
	<p style="text-align: center;">UNIVERSIDAD NACIONAL DE SAN AGUSTIN</p> <p style="text-align: center;">FACULTAD DE INGENIERÍA DE PRODUCCIÓN Y SERVICIOS</p> <p style="text-align: center;">DEPARTAMENTO ACADÉMICO DE INGENIERÍA DE SISTEMAS E INFORMÁTICA</p> <p style="text-align: center;">ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS</p>	
Formato: Guía de Práctica de Laboratorio		
Aprobación: 2022/03/01	Código: GUIA-PRLD-001	Página: 1

**GUÍA DE LABORATORIO  
INFORMACIÓN BÁSICA**

ASIGNATURA:	Programación Web 2
TÍTULO DE LA PRÁCTICA:	Django Rest Framework
NÚMERO DE PRÁCTICA:	07 — — — — — AÑO LECTIVO: 2023 A — — — — — NRO. SEMESTRE: III
FECHA INICIO:	26-Junio-2023 — — — FECHA FIN: 03-Julio-2023 — — — DURACIÓN: 04 horas
Profesor:	ANIBAL SARDON
Alumno:	Muñoz Romani Aroldo Guillermo

### 1. Competencias del curso

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

### 2. Resultado del estudiante

- RE. 2 La capacidad de aplicar diseño de ingeniería para producir soluciones a problemas y diseñar sistemas, componentes o procesos para satisfacer necesidades específicas dentro de consideraciones realistas en los aspectos de salud pública, seguridad y bienestar; factores globales, culturales, sociales, económicos y ambientales.
- RE. 8 La capacidad de crear, seleccionar y utilizar técnicas, habilidades, recursos y herramientas modernas de ingeniería y tecnologías de la información, incluyendo la predicción y el modelamiento, con una comprensión de las limitaciones.

### 3. Equipos, materiales y temas

- Sistema Operativo (GNU/Linux de preferencia).
- GNU Vim.
- Python 3.
- Git.
- Cuenta en GitHub con el correo institucional.
- Entorno virtual.
- Django 4.
- djangorestframework.

#### 4. Directorio de trabajo

- Cree su directorio de trabajo.
- Luego, dirijase a este directorio, para clonar su repositorio y continuar sus practicas.

Listing 1: Creando directorio de trabajo

---

```
$ mkdir -p $HOME/rescobedoq/
```

---

Listing 2: Dirijiendonos al directorio de trabajo

---

```
$ cd $HOME/rescobedoq/
```

---

Listing 3: Clonando repositorio GitHub

---

```
$ git clone [URL_DE_SU_GITHUB_PRIVADO]
```

---

Listing 4: Creando directorio para laboratorio

---

```
$ mkdir -p $HOME/rescobedoq/pw2-lab-23a/lab07/exercises/
```

---

- Siempre evalúe utilizar el archivo `.gitignore` para no considerar algunos archivos innecesarios sobre todo para el repositorio GitHub.
- Pueden haber varios de estos archivos y estar ubicados estratégicamente; por ejemplo solo para un laboratorio en particular.

Listing 5: Creando `.gitignore`

---

```
$ vim $HOME/rescobedoq/pw2-lab-23a/lab07/.gitignore
```

---

Listing 6: Ejemplo de `.gitignore`

---

```
myenv/bin/ *  
myenv/lib/ *  
myenv/src/_pycache_/*.*pyc
```

---

- Estudie el archivo `.gitignore` del proyecto Library :
- <https://github.com/mdn/django-locallibrary-tutorial/blob/main/.gitignore>

#### 5. Marco teórico

##### 5.1 Django Rest Framework

- Django REST framework es un conjunto de herramientas potente y flexible para crear API web.
- Algunas razones por las que podría querer usar el marco REST:
  - La API navegable por la Web es una gran ganancia de usabilidad para sus desarrolladores.
  - Políticas de autenticación que incluyen paquetes para OAuth1 y OAuth2.
  - Serialización que admite fuentes de datos ORM y no ORM.
  - Personalizable hasta el final: solo use las vistas regulares basadas en funciones si no necesita las funciones más potentes.
  - Amplia documentación y gran apoyo de la comunidad.
  - Utilizado y confiado por empresas reconocidas internacionalmente, como Mozilla, Red Hat, Heroku y Eventbrite.

