

Informe de Laboratorio 07

Tema: Relaciones de uno a muchos, muchos a muchos y impresion de pdf y emails

Nota

Estudiantes	Escuela	Asignatura
- Alejandro Sebastian Alfonso Huacasi aalfonso@unsa.edu.pe	Escuela Profesional de Ingenieria de Sistemas	Programación web 2 Semestre: A

Laboratorio	Tema	Duración
07	Relaciones de uno a muchos, muchos a muchos y impresion de pdf y emails	04 horas

Semestre académico:	Fecha de inicio	Fecha de entrega
2023 - A	Del 29 Junio 2023	Al 14 Julio 2023

TAREA

- Reproducir las actividades de los videos donde trabajamos:
- Relación de uno a muchos
- Relación muchos a muchos
- Impresión de pdfs
- Envio de emails

1. Equipos, materiales y temas utilizados

- Sistema Operativo (GNU/Linux de preferencia).
- GNU Vim.
- Python 3
- Git.
- Cuenta en GitHub con el correo institucional.
- Entorno virtual
- Django 4

2. URL DE MI REPOSITORIO DE GIT HUB

- <https://github.com/Sebastianzzzin/proyecto.git>

3. Actividades con el repositorio GitHub

3.1. Creando e inicializando repositorio GitHub

- Como ya tenemos nuestro repositorio GitHub y además esta clonado e inicializado en nuestra máquina local.
- Se realizaron los siguientes comandos en la computadora:

Listing 1: Creando carpeta de trabajo dentro de nuestro repositorio clonado en mi maquina local

```
$ cd proyecto  
$ mkdir lab07
```

Listing 2: Dirigiéndonos a la carpeta de trabajo

```
$ cd lab07
```

Listing 3: Creando carpetas que contendrán el proyecto e imagenes

```
$ cd lab07  
$ mkdir emailexample  
$ mkdir model_examples  
$ mkdir pdfs_examples
```

3.2. Commits

- A continuación se mostraran capturas de los commits, primero de la creacion de la carpeta de trabajo de lab07 y la carpeta de los ejemplos de envio de emails:

```
commit 19fba09639168709b6930fa8c1099eb8057f8aac  
Author: Sebastian Alfonso Huacasi <aalfonso@unsa.edu.pe>  
Date: Fri Jul 14 21:47:16 2023 -0500  
  
Se ha creado la carpeta de trabajo lab07 y tambien se esta subiendo la carpeta que contiene el ejemplo de los envios de emails
```

- Commit de la creacion de la carpeta de los ejemplos de model:

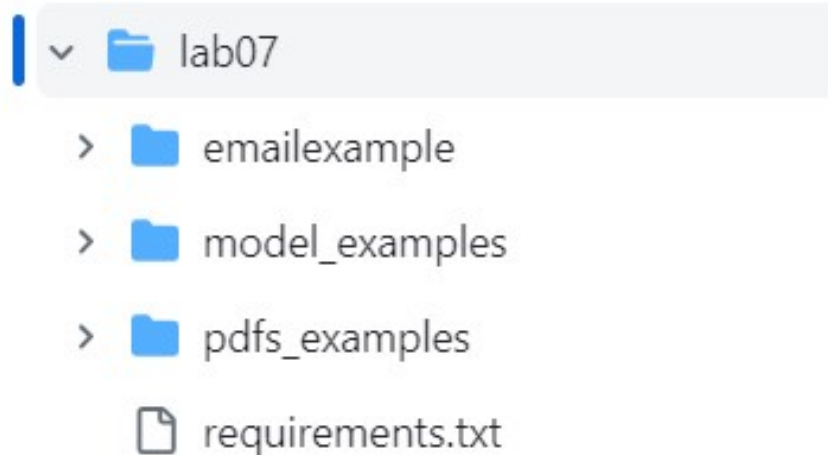
```
commit a525a50cba526baae4f9463da482089458a63c54  
Author: Sebastian Alfonso Huacasi <aalfonso@unsa.edu.pe>  
Date: Fri Jul 14 21:55:47 2023 -0500  
  
Carpeta de los ejemplos de model
```

- Commit que indica que se estan subiendo la carpeta de los ejemplos de pdf y además los template:

```
commit 29696d148d915c8f73f77fcac7325bce70f8dac0
Author: Sebastian Alfonso Huacasi <aalfonso@unsa.edu.pe>
Date: Fri Jul 14 22:03:55 2023 -0500
```

Se esta subiendo la carpeta de los ejemplos de pdf y tmb ahi se encuentra template

- Todo esto serían los archivos y orden de nuestro laboratorio 07:



Listing 4: views.py-emails, se encuentra en la carpeta emailexample, dentro de la carpeta de send

```
1 from django.shortcuts import render
2 from django.core.mail import send_mail
3
4 def index(request):
5     send_mail('Hola desde otra cuenta',
6             'Hello there. Esto es un mensaje automatico.',
7             'sebastianalfonso2004@gmail.com',
8             ['aalfonso@unsa.edu.pe'],
9             fail_silently=False)
10
11     return render(request, 'send/index.html')
```

