

Informe de Laboratorio 08

Tema: Django: Uno a muchos, muchos a muchos, impresión de pdf y emails

Nota

Estudiante	Escuela	Asignatura
- David Alfredo Huamani Ollachica - Marco Antonio Suarez Huamani - Rafael Diego Nina Caliza-ya - Angel Paul Apaza Nazareth	Escuela Profesional de Ingeniería de Sistemas	Programación Web 2 Semestre: III Código: 1702122

Laboratorio	Tema	Duración
08	Django: Uno a muchos, muchos a muchos, impresión de pdf y emails	04 horas

Semestre académico	Fecha de inicio	Fecha de entrega
2024 - A	Del 20 Junio 2024	Al 24 Junio 2024

1. Tarea

- URL GitHub del Proyecto Django https://github.com/DrN25/pw2_lab08.git

2. Ejercicios Propuestos

Se trabajó un proyecto Django en el cual se trabaja con las relaciones de uno a muchos, de muchos a muchos, impresión de pdfs y envío de emails. Para ello, tenemos los siguientes modelos para cada uno de estos puntos.

2.1. Video del Proyecto en Flipgrid:

- Link del video: <https://flip.com/s/z85u-Xz4cV4f>

2.2. Modelos de Relación: De Uno a Muchos

- El código define dos modelos en Django para gestionar información de proyectos y tareas creadas usando la relación de Uno a Muchos. A continuación, se describen brevemente los modelos:

- **Modelo Proyecto:**

- **nombre:** El nombre del proyecto.
- **descripcion:** La descripción del proyecto.
- **fecha_inicio:** La fecha de inicio del proyecto.
- **fecha_fin:** La fecha de fin del proyecto.

- **Modelo Tarea:**

- **proyecto:** Una clave foránea que referencia al modelo Proyecto.
- **nombre:** El nombre de la tarea.
- **descripcion:** La descripción de la tarea.
- **fecha_creacion:** La fecha de creación de la tarea.
- **fecha_vencimiento:** La fecha de vencimiento de la tarea.
- **estado:** El estado de la tarea, con opciones para elegir.

Cada modelo también tiene un método que devuelve una representación en cadena de texto del objeto.

Listing 1: models.py

```
class Proyecto(models.Model):
    nombre = models.CharField(max_length=255)
    descripcion = models.TextField()
    fecha_inicio = models.DateField()
    fecha_fin = models.DateField()

    def __str__(self):
        return self.nombre

class Tarea(models.Model):
    proyecto = models.ForeignKey(Proyecto, related_name='tareas',
                                on_delete=models.CASCADE)
    nombre = models.CharField(max_length=255)
    descripcion = models.TextField()
    fecha_creacion = models.DateField(auto_now_add=True)
    fecha_vencimiento = models.DateField()
    estado = models.CharField(max_length=50, choices=[('pendiente', 'Pendiente'),
                                                    ('en_progreso', 'En Progreso'), ('completada', 'Completada')])

    def __str__(self):
        return self.nombre
```

2.3. Modelos de Relación: De Muchos a Muchos

- El código define dos modelos en Django para gestionar información de trabajadores y tareas-trabajador creadas usando la relación de Muchos a Muchos. A continuación, se describen brevemente los modelos:

- **Modelo Trabajador:**
 - nombre: El nombre del trabajador.
 - correo: El correo del trabajador.
- **Modelo TareaTrabajador:**
 - tarea: Una clave foránea que referencia al modelo Tarea.
 - trabajador: Una clave foránea que referencia al modelo Trabajador.

Cada modelo también tiene un método que devuelve una representación en cadena de texto del objeto.