##### 5.2 Crear un directorio para el entorno virtual de Django

- Para crear un ambiente elija en qué directorio se va a crear el entorno virtual.
- No envíe estos entornos virtuales al GitHub, omitalos en `.gitignore`. Pero no olvide tener `requirements.txt` en el root del Proyecto.

Listing 7: Creando directorio para entorno virtual en Unix

---

```
$ mkdir -p $HOME/rescobedoq/pw2-lab-23a/lab07/myenv
```

---

##### 5.3 Crear entorno virtual en un directorio

- En este directorio crear un entorno virtual ejecutando el siguiente comando:

Listing 8: Creando entorno virtual en GNU/Linux

---

```
$ cd $HOME/rescobedoq/pw2-lab-23a/lab07/myenv
$virtualenv -ppython3.
```

---

Listing 9: Creando entorno virtual en MS Windows

---

```
C: python -m venv c:2-lab-23a07env
```

---

Listing 10: Creando entorno virtual en MacOS

---

```
$ python -m venv $HOME/rescobedoq/pw2-lab-23a/lab07/myenv
```

---

#### 5.4 Activando entorno virtual

- En el directorio de trabajo active el entorno virtual ejecutando el script `activate`.
- Sea cual sea nuestro sistema operativo sabremos que el entorno virtual se ha activado porque su nombre aparece entre parentesis delante del prompt.

Listing 11: Activando entorno virtual en GNU/Linux

---

```
$ cd $HOME/rescobedoq/pw2-lab-23a/lab07/exercises/
$ source ../../myenv/bin/activate
(myenv)user@localhost : $
```

---

Listing 12: Activando entorno virtual en MS Windows

---

```
C:2-lab-23a07env.bat
```

---

Listing 13: Activando entorno virtual en GNU/Linux

---

```
$ source $HOME/rescobedoq/pw2-lab-23a/lab07/myenv/bin/activate
```

---

#### 5.5 Desactivando entorno virtual

- El comando para desactivar el entorno virtual es identico para Windows, macOS y Linux:

Listing 14: Desactivando entorno virtual

---

```
$ deactivate
```

---

#### 5.6 Instalando Django y Django Rest Framework dentro del entorno virtual

- Requisitos.
- El framework REST requiere lo siguiente:
  - Python (3.6, 3.7, 3.8, 3.9, 3.10)
  - Django (2.2, 3.0, 3.1, 3.2, 4.0, 4.1)
- Recomendamos encarecidamente y solo admitimos oficialmente la ultima version del parche de cada serie de Python y Django.
- Los siguientes paquetes son opcionales:
  - PyYAML , writemplate (5.1+, 3.0.0+) : Compatibilidad con la generaci on de esquemas.
  - Markdown (3.0.0+) : compatibilidad con Markdown para la API navegable.
  - Pygments (2.4.0+) : agregue resaltado de sintaxis al procesamiento de Markdown.
  - django-filter (1.0.1+) : Soporte de filtrado.
  - django-guardian (1.1.1+) : Soporte de permisos de nivel de objeto.

Listing 15: Mostrando paquetes instalados en el entorno virtual

---

```
(myenv)user@localhost : $pip list
PackageVersion
-----
pip22.0.4
setuptools62.1.0
wheel0.37.1
```

---

- Instalamos con pip: (Puede incluir los paquetes opcionales que desee)

Listing 16: Instando Django

---

```
(myenv)user@localhost : $pip install django
(myenv)user@localhost : $pip install django rest framework
(myenv)user@localhost : $pip install markdown
(myenv)user@localhost : $pip install django - filter
```

---

- Volvemos a listar los paquetes instalados:

Listing 17: Mostrando Django instalado en el entorno virtual

```
(myenv)user@localhost : $pip list
PackageVersion
-----
asgiref3.7.2
Django4.2.3
django-filter23.2
djangorestframework3.14.0
Markdown3.4.3
pip23.0.1
pytz2023.3
setuptools66.1.1
sqlparse0.4.4
wheel0.38.4
```

- Importante: Como el entorno virtual no se clona, por temas de espacio. Es necesario sacarle un backup en un archivo requirements.txt

Listing 18: Creando requirements.txt

```
(myenv)user@localhost : $pip freeze > requirements.txt
```

- Importante: El archivo requirements.txt, servir a para volver a construir el entorno virtual, preservando los paquetes con sus versiones.

