua um livro para testar:



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



| | | | |
|------------|------|---------|---------------|
| Biblioteca | Home | Reserva | Administração |
|------------|------|---------|---------------|

Olá, é bom te ver por aqui

Esta é a página inicial da Biblioteca

Seja Bem Vindo(A)

Tabela de Livros

| Nome | Autor | Editora | Gênero | Preço | Data Publicação | Status | Excluir |
|-----------------------------|-------------|---------|-------------------------|-------|---------------------|--------|---------|
| Pai pobre, filho mais pobre | Lucas Silva | Riacho | Desenvolvimento Pessoal | 33 | 10 de Março de 2024 | True | Excluir |

6.2.4. Se feito corretamente deve aparecer essa mensagem no topo da tabela:

- Livro excluído com sucesso!

6.3 Editar Front End

6.4 View Edit

6.4.1. Dentro da pasta **“app/forms.py”** vamos criar um forms para a atualização:

```
class LivroForm(forms.ModelForm):
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



```
class Meta:
    model = Livro
    fields = '__all__'
    def __init__(self, *args, **kwargs):
        super(LivroForm, self).__init__(*args, **kwargs)
        self.helper = FormHelper()
        self.helper.layout = Layout(
            'titulo',
            'autor',
            'editora',
            'genero',
            'preco',
            'data_pub',
            'status',
            Submit('submit', 'Salvar')
        )
```

6.4.2. Dentro da pasta **“app/views.py”** crie a view de atualização:

```
class EditarLivroView(View):
    template_name = 'editar_livro.html'

    def get(self, request, id, *args, **kwargs):
        livro = get_object_or_404(Livro, id=id)
        form = LivroForm(instance=livro)
        return render(request, self.template_name, {'livro': livro,
        'form': form})

    def post(self, request, id, *args, **kwargs):
        livro = get_object_or_404(Livro, id=id)
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**



```
form = LivroForm(request.POST, instance=livro)
if form.is_valid():
    form.save()
    messages.success(request, 'As edições foram salvas com
sucesso.')
    return redirect('editar', id=id) # Redirecionar de
volta para a página de edição
else:
    messages.error(request, 'Corrija os erros no formulário
antes de enviar novamente.')

    return render(request, self.template_name, {'livro': livro,
'form': form})
```

6.4.3. Vá para **“app/urls.py”**, vamos adicionar o caminho da view e do id:

```
path('editar/<int:id>/', EditarLivroView.as_view(), name='editar'),
```

6.5 Template Edit

6.5.1. Vá para **“app/templates”** e no arquivo **index.html**, adicione campos dentro da tabela. Primeiro adicione na **thead**:

```
<th scope="col">Editar</th>
```

6.5.2. Segundo já dentro do corpo da tabela adicione o botão que vai nos levar para a página de edição:



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



INSTITUTO
FEDERAL

Sul de Minas Gerais

Campus
Muzambinho

```
<td><a href="editar/{{ livros.id }}" class="btn  
btn-warning">Editar</a></td>
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



6.5.3. Acesse novamente <http://127.0.0.1:8000/> e veja como está, deve estar assim:

[Biblioteca](#) [Home](#) [Reserva](#) [Administração](#)

Olá, é bom te ver por aqui

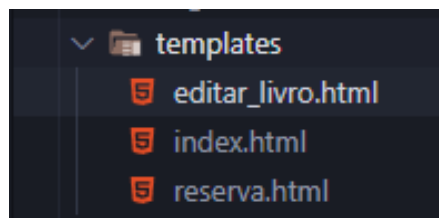
Esta é a página inicial da Biblioteca

Seja Bem Vindo(A)

Tabela de Livros

| Nome | Autor | Editora | Gênero | Preço | Data Publicação | Status | Excluir | Editar |
|-----------------------------|-------------|---------|-------------------------|-------|---------------------|--------|--------------------------|-------------------------|
| Pai pobre, filho mais pobre | Lucas Silva | Riacho | Desenvolvimento Pessoal | 32 | 10 de Março de 2024 | True | <button>Excluir</button> | <button>Editar</button> |

6.5.4. Por fim dentro da pasta **“app/templates”** crie um arquivo html chamado **“editar_livro.html”**.





Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**



6.5.5. Dentro desse html, adicione o seguinte código que vai nos permitir visualizar e editar, e usaremos já o forms feito em forms.py para a edição.

