



UNIVERSIDAD NACIONAL DE SAN AGUSTIN
FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS
DEPARTAMENTO ACADÉMICO DE INGENIERÍA DE SISTEMAS E
INFORMÁTICA
ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS



Formato: Guía de Práctica de Laboratorio

Aprobación: 2022/03/01

Código: GUIA-PRLD-001

Página: 1

INFORME DE LABORATORIO

INFORMACIÓN BÁSICA

ASIGNATURA:	Programación Web 02			
TÍTULO DE LA PRÁCTICA:	Relaciones de uno a muchos, muchos a muchos y impresion de pdf y emails			
NÚMERO DE PRÁCTICA:	07	AÑO LECTIVO:	2023 A	NRO. SEMESTRE: III
FECHA DE PRESENTACIÓN:	19-Julio-2023	HORA DE PRESENTACIÓN:	11:00	
NOMBRE:	<ul style="list-style-type: none"> Añazco Huamanquispe, Andre Renzo 			NOTA:

DOCENTES:

- Anibal Sardon Paniagua

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

rescovedoq [GPL-3.0](#) [Downloads](#) [no releases](#) [Last Commit](#) [august 2022](#)



OBJETIVOS TEMAS Y COMPETENCIAS

- Implementar** una aplicación en Django utilizando una plantilla profesional.
- Utilizar una tabla de Destinos turísticos para leer y completar la página web.
- Utilizar los tags "if" y "for" en los archivos html para leer todos los registros de una tabla desde una base de datos.

TEMAS

- Relacion de uno a muchos
- relacion de muchos a uno
- Impresion de PDFs
- Envio de emails

► Details

COMPETENCIAS

- C.c Diseña responsablemente sistemas, componentes o procesos para satisfacer necesidades dentro de restricciones realistas: económicas, medio ambientales, sociales, políticas, éticas, de salud, de seguridad, manufacturación y sostenibilidad. - C.m Construye responsablemente soluciones siguiendo un proceso adecuado llevando a cabo las pruebas ajustada a los recursos disponibles del cliente. - C.p Aplica de forma flexible técnicas, métodos, principios, normas, estándares y herramientas de ingeniería necesarias para la construcción de software e implementación de sistemas de información.

CONTENIDO DE LA GUÍA

Actividades

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

Desarrollo de los videos

- Se inicia cambiando la base de datos para no interferir o tener conflictos con el proyecto anterior

```
DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.postgresql',
        'NAME': 'Lab07',
        'USER': 'postgres',
        'PASSWORD': '1234',
        'HOST': 'localhost',
    }
}
```

Relacion 1 a muchos

- Como se explica en el video, la interrelacion entre tablas es necesario para un proyecto. De esta forma se puede trabajar con la base de datos relativamente relacionada.
- Se crean 2 modelos, para que formen parte de la tabla en la base de datos Language/framework
- Se establece la relacion 1 a muchos entre lenguaje y framework
- Se Agrega desde el admin site (tambien es posible agregarlo desde el shell de python)

```
class Language(models.Model):
    name= models.CharField(max_length=10)

class Framework(models.Model):
    name= models.CharField(max_length=10)
    language = models.ForeignKey(Language, on_delete=models.CASCADE)
```

Add framework

Name:

Language:

SAVE Save and add another Save and continue editing

DB Browser for SQLite - C:\Users\HP\Documents\andr\Lab07_RelacionesDe1Amuchos\cinemaLog\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project Attach Database

Database Structure Browse Data Edit Pragma Execute SQL

Table: movie_language

id	name
Filter	Filter
1	1 Python

DB Browser for SQLite - C:\Users\HP\Documents\andr\Lab07_RelacionesDe1Amuchos\cinemaLog\db.sqlite3

File Edit View Tools Help

New Database Open Database Write Changes Revert Changes Open Project Save Project

Database Structure Browse Data Edit Pragma Execute SQL

Table: movie_framework

	id	name	language_id
	Filter	Filter	Filter
1	1	Django	1
2	2	Flask	1

- Se evidencia los commits hasta este paso

AndreRH09 new project 0 ca57c30 4 minutes ago 7 commits

cinemaLog	new project 0	4 minutes ago
imgs	readme	4 days ago
.gitignore	cargando antiguo proyecto	4 days ago
README.md	readme	4 days ago
ejemplo_informe_latex.tex	nueva app	yesterday
requirements.txt	requerimientos para Django	4 days ago

Query Relacion 1 a muchos

- Se realizo un filtrado con distintos parametros para probar la funcionalidad de la base de datos.

