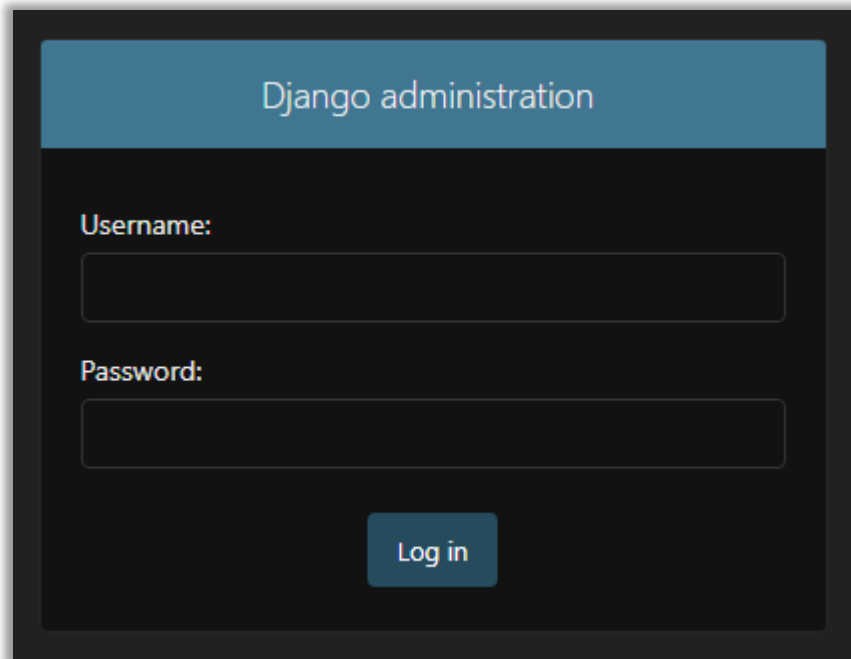


Aula 12

Django

Inserindo Dados

Agora que temos as tabelas criadas, podemos inserir dados nelas. Para isso, temos que acessar a página administrativa do Django através da url <http://127.0.0.1:8000/admin> (ou <http://localhost:8000/admin>).



O problema é que ainda não temos como acessar, pois, ainda não dispomos de um login administrativo.

Para resolver isso, temos que ir ao terminal e executar o comando abaixo:

```
(.venv) C:\Users\augusto.hertzog\Desktop\curso>python manage.py createsuperuser
Username (leave blank to use 'augusto.hertzog'): agosto
Email address: agosto@hertzog.com
Password:
Password (again):
This password is too short. It must contain at least 8 characters.
This password is too common.
This password is entirely numeric.
Bypass password validation and create user anyway? [y/N]: y
Superuser created successfully.
(.venv) C:\Users\augusto.hertzog\Desktop\curso>
```

O comando acima, chamado através do arquivo manage.py, vai executar um terminal interativo.

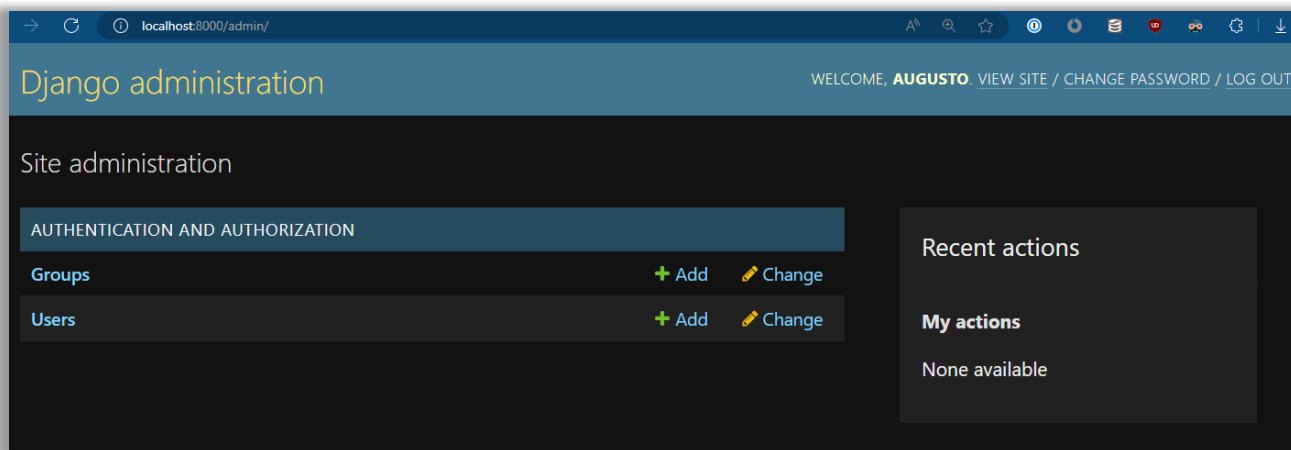
Primeiro, vai ser pedido para criar um nome de usuário.