- A continuación, procederé a explicar un poco el views.py de la parte de los emailexample:
- `from django.shortcuts import render`: Importa la función `render` del módulo `shortcuts` de Django, que se utiliza para renderizar una plantilla y devolver una respuesta HTTP.
- `from django.core.mail import send_mail`: Importa la función `send_mail` del módulo `mail` de Django, que se utiliza para enviar un correo electrónico. Define una función de vista llamada `index` que toma un objeto `request` como argumento. En Django, las funciones de vista deben devolver una respuesta HTTP.
- `send_mail('Holadesdeotracuenta', 'Hellothere.Estoesunmensajeautomatico.', 'sebastianalfonso2004@gmail.com', ['aalfonso@unsa.edu.pe'], fail_silently=False)`: Llama a la función `send_mail` para enviar un correo electrónico. Los parámetros proporcionados son:
 - 'Holadesdeotracuenta': El asunto del correo electrónico.
 - 'Hellothere.Estoesunmensajeautomatico.': El cuerpo del correo electrónico.
 - 'sebastianalfonso2004@gmail.com': La dirección de correo electrónico del remitente.
 - ['aalfonso@unsa.edu.pe']: La dirección de correo electrónico del destinatario.
 - `fail_silently=False`: Indica si se debe silenciar los errores al enviar el correo.

onso@unsa.edu.pe' : Una lista de direcciones de correo electrónico de los destinatarios.

- `fail_silently = False` : Indica que se debe generar una excepción si ocurre un error al enviar el correo electrónico. `return render(r`
`Renderiza la plantilla 'send/index.html' y devuelve la respuesta HTTP resultante. En este caso, la función de vistas simplemente`

Listing 5: `views.py`-pdfs, se encuentra en la carpeta `pdfs_examples`, dentro de la carpeta `pdfs_example`

```
1 from django.http import HttpResponse
2 from django.views.generic import View
3 from django.template.loader import get_template
4
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': 123456,
12            'customer_name': 'Sebas Alfonso',
13            'amount': 1981.99,
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" %("666")
21            content = "inline; filename=\"%s\"" % (filename)
22            download = request.GET.get("download")
23            if download:
24                content = "attachment; filename=\"%s\"" % (filename)
25            response['Content-Disposition'] = content
26            return response
27        return HttpResponse("Not found")
```

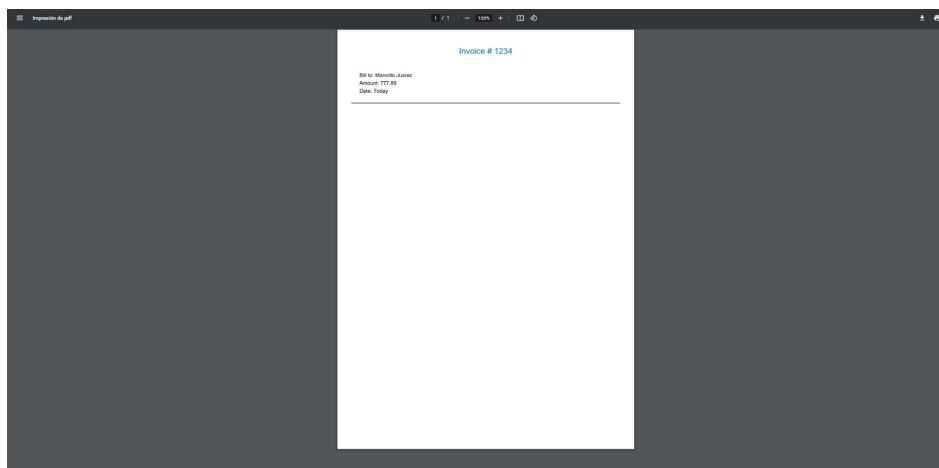
- A continuación, procederé a explicar un poco el `views.py` de la parte de los `pdfs_example` :
`from django.http import HttpResponse` : Importa la clase `HttpResponse` del módulo `http` de Django. `HttpResponse` se utiliza para crear una respuesta HTTP.
- `from django.views.generic import View`: Importa la clase `View` del módulo `generic` de Django. `View` es una clase base para vistas genéricas en Django.
- `from django.template.loader import get_template` : Importa la función `get_template` del módulo `loader` de Django. `get_template` devuelve una instancia de `Template` que representa una plantilla de Django. `Importa una función llamada render_to_pdf del módulo utils en el mismo directorio. Esta función se utiliza para generar un archivo PDF a partir de una plantilla HTML y un contexto.`
- `def get(self, request, *args, **kwargs)`: Define un método `get` que manejará las solicitudes HTTP GET. Toma `request` y otros argumentos opcionales.
- `template = get_template('pdf/invoice.html')` : Carga la plantilla `'pdf/invoice.html'` utilizando la función `get_template`. `context = {'invoice_id': 123456, 'customer_name': 'Sebas Alfonso', 'amount': 1981.99, 'today': 'Today'}` : Crea un diccionario `context` que contiene variables de contexto utilizadas en la plantilla. En este caso, incluye el número de factura.
- `html = template.render(context)`: Renderiza la plantilla con el contexto proporcionado, generando una cadena HTML.
- `pdf = render_to_pdf('pdf/invoice.html', context)` : Genera un archivo PDF utilizando la función `render_to_pdf` y la plantilla junto con el contexto. `HttpResponse(pdf, content_type='application/pdf')` : Crea una instancia de `HttpResponse` con el contenido del archivo PDF y el tipo de contenido `'application/pdf'`.