Table: movie_framework

	id	name	language_id
	Filter	Filter	Filter
1	1	Django	1
2	2	Flask	1
3	3	POCO	2
4	4	AutoMapper	3

	id	name
	Filter	Filter
1	1	Python
2	2	C++
3	3	C#

```

ValueError: Field 'id' expected a number but got 'Python'.
>>> Framework.objects.filter(language__name='Python')
<QuerySet [<Framework: Django>, <Framework: Flask>]>
>>> Framework.objects.filter(language__name='C++')
<QuerySet [<Framework: POCO>]>
>>> Framework.objects.filter(language__name='C#')
<QuerySet [<Framework: AutoMapper>]>
>>> Framework.objects.filter(language__name__startswith='Py')
<QuerySet [<Framework: Django>, <Framework: Flask>]>
>>> Language.objects.filter(framework__name__startswith='Au')
<QuerySet [<Language: C#>]>
>>>

```

- Se evidencia los commits hasta este paso

main	1 branch	0 tags	Go to file	Add file	Code
AndreRH09	query	d88b3ed	now	9	commits
cinemaLog	query		now		
imgs	query		now		
.gitignore	cargando antiguo proyecto		4 days ago		
README.md	query		now		
ejemplo_informe_latex.tex	nueva app		yesterday		
requirements.txt	requerimientos para Django		4 days ago		

Relacion de Mucho a muchos

- Se crean 2 modelos, para que formen parte de la tabla en la base de datos Movie/Character
- Se establece su relacion con **models.ManyToManyField**
- Se agregan desde el admin site

```

class Movie(models.Model):
    name= models.CharField(max_length=20)

    def __str__(self):
        return self.name

class Character(models.Model):
    name= models.CharField(max_length=20)
    movies = models.ManyToManyField(Movie)

    def __str__(self):
        return self.name

```

Change character

Capitan America

Name:

Capitan America

Movies:

Avengers

Civil War

Thor Dark World

+

Hold down "Control", or "Command" on a Mac, to select more than one.

SAVE

Save and add another

Save and continue editing

Select movie to change

Action:

Go

0 of 3 selected

☐ MOVIE

☐ Thor Dark World

☐ Civil War

☐ Avengers

3 movies

DB Browser for SQLite - C:\Users\HP\Docum

File Edit View Tools Help

New Database

Open Database

W

Database Structure

Browse Data

Edit Pragr

Table:

movie_character_movies

	id	character_id	movie_id
	Filter	Filter	Filter
1	1	1	1
2	2	2	1
3	3	2	2
4	4	2	3
5	5	1	2

- Se evidencia los commits hasta este paso


Lab07_RelacionesDe1Amuchos
Public
Pin
Unwatch 1

main
1 branch
0 tags
Go to file
Add file
Code

File	Commit Message	Time
cinemaLog	Mucho a mucho	now
imgs	Mucho a mucho	now
.gitignore	cargando antiguo proyecto	4 days ago
README.md	Mucho a mucho	now
ejemplo_informe_latex.tex	nueva app	yesterday
requirements.txt	requerimientos para Django	4 days ago

Html2PDF

- Se utilizo el tutorial y codigo de [TutorialHTML2PDF](#).
1. Se descargo las dependencias `pip install --pre xhtml2pdf`
 2. Se modifico el codigo