Listing 2: models.py

```
class Trabajador(models.Model):
    nombre = models.CharField(max_length=100)
    correo = models.EmailField(unique=True)

    def _str_(self):
        return self.nombre

class TareaTrabajador(models.Model):
    tarea = models.ForeignKey(Tarea, related_name='trabajadores',
                             on_delete=models.CASCADE)
    trabajador = models.ForeignKey(Trabajador, related_name='tareas_asignadas',
                                  on_delete=models.CASCADE)

    class Meta:
        unique_together = ('tarea', 'trabajador')

    def _str_(self):
        return f'Tarea: {self.tarea.nombre} - Trabajador: {self.trabajador.nombre}'
```

2.4. Función de generación del PDF y envío de correo

Se crea una respuesta http

- **generar_pdf()**: Genera un archivo PDF con información de proyectos, tareas, trabajadores, y sus asignaciones, y lo envía como respuesta HTTP para su descarga.
- **enviar_correo()**: Envía correos electrónicos a todos los trabajadores con un mensaje pre-definido, y muestra mensajes de éxito o error en la interfaz.
 - Cada modelo también tiene un método que devuelve una representación en cadena de texto del objeto.

Listing 3: views.py

```
def generar_pdf(request):
    proyectos = Proyecto.objects.all()
    tareas = Tarea.objects.all()
    trabajadores = Trabajador.objects.all()
    tareas_trabajadores = TareaTrabajador.objects.all()

    response = HttpResponse(content_type='application/pdf')
    response['Content-Disposition'] = 'attachment;
        filename="informe_proyectos_tareas.pdf"'
```

```
buffer = BytesIO()
doc = SimpleDocTemplate(buffer, pagesize=letter, leftMargin=50, rightMargin=50,
    topMargin=50, bottomMargin=50)
elements = []

elements.append(Paragraph("Informe de Proyectos, Tareas y Trabajadores"))

elements.append(Paragraph("Proyectos:"))
proyecto_data = [['ID', 'Nombre', 'Descripcin']]
for proyecto in proyectos:
    proyecto_data.append([str(proyecto.id), proyecto.nombre,
        proyecto.descripcion])

proyecto_table = Table(proyecto_data, colWidths=[50, 200, 300])
proyecto_table.setStyle(TableStyle([
    ('ALIGN', (0, 0), (-1, -1), 'LEFT'),
    ('VALIGN', (0, 0), (-1, -1), 'MIDDLE'),
    ('LINEBELOW', (0, 0), (-1, 0), 1, (0, 0, 0)),
    ('BACKGROUND', (0, 0), (-1, 0), (0.8, 0.8, 0.8)),
]))
elements.append(proyecto_table)

elements.append(Paragraph("<br/><br/>"))