Listing 19: Instalando desde requirements.txt

```
(myenv)user@localhost : $pip install -r requirements.txt
```

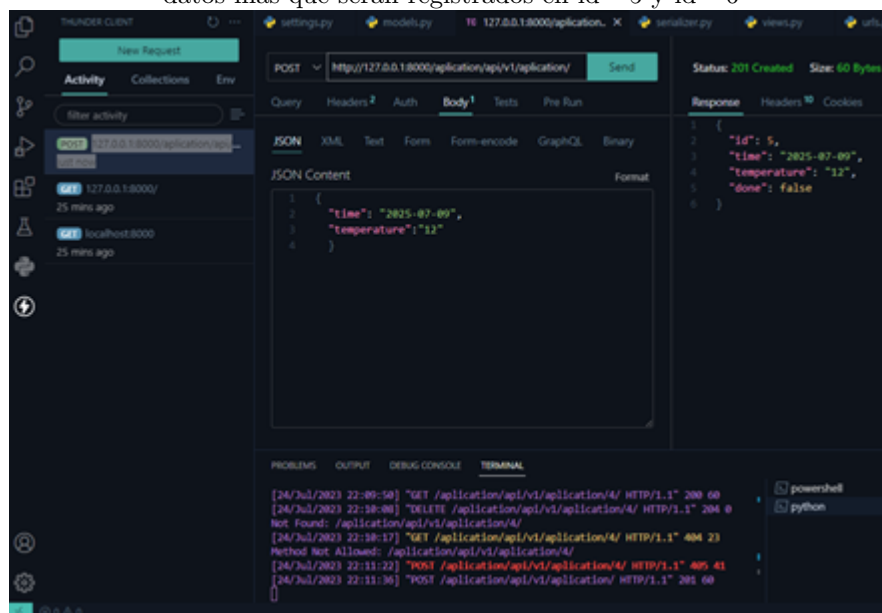
## 6. Ejercicios

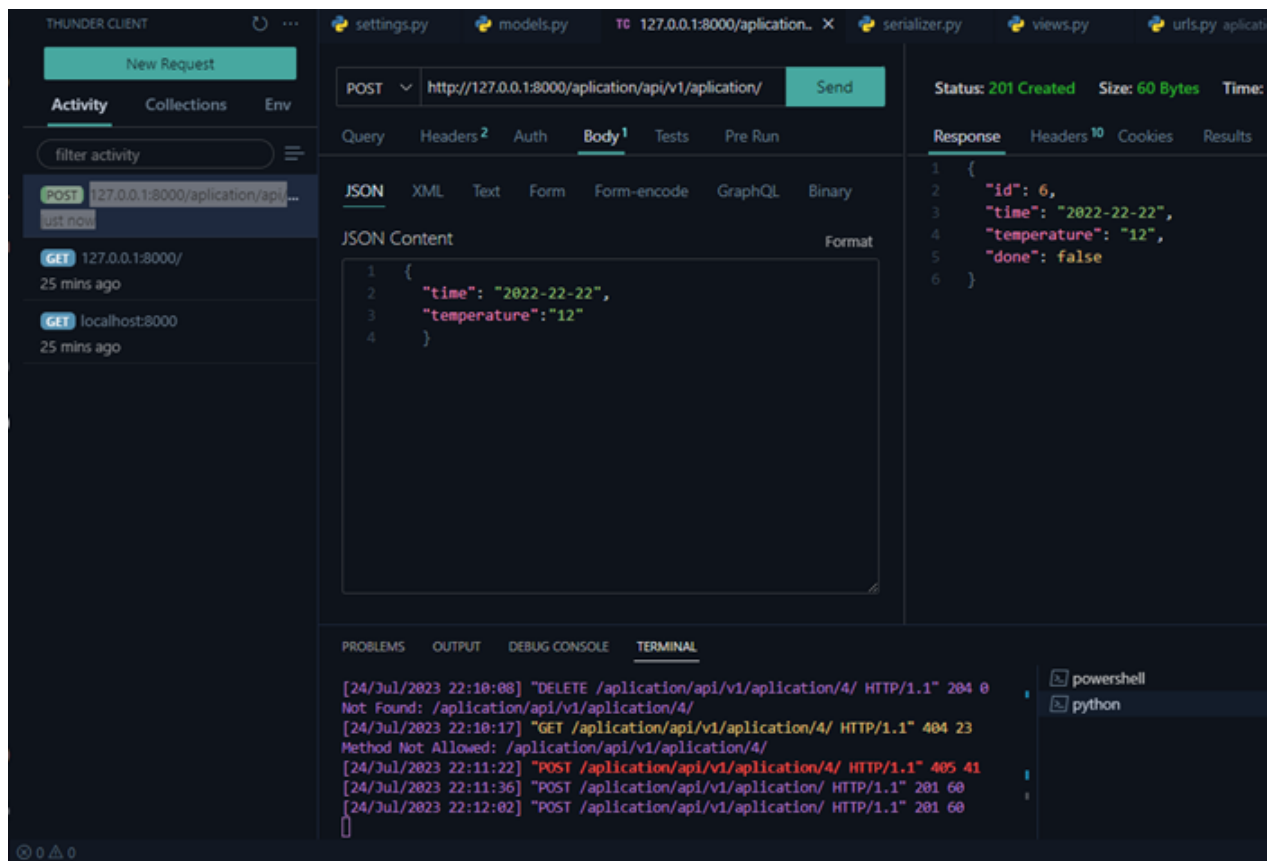
- Practique desarrollando el ejercicio de iniciacion:
- <https://www.django-rest-framework.org/tutorial/quickstart/>
- Para consumir el web-service puede usar el cliente SOAP UI Community: <https://www.soapui.org/downloads/soapui/>