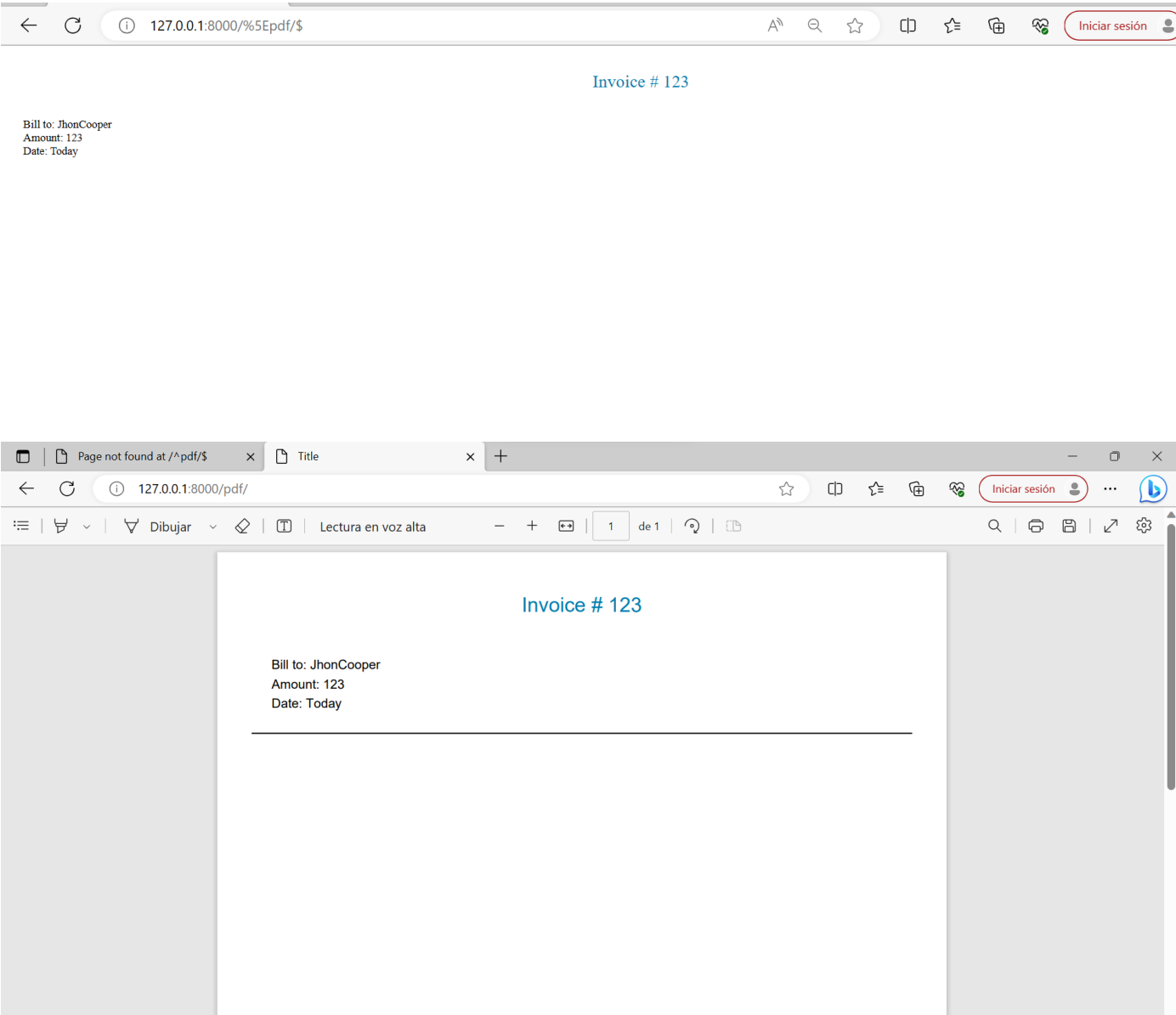
```
from django.template.loader import get_template

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
```

```
}
html= template.render(context)
pdf= render_to_pdf('invoice.html',context)
if pdf:
    response= HttpResponse(pdf, content_type='application/pdf')
    filename= "Invoice_%s.pdf" %("12")
    content = "inline; filename='%s'" %(filename)
    download = request.GET.get("download")
    if download:
        content= "attachment; filename=%s" %(filename)
    response['Content-Disposition'] =content
    return response
return HttpResponse("<h1>PDF no encontrado</h1>")
```