elements.append(Paragraph("Tareas:"))
tarea_data = [['ID', 'Nombre', 'Descripcin']]
for tarea in tareas:
    tarea_data.append([str(tarea.id), tarea.nombre, tarea.descripcion])

tarea_table = Table(tarea_data, colWidths=[50, 200, 300])
tarea_table.setStyle(TableStyle([
    ('ALIGN', (0, 0), (-1, -1), 'LEFT'),
    ('VALIGN', (0, 0), (-1, -1), 'MIDDLE'),
    ('LINEBELOW', (0, 0), (-1, 0), 1, (0, 0, 0)),
    ('BACKGROUND', (0, 0), (-1, 0), (0.8, 0.8, 0.8)),
]))
elements.append(tarea_table)

elements.append(Paragraph("<br/><br/>"))

elements.append(Paragraph("Trabajadores:"))
trabajador_data = [['ID', 'Nombre', 'Correo']]
for trabajador in trabajadores:
    trabajador_data.append([str(trabajador.id), trabajador.nombre,
        trabajador.correo])

trabajador_table = Table(trabajador_data, colWidths=[50, 200, 200])
trabajador_table.setStyle(TableStyle([
    ('ALIGN', (0, 0), (-1, -1), 'LEFT'),
    ('VALIGN', (0, 0), (-1, -1), 'MIDDLE'),
    ('LINEBELOW', (0, 0), (-1, 0), 1, (0, 0, 0)),
    ('BACKGROUND', (0, 0), (-1, 0), (0.8, 0.8, 0.8)),
]))
elements.append(trabajador_table)
```

```
elements.append(Paragraph("<br/><br/>"))

elements.append(Paragraph("Tareas de Trabajadores:"))
tt_data = [['Tarea', 'Trabajador']]
for tt in tareas_trabajadores:
    tt_data.append([tt.tarea.nombre, tt.trabajador.nombre])

tt_table = Table(tt_data, colWidths=[200, 200])
tt_table.setStyle(TableStyle([
    ('ALIGN', (0, 0), (-1, -1), 'LEFT'),
    ('VALIGN', (0, 0), (-1, -1), 'MIDDLE'),
    ('LINEBELOW', (0, 0), (-1, 0), 1, (0, 0, 0)),
    ('BACKGROUND', (0, 0), (-1, 0), (0.8, 0.8, 0.8)),
]))
elements.append(tt_table)

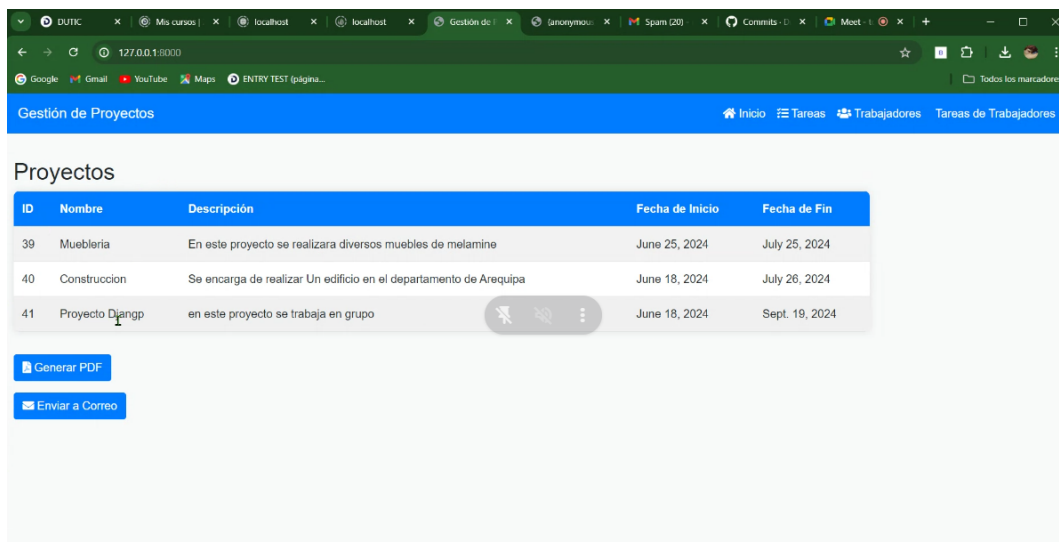
doc.build(elements)
pdf = buffer.getvalue()
buffer.close()
response.write(pdf)

return response

def enviar_correo(request):
    try:
        trabajadores = Trabajador.objects.all()
        for trabajador in trabajadores:
            send_mail(
                'Mensaje del Grupo 3 de PWEB2',
                'Hola, este es un mensaje del Grupo 3 de PWEB2.',
                'grupopweb2sh@gmail.com',
                [trabajador.correo],
                fail_silently=False,
            )
        messages.success(request, 'Correos enviados exitosamente.')
    except Exception as e:
        messages.error(request, f'Hubo un error al enviar los correos: {e}')
    return redirect('index')
```