## 7. Tarea

- En sus grupos de trabajo correspondientes. Elabore un servicio web que tenga un CRUD con el uso de este framework.
- Create - POST

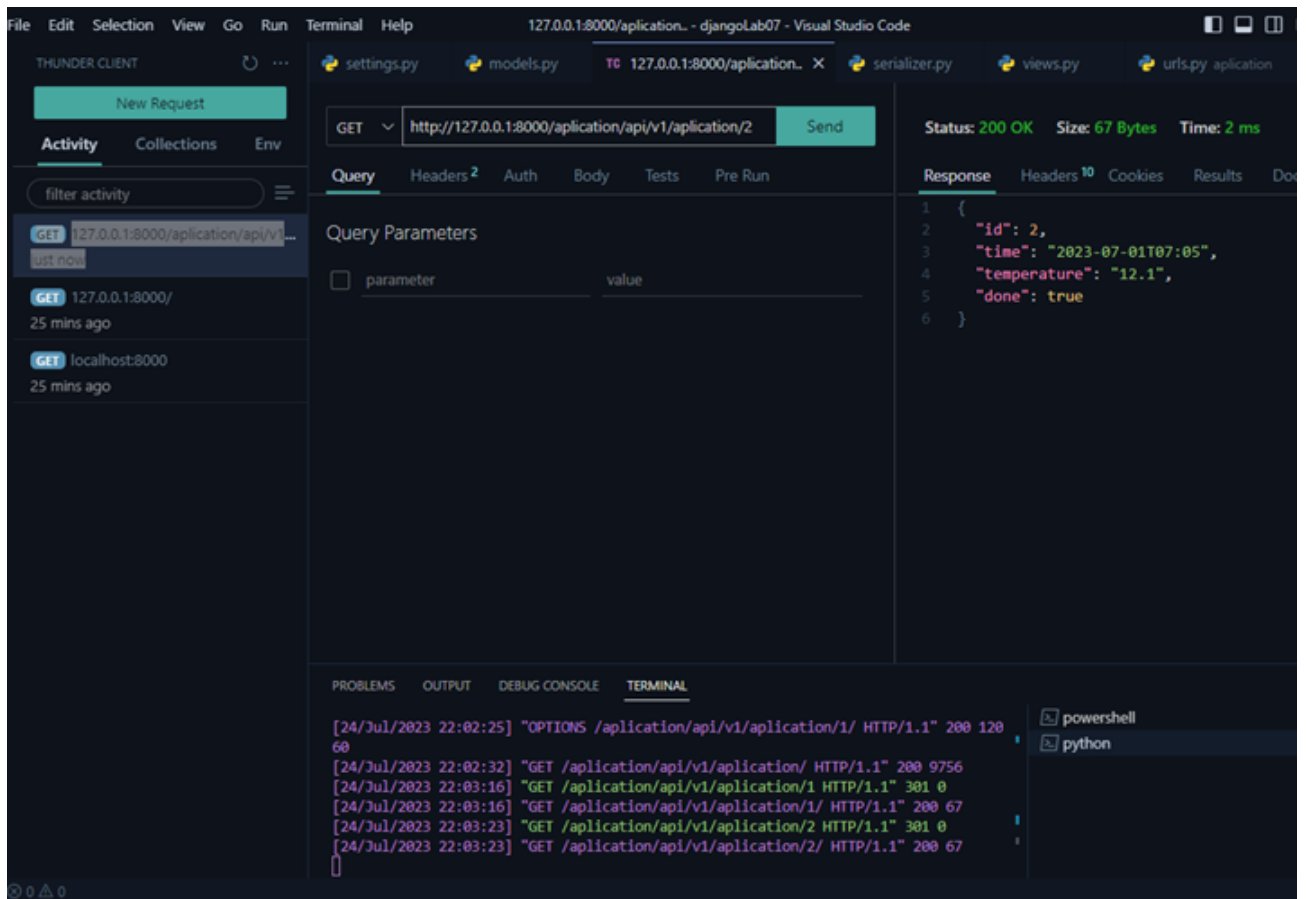
en la caja de texto ponemos la direccion `http://127.0.0.1:8000/aplication/api/v1/aplication` y agregamos 2 datos mas que seran registrados en `id =5` y `id =6`