Depois, vai ser pedido para inserir um e-mail. Como estamos em um ambiente de desenvolvimento, não tem problema usarmos um e-mail falso. Mas, em um ambiente de produção, tem que ser usado um e-mail válido, até porque ele será usado para recuperar a conta caso tenha qualquer problema.

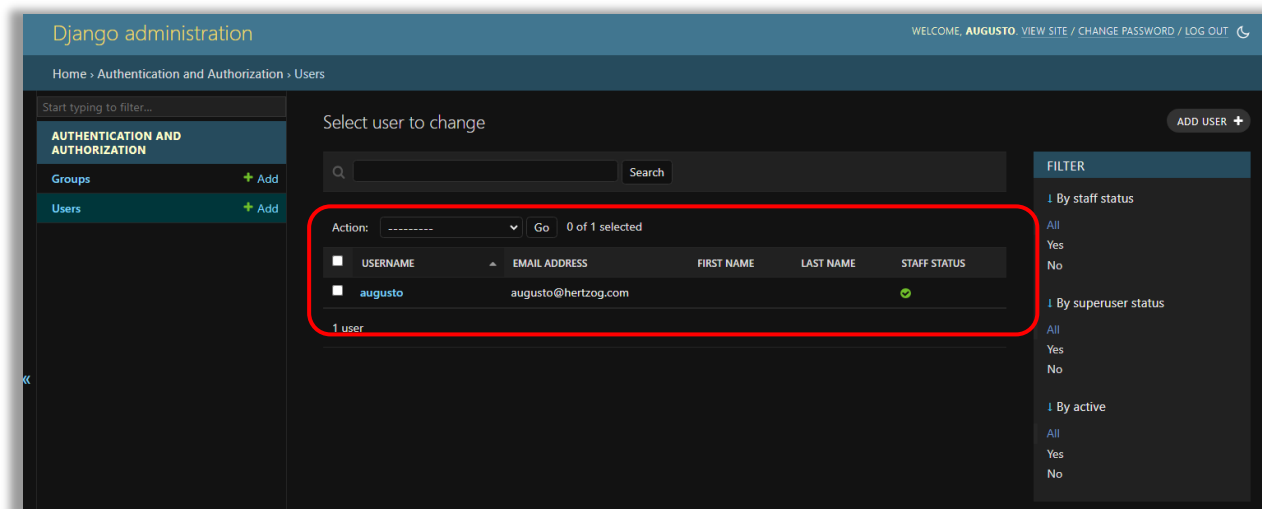
Por fim, temos que escolher uma senha para o usuário. Para essa senha, é fortemente recomendado seguir as recomendações de segurança para a criação dela (destacadas em texto vermelho). Novamente, sendo um ambiente de ensino, não tem problema usarmos uma senha fraca e de mais fácil memorização.

Agora, já podemos entrar com esse login na tela administrativa.

Abaixo, temos a tela de início:



Explorando os links, podemos ver páginas para grupos e usuários. Lá podemos adicionar grupos e usuários. Inclusive, na página de usuário podemos ver o usuário que acabamos de criar no terminal:

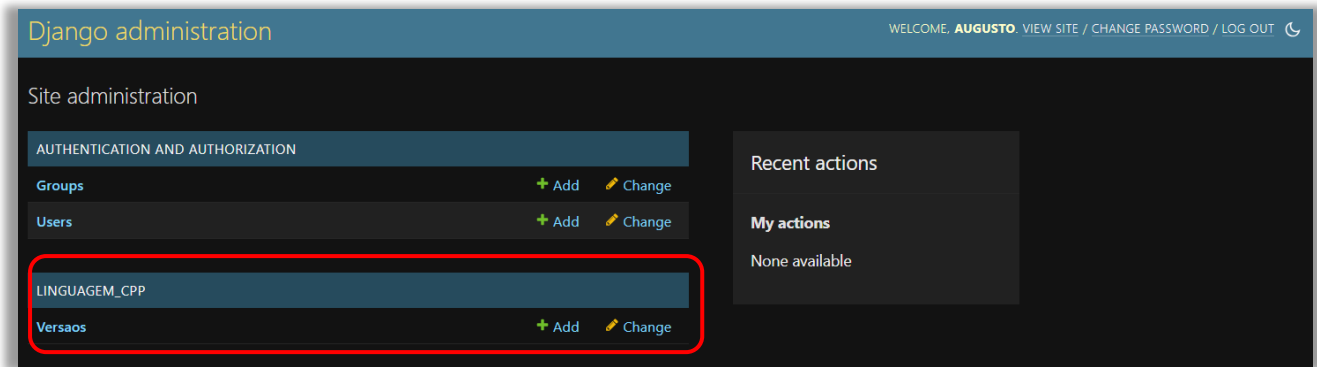


Depois de explorar as páginas, podemos ver que estão faltando os modelos que criamos para nosso projeto. Nesse caso, as versões de lançamento da nossa linguagem de programação.

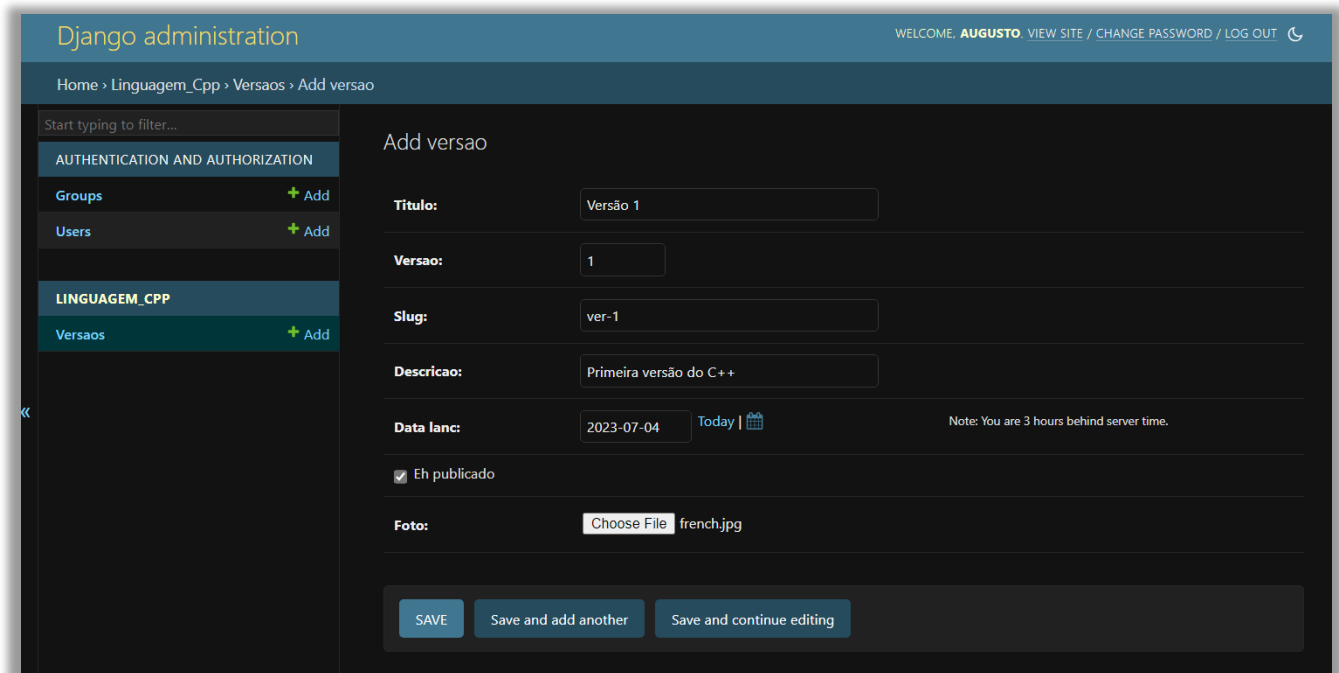
Para eles aparecerem lá, temos que alterar o arquivo `/linguagem_cpp/admin.py`. Originalmente ele está vazio, então vamos adicionar o seguinte código:

```
C:\Users\augusto.hertzog\Desktop\curso\linguagem_cpp\admin.py
1 from django.contrib import admin
2 from . import models
3
4 class VersaoAdmin(admin.ModelAdmin):
5     # adicionaremos código aqui mais tarde
6     ...
7
8 admin.site.register(Versao, VersaoAdmin)
9
```