3. Diseño de la Página Web

3.1. Capturas de pantalla de las tablas y la página web:

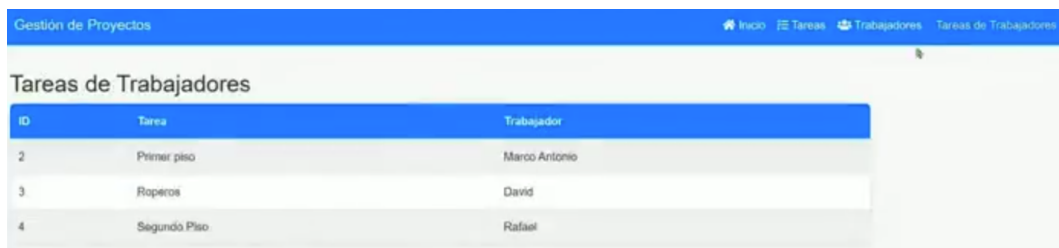


The screenshot shows a web browser window with the URL 127.0.0.1:8000. The page title is 'Gestión de Proyectos'. The navigation bar includes links for 'Inicio', 'Tareas', 'Trabajadores', and 'Tareas de Trabajadores'. The main content area is titled 'Proyectos' and displays a table with the following data:

ID	Nombre	Descripción	Fecha de Inicio	Fecha de Fin
39	Mueblería	En este proyecto se realizara diversos muebles de melamine	June 25, 2024	July 25, 2024
40	Construccion	Se encarga de realizar Un edificio en el departamento de Arequipa	June 18, 2024	July 26, 2024
41	Proyecto Django	en este proyecto se trabaja en grupo	June 18, 2024	Sept. 19, 2024

Below the table, there are two buttons: 'Generar PDF' and 'Enviar a Correo'.

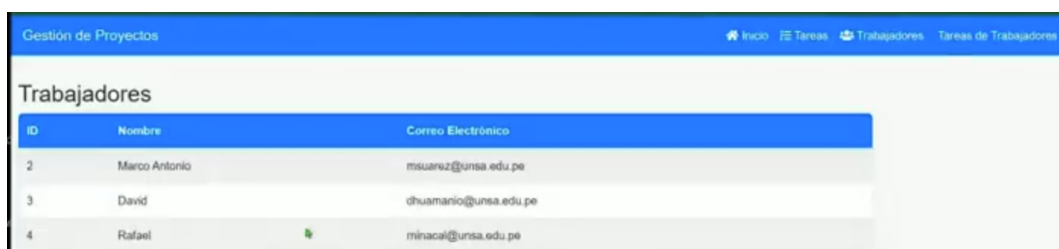
Figura 1: Página de los proyectos creados.



The screenshot shows the 'Tareas de Trabajadores' section of the web application. It displays a table with the following data:

ID	Tarea	Trabajador
2	Primer piso	Marco Antonio
3	Roperos	David
4	Segundo Piso	Rafael

Figura 2: Página de las tareas.



The screenshot shows the 'Trabajadores' section of the web application. It displays a table with the following data:

ID	Nombre	Correo Electrónico
2	Marco Antonio	msuanaz@unsa.edu.pe
3	David	dhuamano@unsa.edu.pe
4	Rafael	minacal@unsa.edu.pe

Figura 3: Página de los trabajadores creados.

