
	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:	Django Relaciones Uno a muchos y Muchos a muchos en BD				
NÚMERO DE PRÁCTICA:	07	AÑO LECTIVO:	2023 A	NRO. SEMESTRE:	III
FECHA DE PRESENTACIÓN:	11-07-2023	HORA DE PRESENTACIÓN:	12:00		
NOMBRE: <ul style="list-style-type: none"> Andre David Delgado Allpan 				NOTA:	
DOCENTES: <ul style="list-style-type: none"> Anibal Sardon Paniagua 					

Django

rescobedoq GPL-3.0

Downloads no releases

Last Commit august 2022



OBJETIVOS TEMAS Y COMPETENCIAS

OBJETIVOS

- **Implementar** una aplicación en Django con email y pdf, además de aprender a utilizar las relaciones en base de datos.

TEMAS

- Proyectos de Django
- Aplicaciones en Django
- Email en Django
- PDF en Django

CONTENIDO DE LA GUÍA

MARCO TEÓRICO

- [Documentación de Django](#)
- Django utiliza el modelo relacional para representar datos en la base de datos. En este modelo, se utilizan tablas para almacenar la información y las relaciones se establecen mediante claves foráneas.

Actividades

1. Crear un proyecto en Django
2. Siga los pasos del video
3. Use git y haga los commits necesarios para manejar correctamente la aplicación.

Ejercicios Propuestos

Video de explicación



: <https://flip.com/s/q9Ky6GHBs9jF>

Link de repositorio GittHub

<https://github.com/andre98652/pweb-lab7.git>

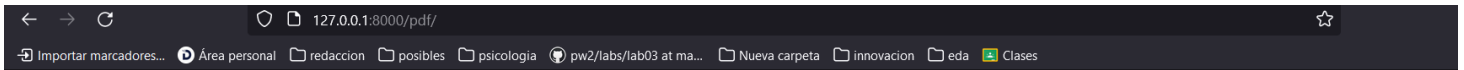
- Django pdf y emails

Django PDF [✓]

Django EMAILS [✓]