Agora, quando vamos na página administrativa principal e podemos ver um novo tópico:

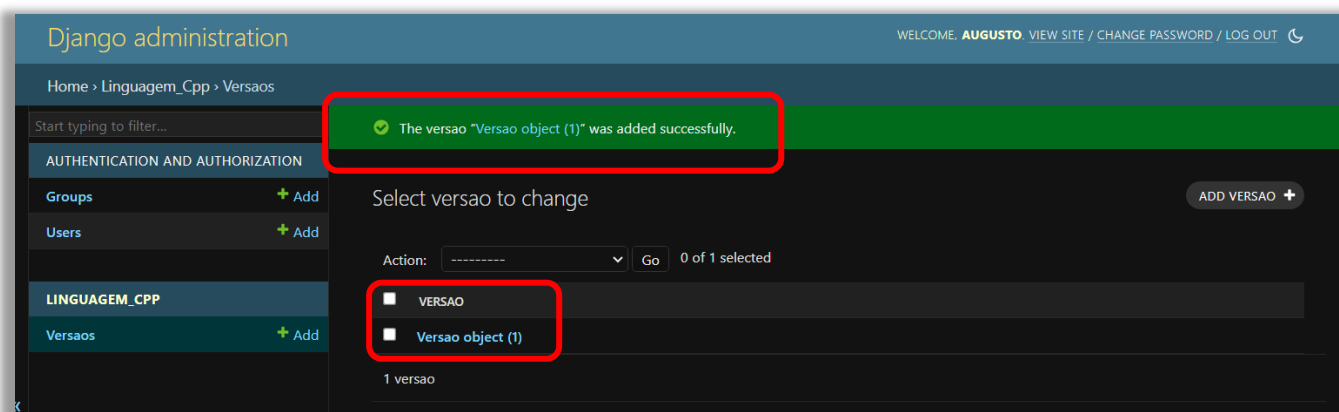


E, finalmente, já podemos adicionar os dados do aplicativo. Veja um modelo:



Para salvar tudo, basta irmos no botão abaixo SAVE.

Veja que, ao salvarmos, aparece um nome bem esquisito no destaque em verde.

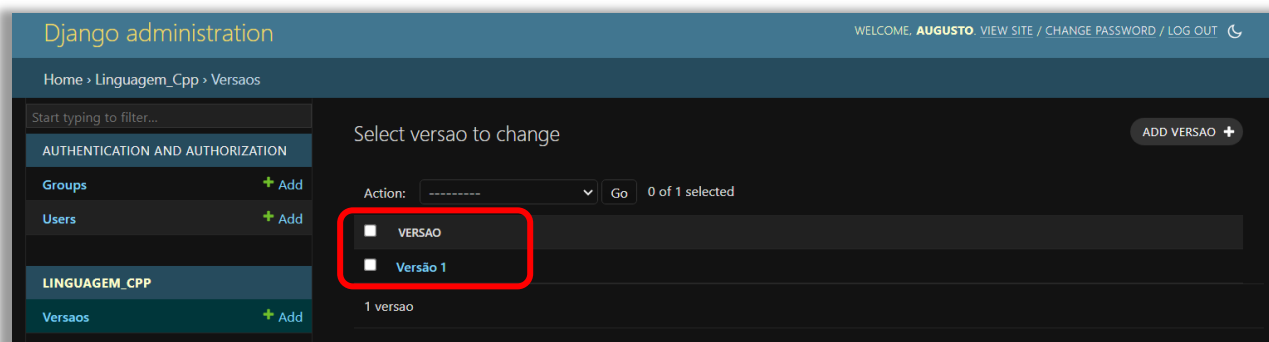


Isso acontece porque a nossa classe não tem um método mágico `__str__` definido. Para arrumar isso, basta adicionarmos o método na classe.

