

# Laboratorio\_08\_PWEB2

---



UNIVERSIDAD NACIONAL DE SAN  
AGUSTIN  
FACULTAD DE INGENIERÍA DE  
PRODUCCIÓN Y SERVICIOS  
ESCUELA PROFESIONAL DE INGENIERÍA  
DE SISTEMAS



**Formato:** Guía de Práctica de Laboratorio / Talleres / Centros de Simulación

**Aprobación:** 2023/03/01

**Código:** GUIA-PRLD-001

**Página:** 1

## INFORME DE LABORATORIO

### INFORMACIÓN BÁSICA

ASIGNATURA: Laboratorio de Programación Web 2 - Grupo "B"

TÍTULO DE LA  
PRÁCTICA: Laboratorio 8

NÚMERO DE PRÁCTICA: 08      AÑO LECTIVO: 2023 A      NRO. SEMESTRE: III

FECHA DE PRESENTACIÓN: 25-Jul-2023      HORA DE PRESENTACIÓN: 23:59

INTEGRANTES:

1. Layme Mamani, Gonzalo Rail

NOTA:

DOCENTE:

Mg. ANIBAL SARDON

### SOLUCIÓN Y RESULTADOS

#### I. SOLUCIÓN DE EJERCICIOS/PROBLEMAS

Repositorio general

[https://github.com/GonzaloRail/Laboratorio\\_08\\_PWEB2](https://github.com/GonzaloRail/Laboratorio_08_PWEB2)

- Proyecto: Lista de Tareas Solucion:  
[https://github.com/GonzaloRail/Laboratorio\\_08\\_PWEB2/tree/main/listaDetareas](https://github.com/GonzaloRail/Laboratorio_08_PWEB2/tree/main/listaDetareas)
- Menu

Raw dataHTML form

Título

Descripcion

Completada

☐

POST

Detalle de las listas

Lista Tareas / Detalle Tarea

Detalle TareaDELETEOPTIONSGET

GET /tareas/1/

HTTP 200 OK  
Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS  
Content-Type: application/json  
Vary: Accept

{  
 "id": 1,  
 "titulo": "tarea",  
 "descripcion": "pweb lab 8",  
 "completada": false  
}

Raw dataHTML form

Título

tarea

Descripcion

pweb lab 8

Completada

☐

PUT

Create

Lista TareasOPTIONSGET

GET /tareas/

HTTP 200 OK  
Allow: GET, POST, HEAD, OPTIONS  
Content-Type: application/json  
Vary: Accept

[  
 {  
 "id": 1,  
 "titulo": "tarea",  
 "descripcion": "pweb lab 8",  
 "completada": false  
 },  
 {  
 "id": 2,  
 "titulo": "tarea",  
 "descripcion": "pweb lab8",  
 "completada": true  
 }  
]

Read

Raw data

HTML form

Titulo

Recordatorio

Descripcion

Hacer la tarea

Completada

☒

Make a POST request on the Lista Tareas resource

POST

• Update

Django REST framework

Lista Tareas / Detalle Tarea

DELETEOPTIONSGET

Detalle Tarea

GET /tareas/2/

HTTP 200 OK

Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

{  
 "id": 2,  
 "titulo": "tarea",  
 "descripcion": "pweb lab8",  
 "completada": true  
}

Raw data

HTML form

Titulo

tarea

Descripcion

pweb lab8 pero para mañana

Completada

☒

PUT

• Delete

Django REST framework

Lista Tareas / Detalle Tarea

DELETEOPTIONSGET

Detalle Tarea

GET /tareas/1/

HTTP 200 OK

Allow: GET, PUT, PATCH, DELETE, HEAD, OPTIONS

Content-Type: application/json

Vary: Accept

{  
 "id": 1,  
 "titulo": "tarea",  
 "descripcion": "pweb lab 8",  
 "completada": false  
}

Raw data

HTML form

Titulo

tarea

Descripcion

pweb lab 8

Completada

☐

PUT

Are you sure you want to delete this Detalle Tarea?  
CancelDelete

- Demostracion del codigo
- Models.py--- Se crea la base de datos, ingresando su parametros

```
1  from django.db import models
2
3
4  class Tarea(models.Model):
5      titulo = models.CharField(max_length=200)
6      descripcion = models.TextField()
7      completada = models.BooleanField(default=False)
8
9      def __str__(self):
10         return self.titulo
11
```

- Views.py--- Se crea views para la funcionalidad, para ver la lista y ver detalladamente la lista

```
1  from rest_framework import generics
2  from .models import Tarea
3  from .serializers import TareaSerializer
4
5  class ListaTareasView(generics.ListCreateAPIView):
6      queryset = Tarea.objects.all()
7      serializer_class = TareaSerializer
8
9  class DetalleTareaView(generics.RetrieveUpdateDestroyAPIView):
10     queryset = Tarea.objects.all()
11     serializer_class = TareaSerializer
12
```

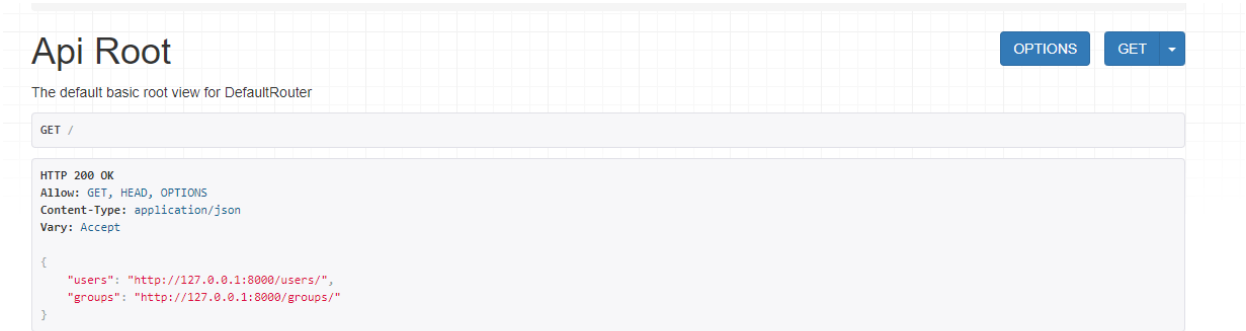
- Serializers.py

```
1  from rest_framework import serializers
2  from .models import Tarea
3
4  class TareaSerializer(serializers.ModelSerializer):
5      class Meta:
6          model = Tarea
7          fields = '__all__'
8
```

