File upload

Configure os arquivos de media

```
MEDIA_ROOT = os.path.join(BASE_DIR, 'media') MEDIA_URL = "/media/"
```

Adicione uma URL para os arquivos de media

```
from django.conf import settings from django.conf.urls.static import
static urlpatterns += static(settings.MEDIA_URL,
document_root=settings.MEDIA_ROOT)
```

Crie o formulário

```
<form action="{% url 'home' %}" method="POST" enctype="multipart/form-
data">{% csrf_token %} <input type="file" name="my_file"> <input
type="submit" value="enviar"> </form>
```

Receba os dados

```
def home(request): if request.method == "GET": return render(request,
'home.html') elif request.method == "POST": file = request.FILES
print(file) return HttpResponse('Teste')
```

Configure o tamanho máximo do InMemory

```
FILE_UPLOAD_MAX_MEMORY_SIZE = 600000000
```

Salvando a imagem com pillow

```
def home(request): if request.method == "GET": return render(request,
'home.html') elif request.method == "POST": file =
request.FILES.get('my_file') img = Image.open(file) path =
os.path.join(settings.BASE_DIR, f'media/{file.name}') img =
img.save(path) print(file) return HttpResponse('Teste')
```

Crie a model

```
class MyFile(models.Model): title = models.CharField(max_length=20) arq =
models.FileField(upload_to='img') def __str__(self) -> str: return
self.title
```

Salve os arquivos utilizando as models do Django

```
def home(request): if request.method == "GET": return render(request,
'home.html') elif request.method == "POST": title =
request.POST.get('title') file = request.FILES.get('my_file') mf =
MyFile(title=title, arq=file) mf.save() return HttpResponse('Teste')
```

Retorna os bytes do arquivo

```
file.file
```

Retorna o nome do arquivo

```
file.name
```

Retorna o tamanho do arquivo em bytes

```
file.size
```

Gera os bytes através de chunks

```
file.chunks()
```

Editando a imagem antes de salvar

```
name = f'{date.today()}-{title}.jpg' img = Image.open(file) img =
img.convert('RGB') img = img.resize((300, 300)) draw =
ImageDraw.Draw(img) draw.text((20, 280), f"Marca d'agua {date.today()}",
(255, 255, 255)) output = BytesIO() img.save(output, format="JPEG",
quality=100) output.seek(0) img_final = InMemoryUploadedFile(output,
'ImageField', name, 'image/jpeg', sys.getsizeof(output), None ) mf =
MyFile(title=title, arq=img_final) mf.save()
```