Veja como ficará:

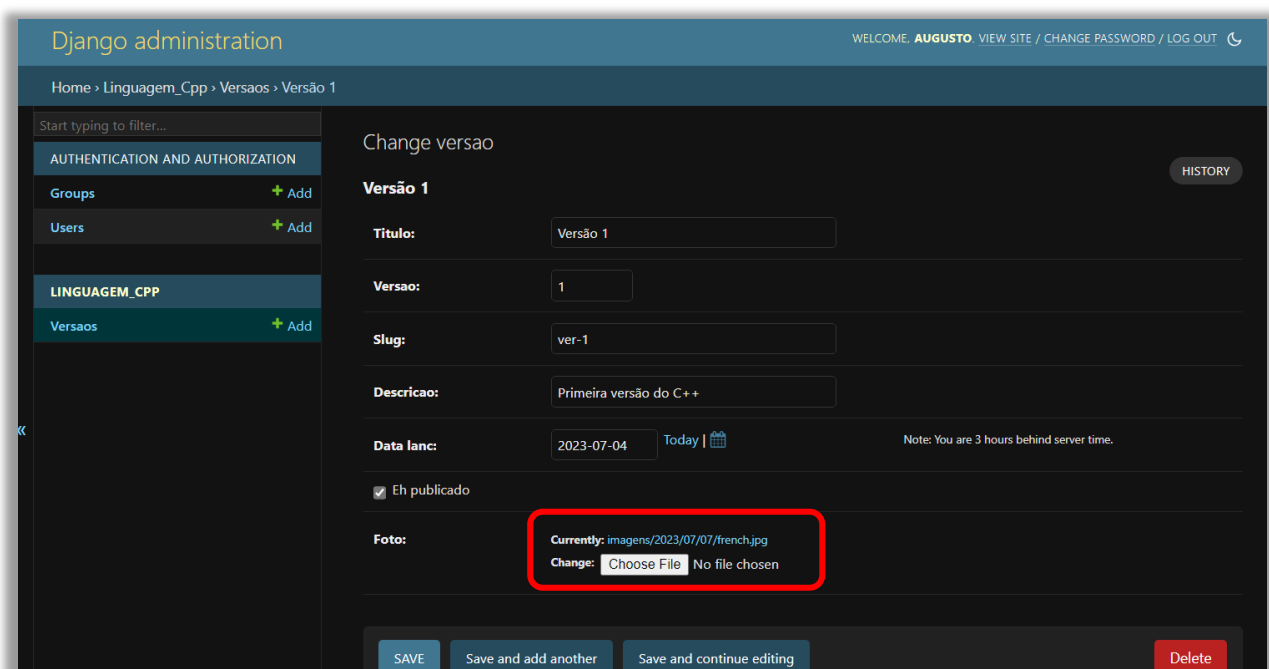
```
C:\Users\augusto.hertzog\Desktop\curso\linguagem_cpp\models.py
1 from django.db import models
2
3 class Versao(models.Model):
4     titulo = models.CharField(max_length=100)
5     versao = models.IntegerField()
6     slug = models.SlugField()
7     descricao = models.CharField(max_length=200)
8     data_lanc = models.DateField()
9     eh_publicado = models.BooleanField(default=False)
10    foto = models.ImageField(upload_to='imagens/%Y/%m/%m')
11
12    def __str__(self):
13        return self.titulo
14
```

Agora, ao salvarmos, será apresentado o nome correto.

