

Informe de Laboratorio 07

Tema: Django 03

Nota

Estudiante	Escuela	Asignatura
Jhastyn Jefferson Payehuancar Riquelme jpayehuancar@unsa.edu.pe	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: III Código: 20222064

Laboratorio	Tema	Duración
07	Django 03	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2023 - A	Del 04 Julio 2023	Al 14 Julio 2023

1. Tarea

Reproducir las actividades de los videos donde trabajamos:

1. Relación de uno a muchos
2. Relación muchos a muchos
3. Impresión de pdfs
4. Envío de emails
5. Crear su video Flipgrid

2. URL GitHub/Video

- URL del repositorio GitHub.
- <https://github.com/Jh4stYn/Lab07.git>
- URL del Video.
- <https://flip.com/s/GzyVNPBYNt1Q>

3. Relación de uno a muchos - Relación muchos a muchos

Creamos los modelos para poder trabajar

Listing 1: Models.py

```
1 from django.db import models
2
3 class Simple(models.Model):
4     text = models.CharField(max_length=10)
5     number = models.IntegerField(null=True)
6     url = models.URLField(default='www.example.com')
7
8     def __str__(self):
9         return self.url
10
11 class DateExample(models.Model):
12     the_date = models.DateTimeField()
13
14 class NullExample(models.Model):
15     col = models.CharField(max_length=10, blank=True, null=True)
16
17 class Language(models.Model):
18     name = models.CharField(max_length=10)
19
20     def __str__(self):
21         return self.name
22
23 class Framework(models.Model):
24     name = models.CharField(max_length=10)
25     language = models.ForeignKey(Language, on_delete=models.CASCADE)
26
27     def __str__(self):
28         return self.name
29
30 class Movie(models.Model):
31     name = models.CharField(max_length=10)
32
33     def __str__(self):
34         return self.name
35
36 class Character(models.Model):
37     name = models.CharField(max_length=10)
38     movies = models.ManyToManyField(Movie)
39
40     def __str__(self):
41         return self.name
```

model_examples > db.sqlite3

Search tables... Reset Filters Records: 2

Tables (19)	id	name
django_migrations		
sqlite_sequence		
auth_group_permissions		
auth_user_groups		
auth_user_user_permissions		
django_admin_log		
django_content_type		
auth_permission		
auth_group		
auth_user		
django_session		
example_dateexample		
example_language		
example_nullexample		
example_simple		
example_framework		
example_movie		
example_character		
example_character_movies		

model_examples > db.sqlite3

Search tables... Reset Filters Records: 4

Tables (19)	id	name	language_id
django_migrations			
sqlite_sequence			
auth_group_permissions			
auth_user_groups			
auth_user_user_permissions			
django_admin_log			
django_content_type			
auth_permission			
auth_group			
auth_user			
django_session			
example_dateexample			
example_language			
example_nullexample			
example_simple			
example_framework			
example_movie			
example_character			
example_character_movies			

model_examples > db.sqlite3		Reset Filters	Records: 4
Search tables...	id	name	
Tables (19)	Search column...	Search column...	
> django_migrations	1	1	Avengers
> sqlite_sequence	2	2	Civil War
> auth_group_permissions	3	3	Thor : Dark World
> auth_user_groups	4	4	Winter Soldier
> auth_user_user_permissions			
> django_admin_log			
> django_content_type			
> auth_permission			
> auth_group			
> auth_user			
> django_session			
> example_dateexample			
> example_language			
> example_nullexample			
> example_simple			
> example_framework			
> example_movie			
> example_character			
> example_character_movies			

model_examples > db.sqlite3		Reset Filters	Records: 2
Search tables...	id	name	
Tables (19)	Search column...	Search column...	
> django_migrations	1	1	Captain America
> sqlite_sequence	2	2	Thor
> auth_group_permissions			
> auth_user_groups			
> auth_user_user_permissions			
> django_admin_log			
> django_content_type			
> auth_permission			
> auth_group			
> auth_user			
> django_session			
> example_dateexample			
> example_language			
> example_nullexample			
> example_simple			
> example_framework			
> example_movie			
> example_character			
> example_character_movies			