```
{% load crispy_forms_tags %}
<!DOCTYPE html>
<html lang="pt-BR">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width,
initial-scale=1.0">
    <title>Detalhes e Edição do Livro</title>

    <!-- Incluindo Bootstrap CSS -->
    <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/css/boot
strap.min.css">

    <!-- Incluindo estilos do crispy_forms -->

    <!-- Estilos personalizados (se houver) -->
    <!-- Adicione links para seus próprios estilos aqui, se
necessário -->
</head>
<body>
<div class="container mt-5">
    <a href="/" class="btn btn-warning">Voltar para a Home</a>

    <h2>Detalhes do Livro: {{ livro.nome }}</h2>

    <p>Nome: {{ livro.nome }}</p>
    <p>Autor: {{ livro.autor }}</p>
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**



INSTITUTO
FEDERAL
Sul de Minas Gerais
Campus
Muzambinho

```
<p>Editora: {{ livro.editora }}</p>
    <p>Gênero: {% for genero in livro.genero.all %}{{
genero.nome }}{% if not forloop.last %}, {% endif %}{% endfor
%}</p>
    <p>Preço: R$ {{ livro.preco }}</p>
    <p>Data de Publicação: {{ livro.data_plub }}</p>
    <p>Status: {{ livro.status }}</p>

<h3 class="mt-4">Editar Livro</h3>
{% if messages %}
    <ul class="messages">
        {% for message in messages %}
            <b><li{% if message.tags %} class="{ { message.tags
}}"{% endif %} style="color: red;">{{ message }}</li></b>
        {% endfor %}
    </ul>
{% endif %}

<form method="post">
    {% csrf_token %}
    {{ form|crispy }}
    <button type="submit" class="btn btn-success btn-block
btn-lg mt-2">Salvar Edições do Livro</button>

</form>
<br>
</div>

<!-- Incluindo jQuery e Bootstrap JS -->
<script
src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></scrip
t>
```




Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento **DJANGO WEB**



INSTITUTO
FEDERAL

Sul de Minas Gerais

Campus
Muzambinho

```
<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"></script>

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>
</html>

<!-- Incluindo jQuery e Bootstrap JS -->

<script
src="https://code.jquery.com/jquery-3.5.1.slim.min.js"></script>

<script
src="https://cdn.jsdelivr.net/npm/@popperjs/core@2.9.1/dist/umd/popper.min.js"></script>

<script
src="https://maxcdn.bootstrapcdn.com/bootstrap/4.5.2/js/bootstrap.min.js"></script>

</body>
</html>
```



Programa Campus Inteligente

Iniciação Educacional no Desenvolvimento *DJANGO WEB*



- 6.5.6. Agora ao clicar no botão de **editar** no livro respectivo cadastrado por você, ele por meio de uma requisição de id, vai puxar os dados e permitir nossa edição. Retornando ao <http://127.0.0.1:8000/> e botão editar, abrirá essa página com as informações de **seu livro cadastrado se divirta:**

[Voltar para a Home](#)

Detalhes do Livro: Pai pobre, filho mais pobre

Nome: Pai pobre, filho mais pobre

Autor: Lucas Silva

Editora: Riacho

Gênero: Desenvolvimento Pessoal

Preço: R\$ 32

Data de Publicação: 10 de Março de 2024

Status: True

Editar Livro

Nome do livro*

Autor do livro*

Editora do livro*

Gênero do livro*

Romance

Suspense

Drama

Desenvolvimento Pessoal

Preço do livro*

Data de publicação do livro*

☒ Status do livro

Salvar Edições do Livro



Programa Campus Inteligente
Iniciação Educacional
no Desenvolvimento
DJANGO WEB



**INSTITUTO
FEDERAL**

Sul de Minas Gerais

Campus
Muzambinho

