

# Programación Web 2 - Grupo B/Lab7

## Relaciones de uno a muchos, muchos a muchos y generación de PDF y envío de emails

Alumno: Vilca Peralta, Jimmy Joaquín  
Docente: Corrales Delgado, Carlo Jose Luis

14 de julio de 2023

### 1. ACTIVIDAD

**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

### 2. RESOLUCIÓN

Durante la realización de esta actividad utilizando Django, visualizamos cuatro actividades: la relación de uno a muchos, la relación muchos a muchos, la generación de PDFs y el envío de correos electrónicos.

#### 2.1. Impresión de PDF's

En cuanto a la generación de PDFs, utilizamos la biblioteca externa xhtml2pdf junto con Django. Creamos una función llamada `render_to_pdf` que tomó como argumento la ruta de una plantilla HTML. Esta función utilizó la biblioteca xhtml2pdf para convertir el HTML resultante en un archivo PDF. El archivo PDF se devolvió como una respuesta HTTP, lo que permitió su descarga o visualización por parte del usuario.



Figura 1: Visualización

```

from io import BytesIO
from django.http import HttpResponse
from django.template.loader import get_template

from xhtml2pdf import pisa

def render_to_pdf(template_src, context_dict={}):
    template = get_template(template_src)
    html = template.render(context_dict)
    result = BytesIO()
    pdf = pisa.pisaDocument(BytesIO(html.encode("ISO-8859-1")), result)
    if not pdf.err:
        return HttpResponse(result.getvalue(), content_type='application/pdf')
    return None

```

Figura 2: Función render to PDF

```

from django.http import HttpResponse
from django.views.generic import View

from .utils import render_to_pdf #created in step 4
from django.template.loader import get_template

class GeneratePdf(View):
    def get(self, request, *args, **kwargs):
        template = get_template('invoice.html')
        context = {
            "invoice_id": 123,
            "customer_name": "John Cooper",
            "amount": 1399.99,
            "today": "Today"
        }
        html = template.render(context)
        pdf = render_to_pdf('invoice.html', context)
        return HttpResponse(pdf, content_type='application/pdf')

```

Figura 3: Generador del PDF

## 2.2. Envío de Emails

Por otra parte, implementamos la funcionalidad de envío de correos electrónicos utilizando las capacidades integradas de Django. Configuramos los ajustes necesarios en el archivo de configuración de Django (settings.py) para el envío de correos electrónicos. Establecimos el backend de correo electrónico como `django.core.mail.backends.smtp.EmailBackend` y proporcionamos los detalles de configuración necesarios, como el servidor de correo saliente (EMAIL\_HOST), el puerto (EMAIL\_PORT), el nombre de usuario (EMAIL\_HOST\_USER), la contraseña (EMAIL\_HOST\_PASSWORD) y si se debe utilizar TLS (EMAIL\_USE\_TLS). Además, establecimos la dirección de correo electrónico predeterminada desde la cual se enviarían los correos electrónicos (DEFAULT\_FROM\_EMAIL).

Al utilizar la configuración mencionada, pudimos enviar correos electrónicos desde nuestra aplicación Django. Esto nos permitió enviar notificaciones, informes generados en formato PDF u otro tipo de comunicación relevante a los usuarios de nuestra aplicación.

```
EMAIL_BACKEND = 'django.core.mail.backends.smtp.EmailBackend'
EMAIL_HOST = 'smtp.gmail.com'
EMAIL_PORT = 587
EMAIL_HOST_USER = 'jvilcapera@unsa.edu.pe'
EMAIL_HOST_PASSWORD = ''
EMAIL_USE_TLS = True
DEFAULT_FROM_EMAIL = 'jvilcapera@unsa.edu.pe'
```

Figura 4: Configuración para el envío de Email

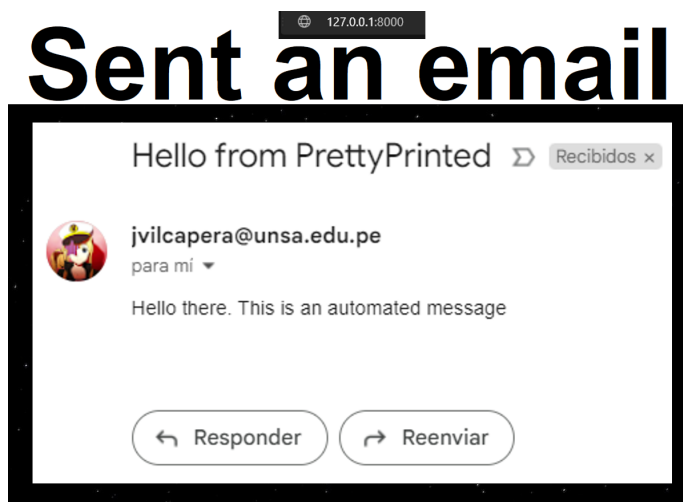


Figura 5: Envío y confirmación del Email

Finalmente, Mediante el uso de Django logramos implementar relaciones de

uno a muchos y muchos a muchos en nuestra base de datos, generar PDFs personalizados y enviar correos electrónicos utilizando la configuración adecuada en el archivo settings.py. Estas funcionalidades son fundamentales en el desarrollo de aplicaciones web y permiten una interacción eficiente y efectiva con los usuarios.

ENLACE ARCHIVOS: GitHub - JimmyVilcaP/Pweb2A-Laboratorio7

### 3. Conclusiones

Durante la realización de esta actividad utilizando Django, he explorado y aplicado diversas funcionalidades fundamentales para el desarrollo de aplicaciones web. Mediante el uso de relaciones de uno a muchos y muchos a muchos, hemos sido capaces de establecer conexiones entre entidades en nuestra base de datos, lo cual nos permite representar y acceder a datos relacionados de manera eficiente.

Además, he aprendido a generar archivos PDF dinámicamente utilizando la biblioteca xhtml2pdf, lo cual nos ha permitido crear informes personalizados y otros documentos en formato PDF. Esto resulta especialmente útil para presentar información de manera estructurada y profesional.

Por último, hemos implementado la funcionalidad de envío de correos electrónicos utilizando las capacidades integradas de Django. Esto nos ha brindado la capacidad de comunicarnos con los usuarios de nuestras aplicaciones mediante el envío automatizado de mensajes y notificaciones importantes.

Estas funcionalidades, en conjunto, nos proporcionan las herramientas necesarias para desarrollar aplicaciones web completas, capaces de gestionar y presentar datos de manera eficiente, así como de interactuar con los usuarios de manera efectiva. A través de esta actividad, he adquirido conocimientos y habilidades que serán de gran utilidad en proyectos futuros.

### 4. Referencias

#### Referencias

- [1] One To Many Relationships-8.m4v
- [2] Query One To Many-10.m4v
- [3] Many To Many Relationships-11.m4v
- [4] Many To Many Query-12.m4v
- [5] Database Settings-13.m4v
- [6] Render a Django HTML Template to a PDF file Django Utility CFE Render to PDF.mp4

[7] Sending Emails in Django.mp4