model_examples > db.sqlite3

Search tables... Reset Filters Records: 5

Tables (19)	id	character_id	movie_id
django_migrations	1	1	1
sqlite_sequence	2	2	2
auth_group_permissions	3	3	2
auth_user_groups	4	4	3
auth_user_user_permissions	5	5	1
django_admin_log			
django_content_type			
auth_permission			
auth_group			
auth_user			
django_session			
example_dateexample			
example_language			
example_nullexample			
example_simple			
example_framework			
example_movie			
example_character			
example_character_movies			

4. IMPRESION DE PDFs

Se logró replicar el código siendo estos los más importantes

[-] utils.py, es el encargado de renderizar el pdf, transformando el Html en un PDF.

Listing 2: Utils.py

```
1 from io import BytesIO
2 from django.http import HttpResponse
3 from django.template.loader import get_template
4
5 from xhtml2pdf import pisa
6
7 def render_to_pdf(template_src, context_dict={}):
8     template = get_template(template_src)
9     html = template.render(context_dict)
10    result = BytesIO()
11    pdf = pisa.pisaDocument(BytesIO(html.encode("ISO-8859-1")), result)
12    if not pdf.err:
13        return HttpResponse(result.getvalue(), content_type='application/pdf')
14    return None
```

[-] views.py, este es el encargado de generar el pdf, dando el nombre y la información necesario al template para que esté completo.

Listing 3: Views.py

```
1 from django.http import HttpResponse
2 from django.views.generic import View
3
4 from django.template.loader import get_template
5 from .utils import render_to_pdf
6
7 class GeneratePdf(View):
8     def get(self, request, *args, **kwargs):
9         template = get_template('pdf/invoice.html')
```

```

10     context = {
11         'invoice_id': 123,
12         'customer_name': 'John Carter',
13         'amount': 139.99,
14         'today': 'Today',
15     }
16     html = template.render(context)
17     pdf = render_to_pdf('pdf/invoice.html', context)
18     if pdf:
19         response = HttpResponse(pdf, content_type='application/pdf')
20         filename = "Invoice_%s.pdf" %("12341231")
21         content = "inline; filename=%s" %(filename)
22         download = request.GET.get("download")
23         if download:
24             content = "attachment; filename='%s'" %(filename)
25         print(content)
26         response['Content-Disposition'] = content
27         return response
28     return HttpResponse("Not found")
29
30 # def generate_view(request, *args, **kwargs):
31 #     template = get_template('pdf/invoice.html')
32 #     context = {
33 #         'invoice_id': 123,
34 #         'customer_name': 'John Carter',
35 #         'amount': 139.99,
36 #         'today': 'Today',
37 #     }
38 #     html = template.render(context)
39 #     return HttpResponse(html)

```

[–] invoice.html, es lo que recibirá información de la Vista y se transformará en PDF

Listing 4: Invoice.html

```

1  <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
2    "http://www.w3.org/TR/html4/loose.dtd">
3  <html>
4    <head>
5      <title>Title</title>
6      <style type="text/css">
7        body {
8          font-weight: 200;
9          font-size: 14px;
10        }
11        .header {
12          font-size: 20px;
13          font-weight: 100;
14          text-align: center;
15          color: #007cae;
16        }
17        .title {
18          font-size: 22px;
19          font-weight: 100;
20          /* text-align: right;*/
21          padding: 10px 20px 0px 20px;