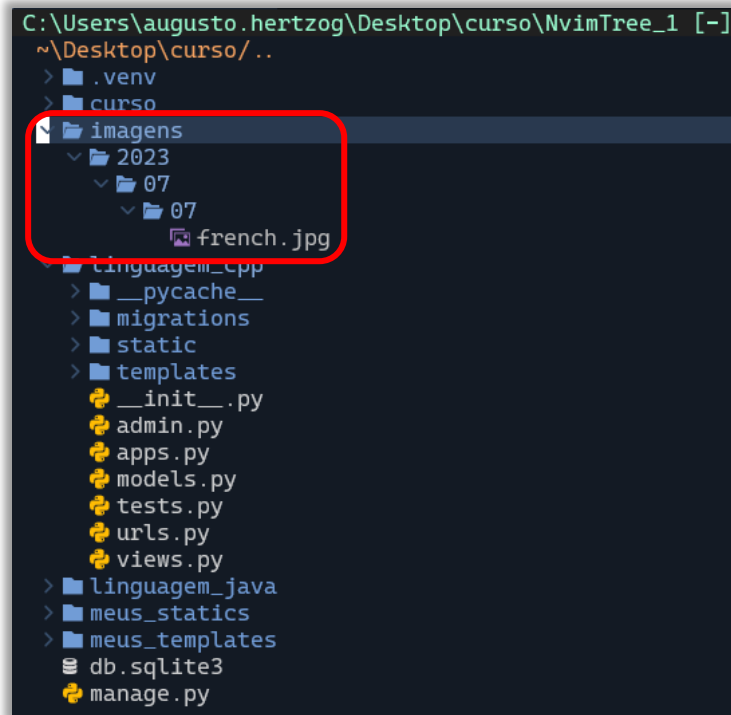


Dessa forma, podemos adicionar, alterar, atualizar e excluir os dados para nosso site sem a necessidade de usar a biblioteca Faker do Python.

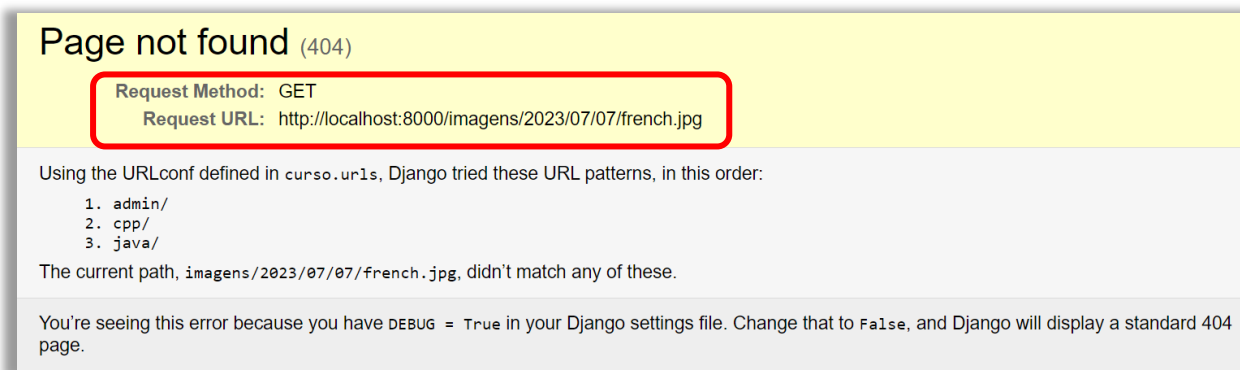
Se inspecionarmos a Versão 1, podemos alterar seus dados. Se observar, temos o caminho onde a imagem carregada foi salva:



Se olharmos a pasta do nosso projeto, vamos ver a pasta criada e todas as suas subpastas, que nesse caso foi dividida em ano, mês e dia:



Mas ainda tem um problema. Se tentarmos acessar o link da imagem, vamos ter uma página inexistente:



Isso acontece porque não registramos essa pasta de imagens nas configurações do nosso projeto.