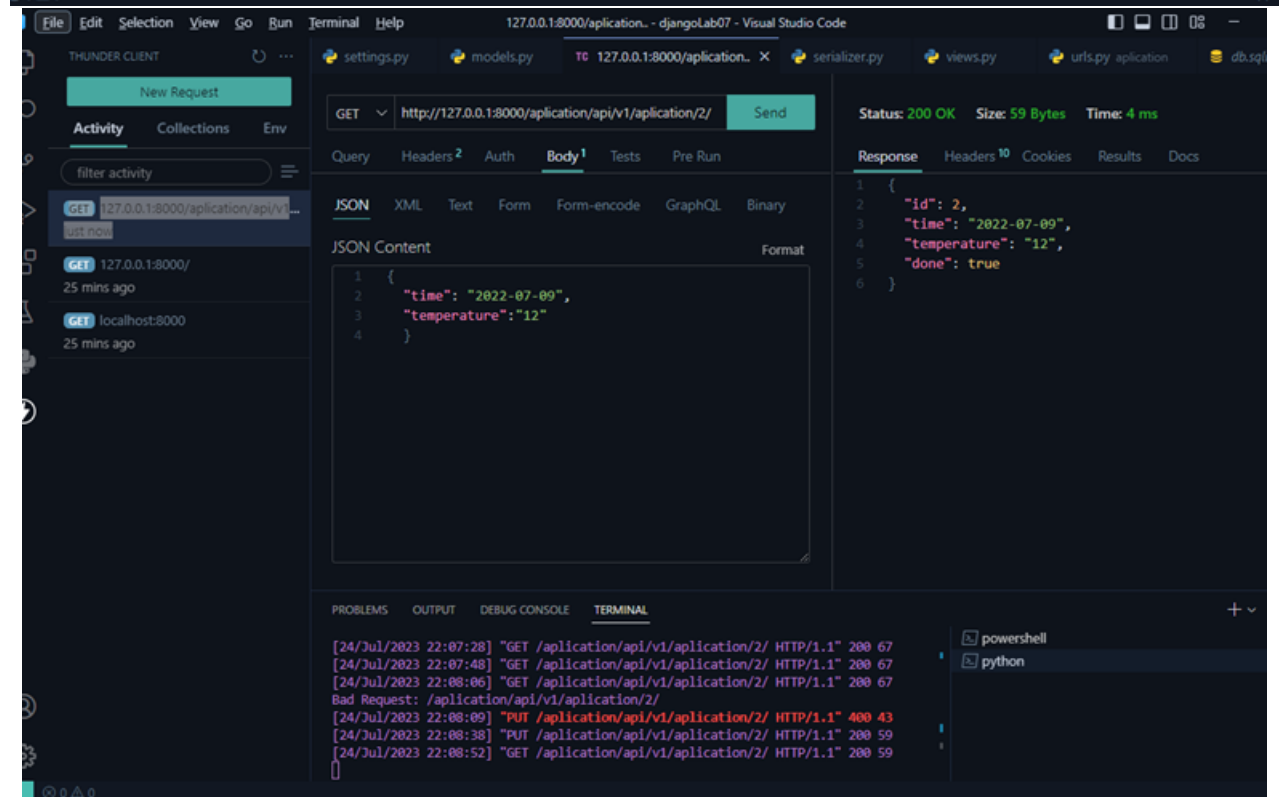