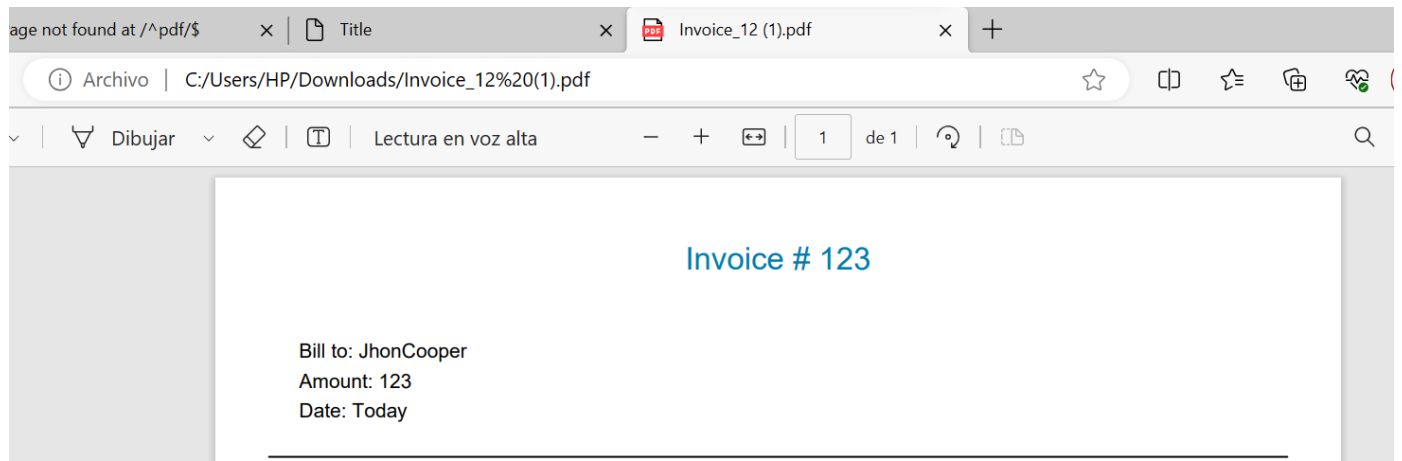
- Como resultado se paso de una renderizacion de HTML a PDF



- Finalmente se modifica la respuesta del renderizado para que se pueda descargar el pdf usando unicamente el condicional *download*

	Fecha de Modificación	tipo	tamaño
Hoy			
Invoice_12 (1)	9/7/2023 11:19	Microsoft Edge PDF ...	2 KB
Invoice_12	9/7/2023 11:19	Microsoft Edge PDF ...	2 KB
'Invoice_12.pdf'	9/7/2023 11:18	Archivo PDF'	2 KB
'{"Invoice_12 (1).pdf}'	9/7/2023 11:13	Archivo PDF{'	2 KB
'{"Invoice_12.pdf}'	9/7/2023 11:11	Archivo PDF{'	2 KB

- Se puede ver que el contenido del archivo descargado es el mismo que el renderizado, por lo que hay un correcto flujo de datos.



- Se evidencia los commits hasta este paso

	AndreRH09 render2PDF	d4e402a 29 minutes ago	🕒 13 commits
📁	cinemaLog	render2PDF	29 minutes ago
📁	imgs	pdf	11 hours ago
📄	.gitignore	cargando antiguo proyecto	5 days ago
📄	README.md	pdf	11 hours ago
📄	requirements.txt	pdf	11 hours ago

Django_Emails

- Se utilizó el tutorial y código de [TutorialHTML2PDF](#).
- Se descargó las dependencias `pip install --pre xhtml2pdf`
- Se modificó el código

```
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
```

REFERENCIAS

- https://www.w3schools.com/python/python_reference.asp
- <https://docs.python.org/3/tutorial/>