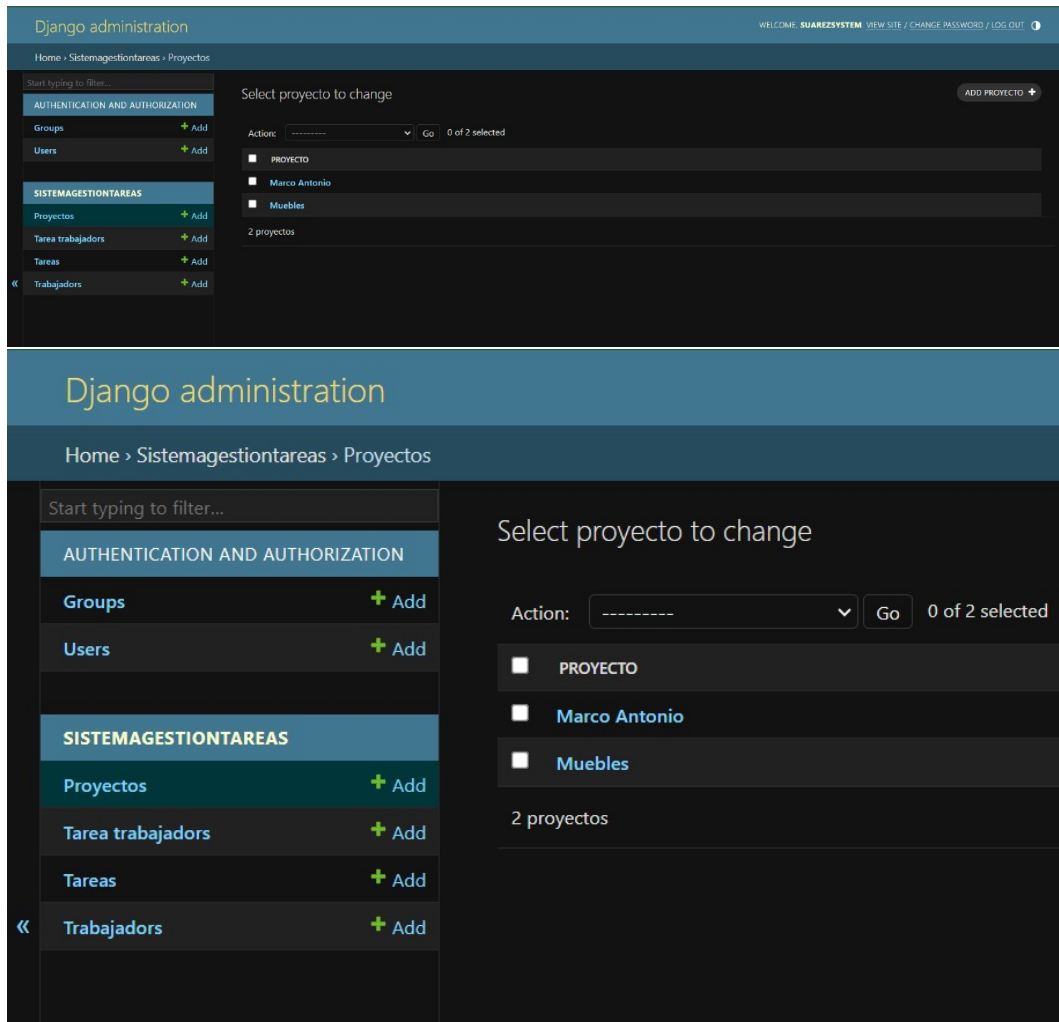


Figura 4: Página admin de las tablas creadas.