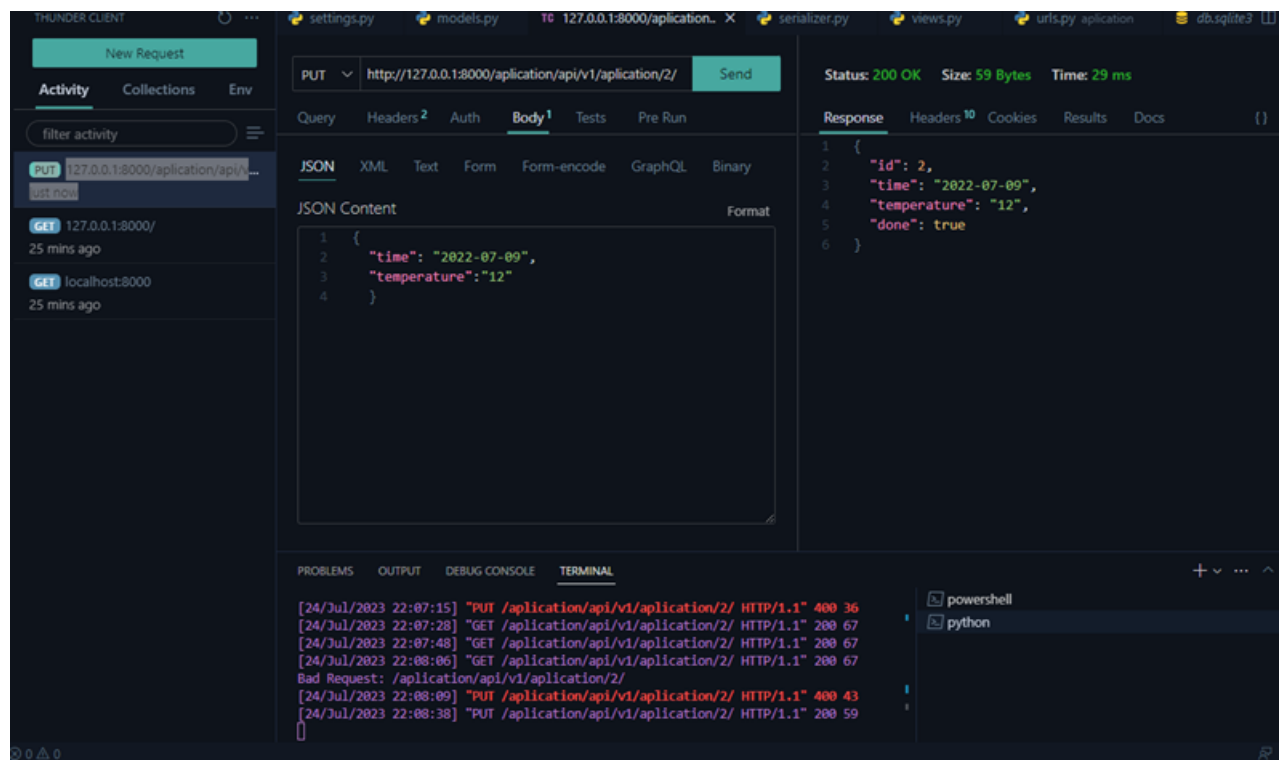
- Read - GET