```

```
21     }
22     .title span {
23         color: #007cae;
24     }
25     .details {
26         padding: 10px 20px 0px 20px;
27         text-align: left !important;
28         /*margin-left: 40%;*/
29     }
30     .hrItem {
31         border: none;
32         height: 1px;
33         /* Set the hr color */
34         color: #333; /* old IE */
35         background-color: #fff; /* Modern Browsers */
36     }
37 </style>
38 </head>
39 <body>
40     <div class='wrapper'>
41         <div class='header'>
42             <p class='title'>Invoice # {{ invoice_id }} </p>
43         </div>
44         <div>
45             <div class='details'>
46                 Bill to: {{ customer_name }} <br/>
47                 Amount: {{ amount }} <br/>
48                 Date: {{ today }}
49                 <hr class='hrItem' />
50             </div>
51         </div>
52     </body>
53 </html>
```

Para poder ejecutar este ejercicio, necesitaremos escribir el siguiente comando en la ubicación del 'manage.py':

-

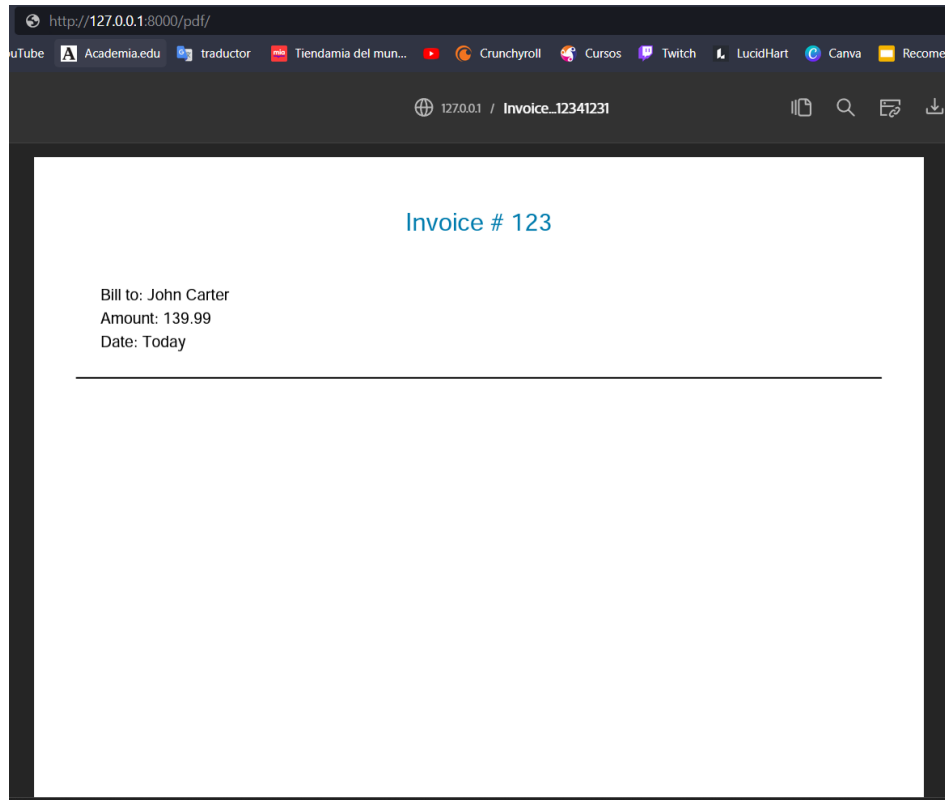
```
python .\manage.py runserver
```

Una vez hayamos ejecutado el comando, se encenderá nuestro servidor, el cual podremos abrir ingresando a la siguiente dirección:

(<http://127.0.0.1:8000/pdf/>)

Una vez dentro podremos visualizar todo el contenido de nuestro PDF, el cuál podremos descargar sin ningún problema.

Se recuerda que el contenido es totalmente modificable, tan solo tendremos que editar nuestro archivo 'invoice.html a nuestra conveniencia'



5. ENVÍO DE EMAILS

[–] views.py, aquí debemos colocar el Asunto, el Cuerpo, el Correo con el que se enviará y el/los correo(s) a donde llegará(destinatario). Para el ejemplo se usará correos temporales como destinatarios.