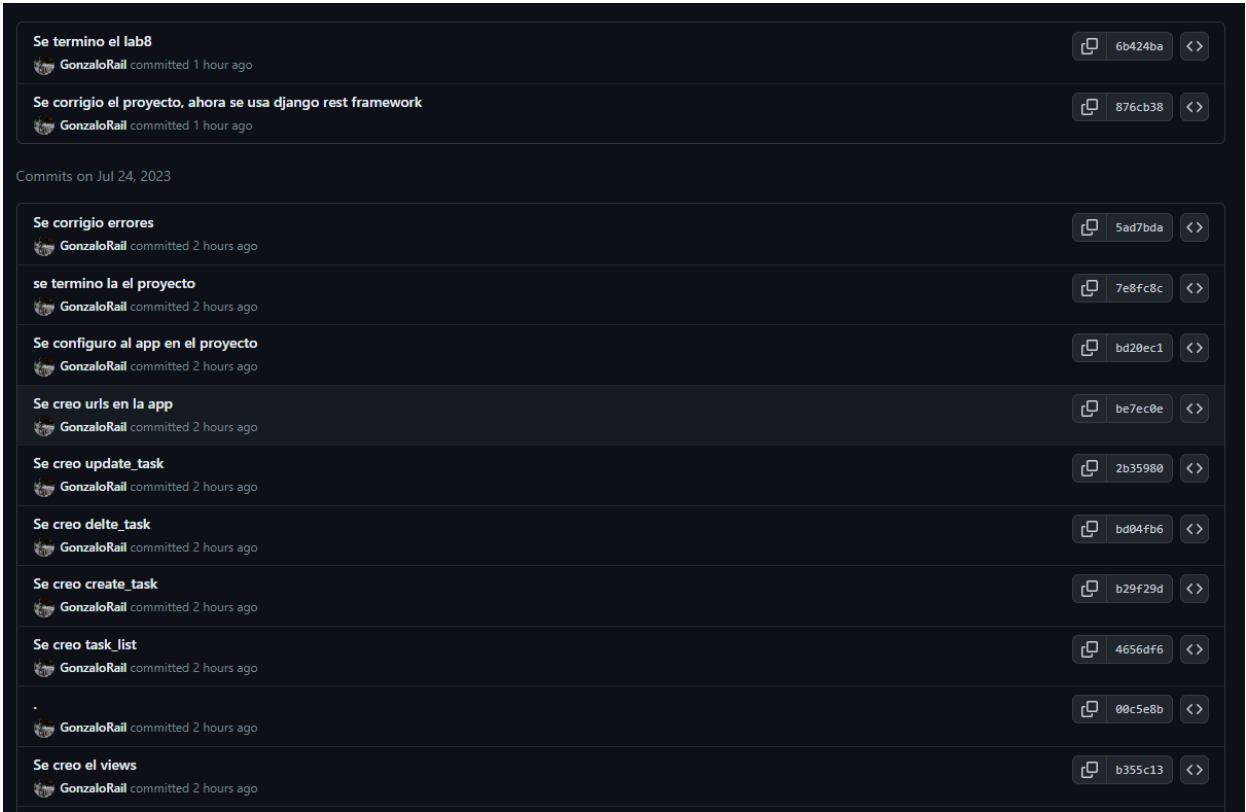
- Urls.py--- Vemos las urls y podremos movernos de pagina en pagina

```
1 from django.urls import path
2 from .views import ListaTareasView, DetalleTareaView
3
4 urlpatterns = [
5     path('tareas/', ListaTareasView.as_view(), name='lista_tareas'),
6     path('tareas/<int:pk>/', DetalleTareaView.as_view(), name='detalle_tarea'),
7 ]
8
```

- 
- Proyecto: Tutorial  
Solucion: [https://github.com/GonzaloRail/Laboratorio\\_08\\_PWEB2/tree/main/tutorial/tutorial](https://github.com/GonzaloRail/Laboratorio_08_PWEB2/tree/main/tutorial/tutorial)
- tutorial



- 
- Commits



PREGUNTA

- ¿Cual fue la mayor dificultad del uso de este framework?.

Al principio, puede ser abrumador configurar correctamente Django REST framework, especialmente para aquellos que son nuevos en el desarrollo de API REST. Configurar las vistas, serializadores, autenticación, permisos y enrutamiento puede llevar tiempo y requerir un entendimiento sólido de cómo funciona cada componente.

- Rubrica

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

## REFERENCIAS Y BIBLIOGRAFÍA

- <https://www.django-rest-framework.org/>
- <https://www.django-rest-framework.org/tutorial/quickstart/>
- <https://www.django-rest-framework.org/tutorial/1-serialization/>
- <https://www.django-rest-framework.org/tutorial/3-class-based-views/>
- <https://www.django-rest-framework.org/api-guide/authentication/>
- <https://www.bezkoder.com/django-rest-api/>
- <https://stackoverflow.com/questions/70319606/importerror-cannot-import-name-url-from-django-conf-urls-after-upgrading-to>