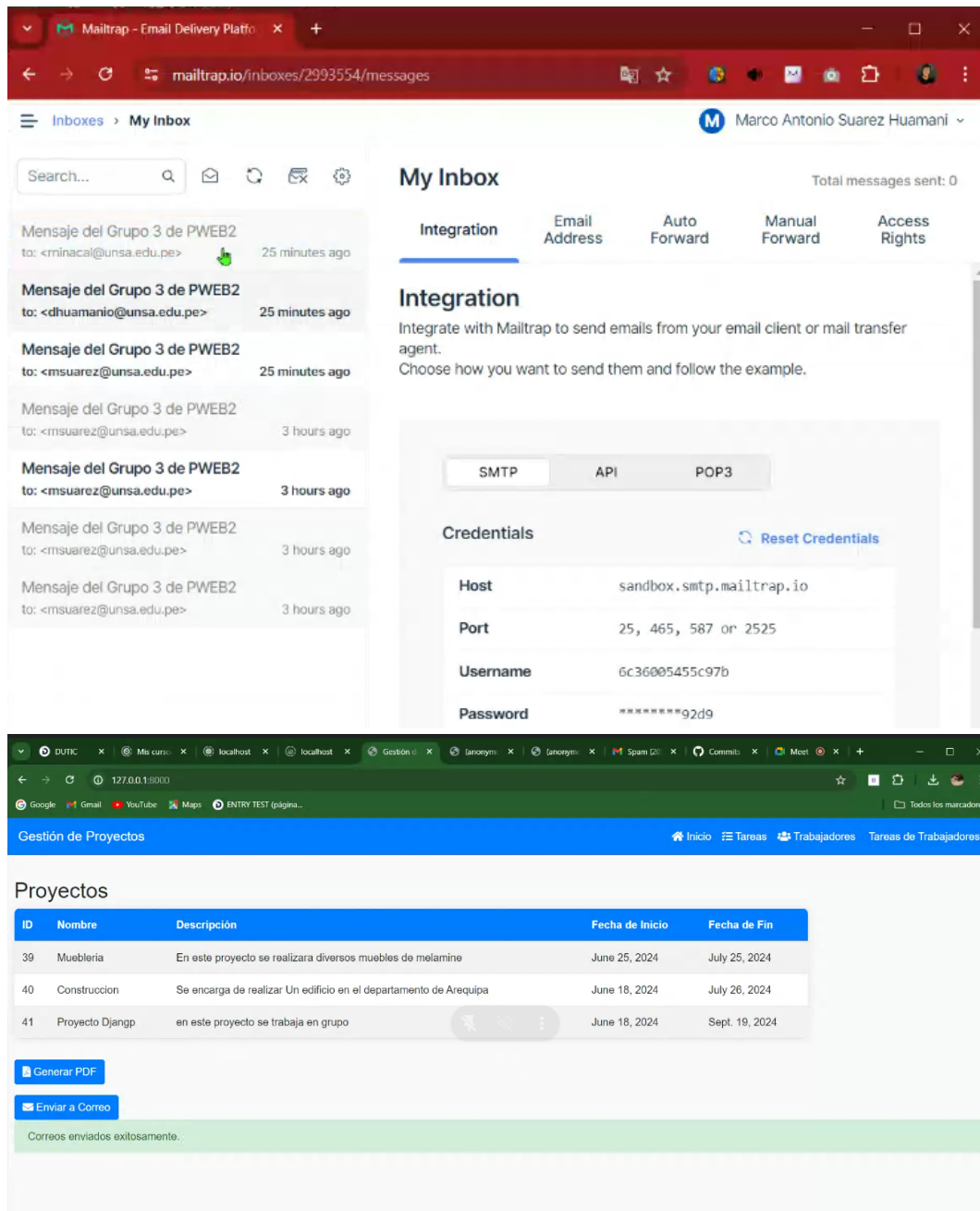


Figura 5: Imagenes de los correos enviados

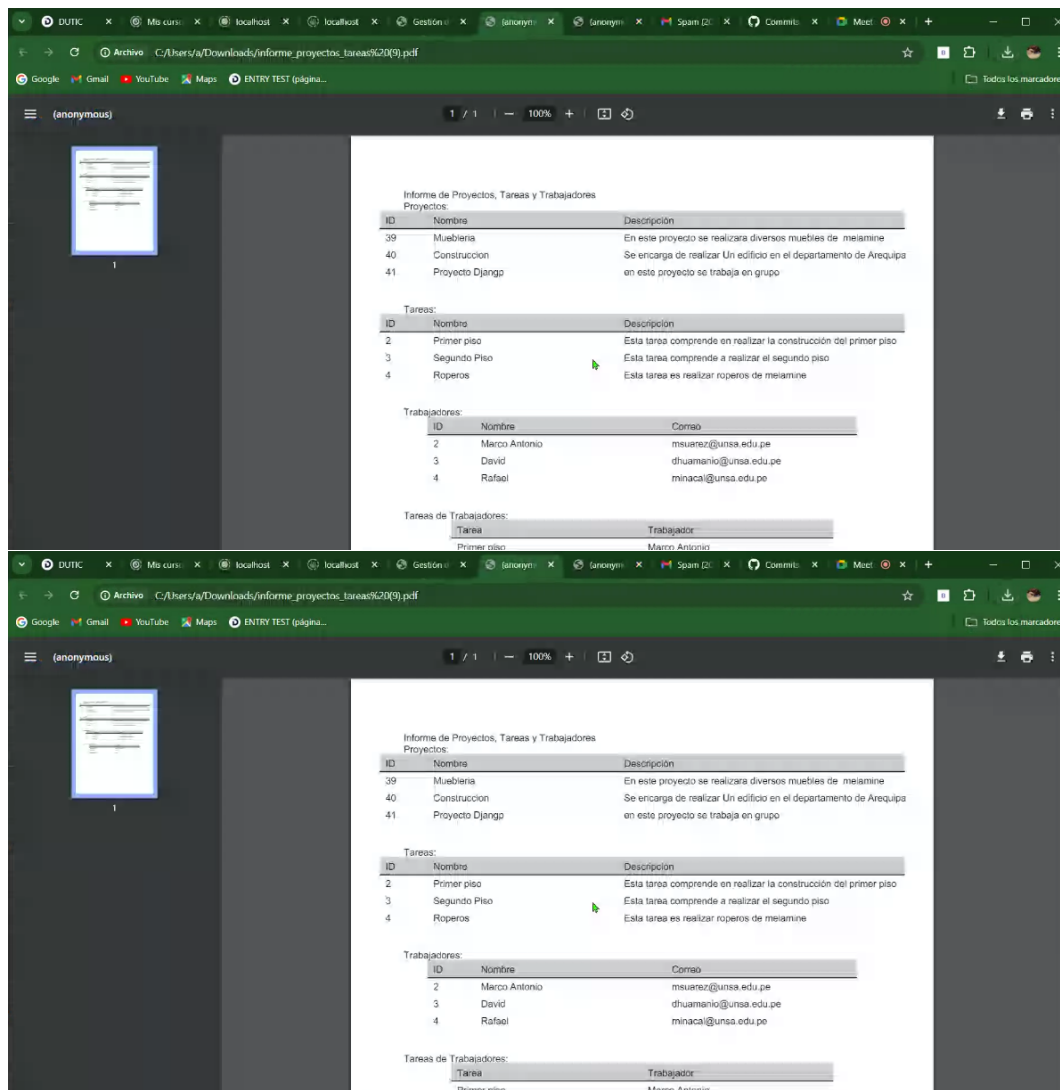


Figura 6: Imagenes del pdf generado.

4. Commits realizados

:

Listing 4: commits

```
* 893fcc5 (HEAD -> main, origin/main, origin/HEAD) Final del manage.py completo
* 88f6cdb funcionalidades completas , final
* 0a500f0 eliminacin de credenciales de correo temporal
* d9dcd46 funcionalidades de enviar correo y generar pdf
* 3af2a73 Se quito las opciones de agregar y eliminar
* 2aa7418 Aadiendo latex y pdf
* f86e9db eliminando pdf
* b262f4c eliminando latex
* e3ee7f3 Add files via upload
* 2c46a13 proyecto culminado
* 4113a4d aadiendo el gestiontareas completado
* b7ab375 Delete Project_lab08 directory
* d267ccb Delete sistemagegestiontareas
* 97c9907 actualizado el sistemagegestiontareas
* 6695b81 todo settings.py actualizado
* cb39dbe aadiendo la base de datos
* 68fe603 Add files via upload
```

5. Rúbricas

5.1. Sobre el 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

	Nivel			
Puntos	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	2	
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	0.5	
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		15.5	

6. Referencias

- https://docs.djangoproject.com/en/5.0/topics/db/examples/many_to_one/
- <https://docs.djangoproject.com/>
- <https://www.w3schools.com/python/>
- <https://www.w3schools.com/django/>