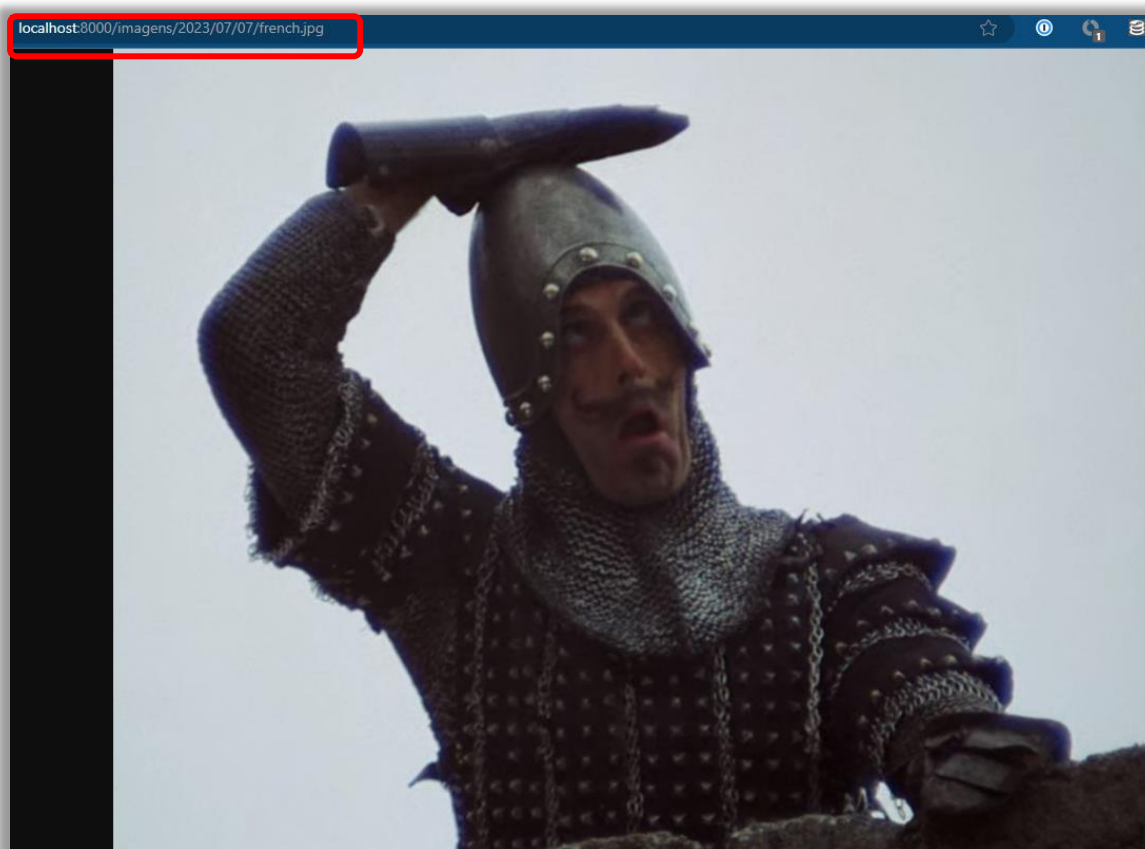
Para arrumar isso, vamos adicionar o caminho da pasta no nosso arquivo `/curso/settings.py` (mais ou menos como fizemos com a pasta global de nossos arquivos estáticos):

```
120 # Static files (CSS, JavaScript, Images)
121 # https://docs.djangoproject.com/en/4.2/howto/static-files/
122
123 STATIC_URL = 'static/'
124 STATICFILES_DIRS = [
125     BASE_DIR / 'meus_statics',
126 ]
127
128 MEDIA_URL = '/imagens'
129 MEDIA_ROOT = BASE_DIR / 'imagens'
130
```

Depois, adicionamos o seguinte no arquivo **/curso/urls.py** (repare que o arquivo urls.py é o arquivo do projeto, e não dos aplicativos):

```
C:\Users\augusto.hertzog\Desktop\curso\curso\urls.py
1 from django.contrib import admin
2 from django.urls import path, include
3 from django.conf.urls.static import static
4 from django.conf import settings
5
6 urlpatterns = [
7     path('admin/', admin.site.urls),
8     path('cpp/', include('linguagem_cpp.urls')),
9     path('java/', include('linguagem_java.urls')),
10 ]
11
12 urlpatterns += static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)
13
```

Agora, finalmente, nossa imagem será carregada:



Exercícios para Praticar

1. Use o que foi aprendido em aula e aplique nos seus projetos.

Atividade para Entregar

1. Requisitos para a entrega final:
 - a. seu projeto tem que ter, ao menos, dois aplicativos de sua escolha e um aplicativo para servir de home para todo seu projeto;
 - b. ter um menu de navegação entre TODOS os aplicativos;
 - c. usar a organização de arquivos nas pastas templates (HTMLs) e static (CSSs, JSs e imagens) dentro de cada aplicativo;
 - d. usar a organização de arquivos nas pastas templates (HTMLs) e static (CSSs, JSs e imagens) global;
 - e. uso das tags Django if, elif, else, for, include, url, static, block, extends;
 - f. criação de classes com os campos, ao menos um de cada listado abaixo:
 - i. charfield
 - ii. integerfield
 - iii. slugfield
 - iv. datefield
 - v. booleanfield
 - vi. imagefield
 - g. usar o banco de dados para guardar os dados inseridos pelo administrador;
 - h. o administrador do projeto deve ter senha e usuário "admin";
 - i. usar imagens, css e js nas páginas;
2. O seu site deve ter uma interface viável;
 - a. ter css, js, imagens e um conteúdo;
 - b. Dica: use as grades do bootstrap para organizar rapidamente o geral do site (não obrigatório);
 - i. o uso do bootstrap não exclui o uso de CSS próprio;