Listing 5: Views.py

```
1 from django.shortcuts import render
2 from django.core.mail import send_mail
3
4 # Create your views here.
5 def index(request):
6
7     send_mail('Hello from Django',
8             'Hello there. This is an automated message.',
9             'jhastyn57@gmail.com',
10            ['jpayehuancar@unsa.edu.pe'],
11            fail_silently=False)
12
13     return render(request, 'send/index.html')
```

[–] settings.py, aquí debemos configurar el servicio SMTP del cual haremos uso para el envío de correo, en el que debemos de colocar, dominio, puerto y el correo(user) junto con la contraseña con la que se enviará el correo, es importante que estos sean correctos. Algo importante para que funcione con el servicio SMTP de Gmail, es habilitar el acceso de aplicaciones menos seguras en

tu cuenta de Gmail. Puedes hacerlo siguiendo estas instrucciones: <https://support.google.com/accounts/answer/6010255>

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

```
EMAIL_HOST = 'smtp.gmail.com'  
EMAIL_PORT = 587  
EMAIL_HOST_USER = 'jhastyn57@gmail.com'  
EMAIL_HOST_PASSWORD = 'apps password'  
EMAIL_USE_TLS = True  
EMAIL_USE_SSL = False
```

[–] index.html, si todo va bien, esta página se cargará correctamente:

Listing 6: Views.py

```
1 <!DOCTYPE html>  
2 <html lang="en">  
3 <head>  
4     <meta charset="UTF-8">  
5     <meta name="viewport" content="width=device-width, initial-scale=1.0">  
6     <title>Sent!/title>  
7 </head>  
8 <body>  
9     <h1>Sent an email!</h1>>  
10 </body>  
11 </html>
```

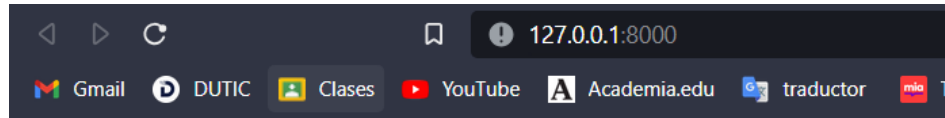
-

```
python .\manage.py runserver
```

Una vez hayamos ejecutado el comando, se encenderá nuestro servidor, el cual podremos abrir ingresando a la siguiente dirección:

(<http://127.0.0.1:8000>)

Una vez dentro podremos visualizar el mensaje 'Sent an email!', que nos indica que el email ya fue enviado a nuestro correo



Sent an email!

>

Email enviado

Hello from Django



jhastyn57@gmail.com

para jpayehuancar ▾

Hello there. This is an automated message.



Email recibido

Hello from Django Externo Recibidos x



jhastyn57@gmail.com

para mí ▾

🌐 inglés ▾ > español ▾ [Traducir mensaje](#)

Hello there. This is an automated message.



Estructura de laboratorio 07

- El contenido que se entrega en este laboratorio es el siguiente:

```
lab07/
|--- requirements.txt
|--- latex
|   |--- img
|       |--- models
|           |--- language.png
|           |--- framework.png
|           |--- character.png
|           |--- character_movies.png
|           |--- movies.png
|       |--- pdf
|           |--- pdf.png
|       |--- email
|           |--- apps.png
|           |--- email.png
|           |--- emailConfig.png
|           |--- emisor.png
|           |--- receptor.png
|       |--- logo_abet.png
|       |--- logo_episunsa.png
|   |--- src
|       |--- models
|           |--- models.py
|       |--- pdf
|           |--- invoice.html
|           |--- utils.py
|           |--- views.py
|       |--- email
|           |--- index.html
|           |--- views.py
|   |--- informe_lab07_Jh4stYn.pdf
|   |--- informe_lab07_Jh4stYn.tex
|--- model_examples
|   |--- example
|   |--- model_example
|   |--- manage.py
|--- pdf_example
|   |--- pdf_example
|   |--- templates/pdf
|   |--- manage.py
|--- emails_example
|   |--- emails_example
|   |--- send
|   |--- manage.py
```