3.3. Capturas de la ejecucion del laboratorio 07

- Index email:



- Demostración del pdf:



- Email recibido, aumentandole algunas cosas más:

CONTACTO

Asunto: SEGUNDA PRUEBA

Email: Sebas@gmail.com

Mensaje: Laboratorio 7

Enviar

□ ☆ yo

SEGUNDA PRUEBA - Laboratorio 7

22:09

SEGUNDA PRUEBA

Recibidos ×



luisvaravega@gmail.com

para luis.vara.vega ▼

Laboratorio 7

Gracias por la ---INFO---

3.4. Estructura de laboratorio 07

- El contenido que se entrega del informe del laboratorio 07 es el siguiente:

```
Lab07-Informe/  
|--- informe-latex
```

```
|--- contenido
| |--- actividades.tex
| |--- caratula.tex
| |--- github.tex
| |--- materiales.tex
| |--- preguntas.tex
| |--- referencia.tex
| |--- rubrica.tex
| |--- tarea.tex
|--- img
| |--- logo_abet.png
| |--- logo_episunsa.png
| |--- logo_unsa.jpg
| |--- 1commit.jpg
| |--- 2commit.jpg
| |--- 3commit.jpg
| |--- e1.jpeg
| |--- e2.jpeg
| |--- e3.jpeg
| |--- e4.jpeg
| |--- estructura7.jpg
| |--- index.jpeg
| |--- pdf.jpeg
|--- src
| |--- views.py
| |--- viewss.py
|--- pweb2_lab07_aalfonso.pdf
|--- main.tex
```

4. Pregunta: Como te fue con el trabajo del laboratorio 07?

- Fue un laboratorio interesante para realizarlo con algunas dificultades, pero se pudo terminar. Realizando los 4 puntos que se pedía.

5. Rúbricas

5.1. Entregable Informe

Tabla 1: Tipo de Informe

Informe	
Latex	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.

5.2. Rúbrica para el contenido del Informe y demostración

- El alumno debe marcar o dejar en blanco en celdas de la columna **Checklist** si cumple con el ítem correspondiente.
- Si un alumno supera la fecha de entrega, su calificación será sobre la nota mínima aprobada, siempre y cuando cumpla con todos lo items.

- El alumno debe autocalificarse en la columna **Estudiante** de acuerdo a la siguiente tabla:

Tabla 2: Niveles de desempeño

Puntos	Nivel			
	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
2.0	0.5	1.0	1.5	2.0
4.0	1.0	2.0	3.0	4.0

Tabla 3: Rúbrica para contenido del Informe y demostración

Contenido y demostración		Puntos	Checklist	Estudiante	Profesor
1. GitHub	Hay enlace URL activo del directorio para el laboratorio hacia su repositorio GitHub con código fuente terminado y fácil de revisar.	2	X	2	
2. Commits	Hay capturas de pantalla de los commits más importantes con sus explicaciones detalladas. (El profesor puede preguntar para refrendar calificación).	4	X	2	
3. Código fuente	Hay porciones de código fuente importantes con numeración y explicaciones detalladas de sus funciones.	2	X	1	
4. Ejecución	Se incluyen ejecuciones/pruebas del código fuente explicadas gradualmente.	2	X	2	
5. Pregunta	Se responde con completitud a la pregunta formulada en la tarea. (El profesor puede preguntar para refrendar calificación).	2	X	2	
6. Fechas	Las fechas de modificación del código fuente estan dentro de los plazos de fecha de entrega establecidos.	2	X	2	
7. Ortografía	El documento no muestra errores ortográficos.	2	X	2	
8. Madurez	El Informe muestra de manera general una evolución de la madurez del código fuente, explicaciones puntuales pero precisas y un acabado impecable. (El profesor puede preguntar para refrendar calificación).	4	X	3	
Total		20		16	

6. Referencias

- https://www.w3schools.com/python/python_reference.asp
- <https://docs.djangoproject.com/en/4.2/howto/outputting-pdf/>
- <https://www.scaler.com/topics/django/relationships-in-django-models/>
- <https://docs.djangoproject.com/en/4.2/topics/email/>