PDF

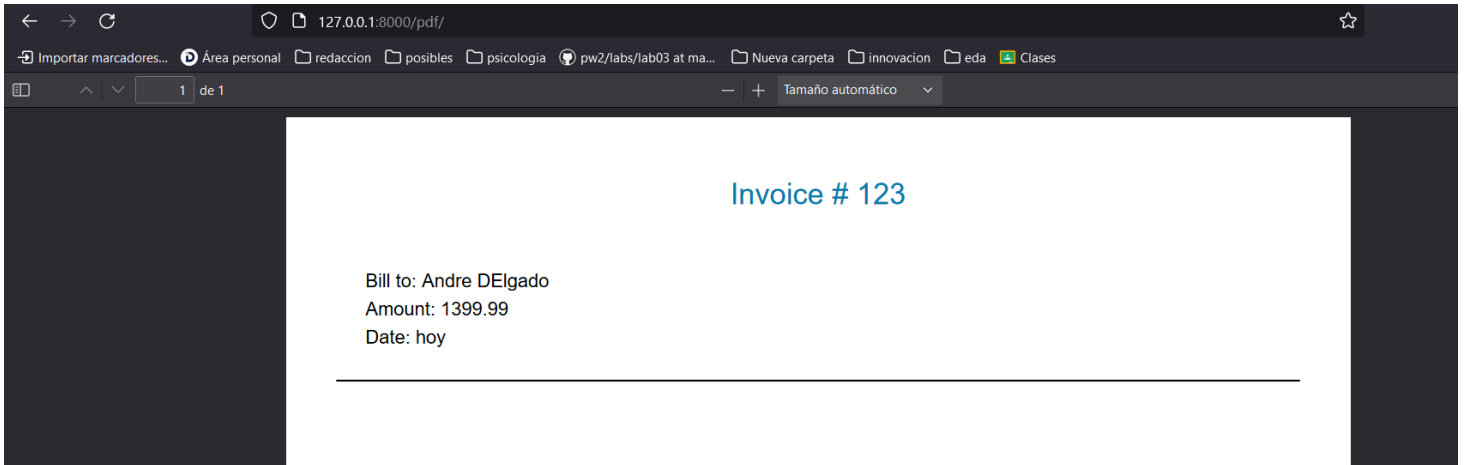
- Ejecucion 1 de ejercicio: Visualizar



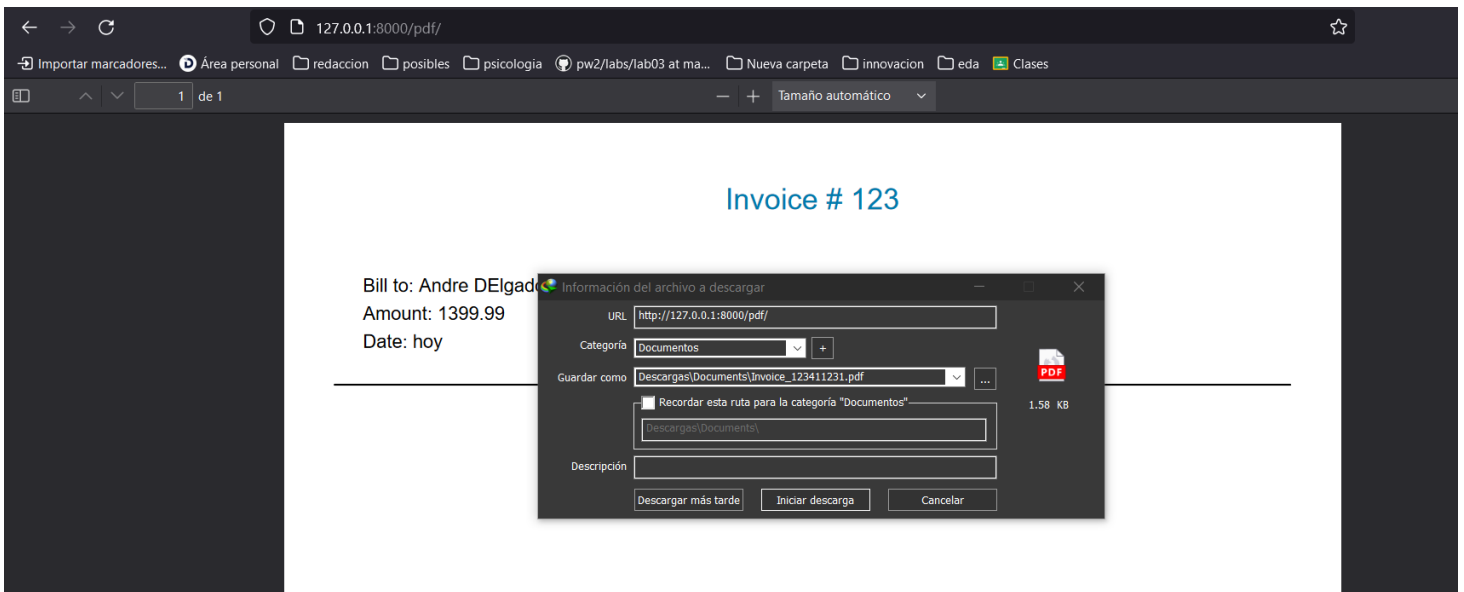
Invoice # 123

Bill to: Andre DElgado
Amount: 1399.99
Date: hoy

• Ejecucion 2 de ejercicio: Visualizar como PDF

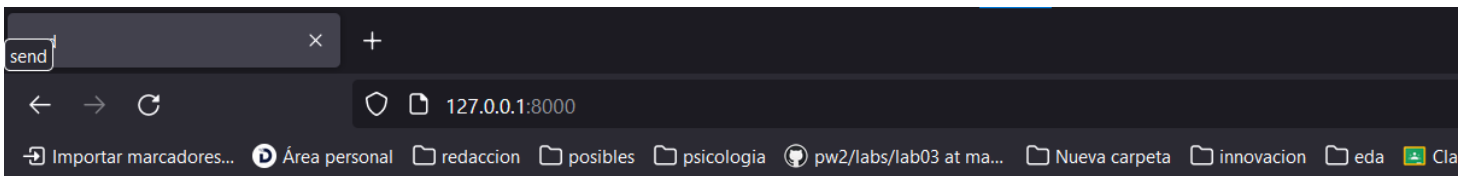


• Ejecucion 3 de ejercicio: Descargar automatica



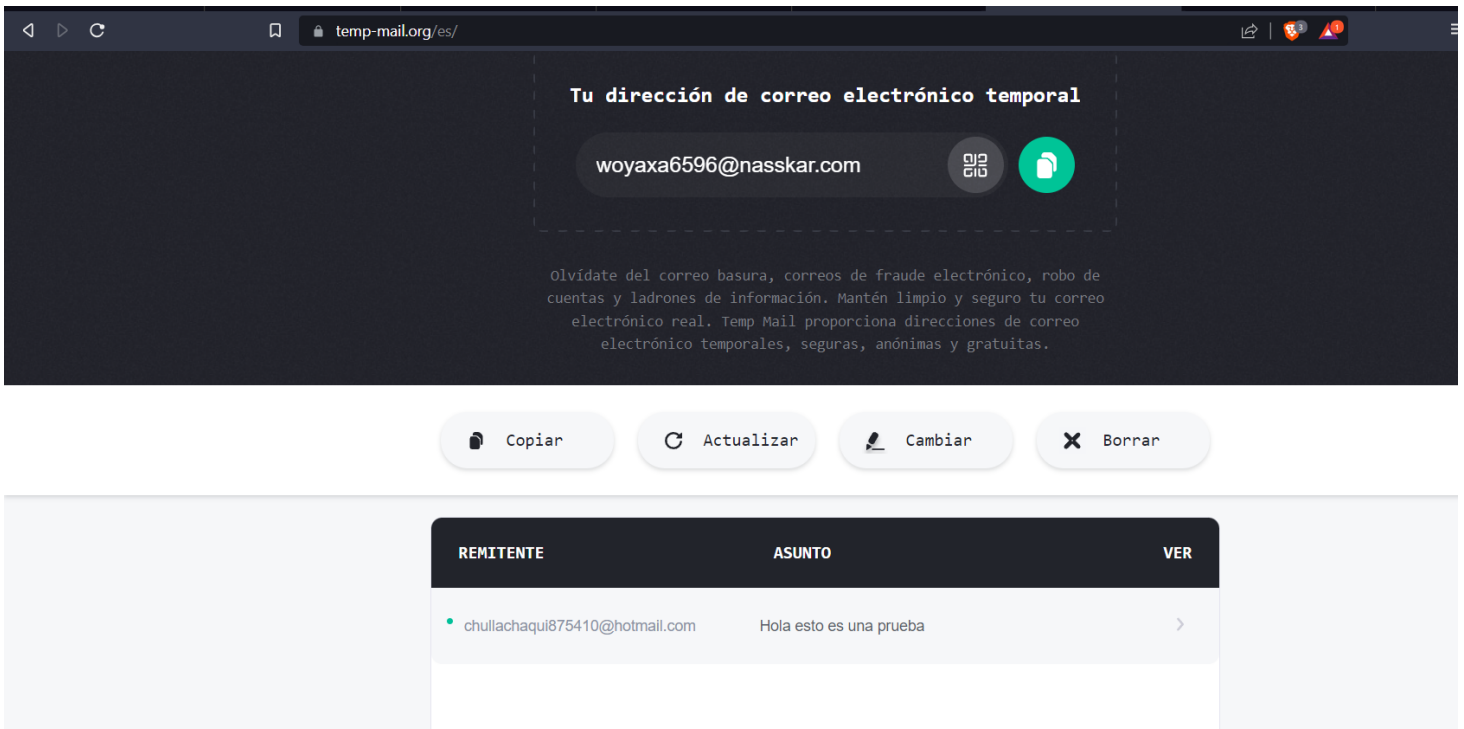
EMAIL

• Ejecucion 1 de ejercicio: Visualizar



send a email

- Ejecucion 1 de ejercicio: Mensaje enviado



Django Relaciones Uno a muchos y Muchos a muchos en BD

One To Many Relationships [✓]
Query One To Many [✓]
Many To Many Relationships [✓]
Many To Many Query [✓]
Database Settings [✓]

- Consulta con filtro

```
>>> Framework.objects.filter(language__name='Python')
<QuerySet [(<Framework: Django>, <Framework: Flask>, <Framework: Bottle>)]>
```

- Consulta con filtro 2

```
>>> Framework.objects.filter(language__name__startswith='Py')
<QuerySet [(<Framework: Django>, <Framework: Flask>, <Framework: Bottle>)]>
>>> Framework.objects.filter(language__name__startswith='Pe')
<QuerySet []>
```

- Agregando datos a tabla movies y character

```
(InteractiveConsole)
>>> from example.models import Movie, Character
>>> avengers = Movie(name='Avengers')
>>> avengers.save()
>>> captain_america = Character(name='Captain America')
>>> captain_america.save()
>>> captain_america.movies.add(avengers)
>>>
```

- Visualizacion de tabla movie y character

id	name
Filter	Filter
1	Avengers

id	name
Filter	Filter
1	1 Captain America

- Vizualizacion de tabla ManyToMany

	id	character_id	movie_id
	Filter	Filter	Filter
1	1	1	1

- Agregando datos a tablas con ManyToMany

```
>>> from example.models import Movie, Character
>>> avengers = Movie(name='Avengers')
>>> avengers.save()
>>> captain_america = Character(name='Captain America')
>>> captain_america.save()
>>> captain_america.movies.add(avengers)
>>> thor = Movie(name='Thor: The Dark World')
>>> thor_character = Character(name='Thor')
>>> civil_war = Movie(name='Civil War')
>>> thor.save()
>>> thor_character.save()
>>> civil_war.save()
>>> captain_america.movies.add(civil_war)
>>> thor_character.movies.add(avengers)
>>> thor_character.movies.add(thor)
>>> captain_america.movie.create(name='Winter Soldier')
Traceback (most recent call last):
  File "<console>", line 1, in <module>
AttributeError: 'Character' object has no attribute 'movie'
>>> captain_america.movies.create(name='Winter Soldier')
<Movie: Winter Soldier>
>>> █
```

- Consulta a tabla movies

```

>>>
>>> Character.objects.filter(movies__name='Civil War')
<QuerySet [<Character: Captain America>]>
>>> Movie.objects.filter(character__name='Captain America')
<QuerySet [<Movie: Avengers>, <Movie: Civil War>, <Movie: Winter Soldier>]>
>>> captain_america= Character.objects.get(name='Captain America')
>>> captain_america
<Character: Captain America>
>>> captain_america.movies.all()
Traceback (most recent call last):
  File "<console>", line 1, in <module>
NameError: name 'captain' is not defined
>>> captain_america.movies.all()
<QuerySet [<Movie: Avengers>, <Movie: Civil War>, <Movie: Winter Soldier>]>
>>> █

```

- Consulta através de ManyToMany

```

>>> avenger = Movie.objects.get(name='Avengers')
>>> avenger
<Movie: Avengers>
>>> avenger.character_set.all()
<QuerySet [<Character: Captain America>, <Character: Thor>]>
>>> █

```

REFERENCIAS

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





rescobedoq GPL-3.0

Downloads no releases

Last Commit august 2022