ponemos en la caja de texto `http://127.0.0.1:8000/aplicacion/api/v1/aplicacion/2` para ver el contenido del tiempo guardado en el `id= 2`



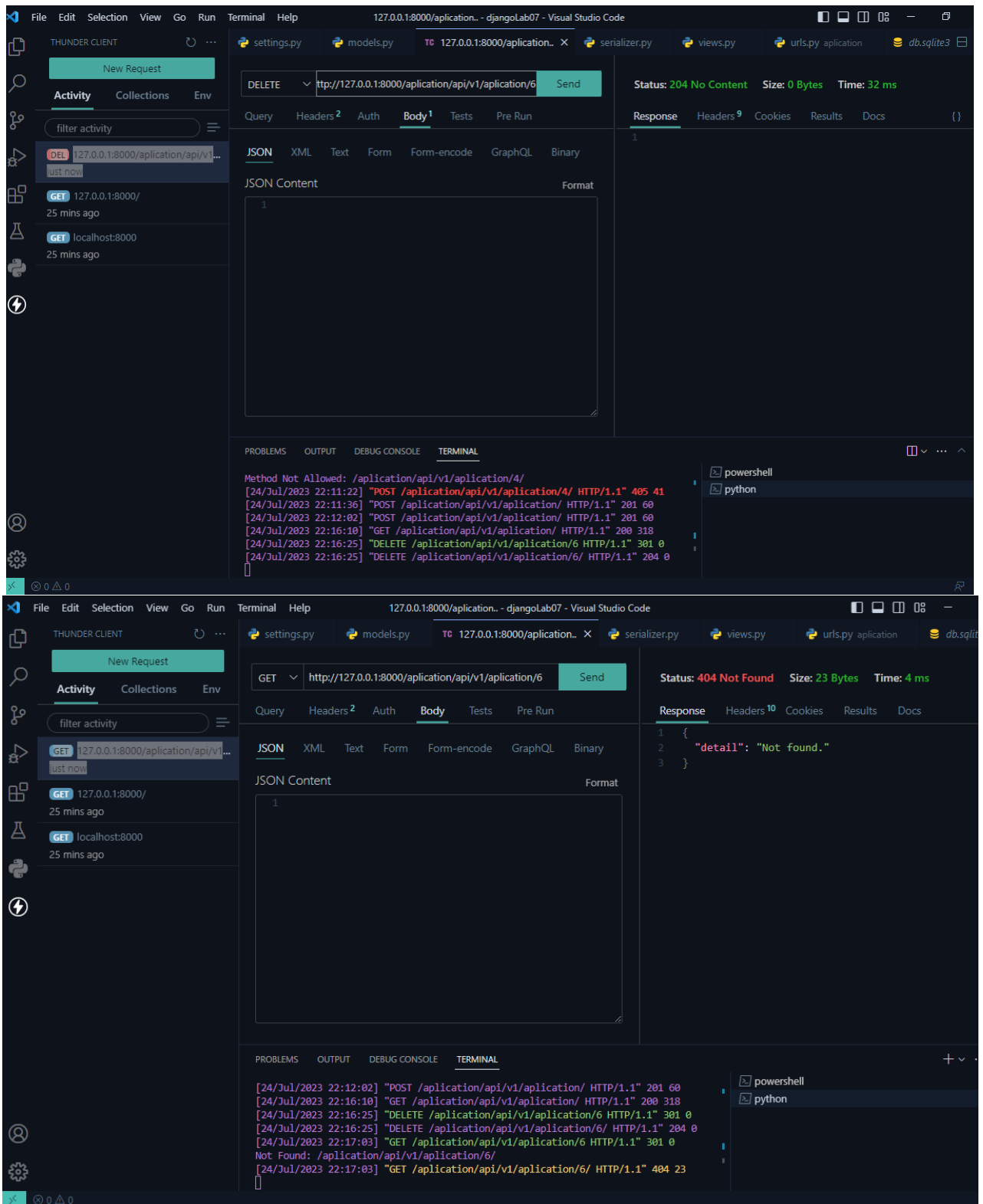
- Update - PUT

En la caja de texto vamos a modificar los datos del id=2 usando `http://127.0.0.1:8000/aplication/api/v1/aplication/2` luego en pestaña body y json y agregamos en la caja de texto el nuevo time y la nueva temperatura para luego click en send, la siguiente imagen es el dato antiguo del id=2 y la imagen siguiente visualizamos con get para ver la modificacion que realizamos



- Delete - DELETE

Ahora eliminaremos el contenido del id=6 con el siguiente `http://127.0.0.1:8000/application/api/v1/application/6`, la primera imagen es la ejecucion y la siguiente imagen es la verificacion que nos muestra detalles de no found



- Centrarse en el Core business de su aplicacion web. Los mas importante y necesario que este disponible a traves de un servicio web.
- Ejemplos: <https://reqbin.com/>, <https://www.googleapis.com/youtube/v3/playlistItems>
- Muestre la funcionalidad consumiendola desde el cliente Rest de su preferencia.
- El metodo GET puede ser directamente consumido por un navegador web:
- Por ejemplo: En esta API se puede obtener la temperatura de Arequipa en un rango de fechas: (La version



gratuita tiene un retraso de 7 dias, por tanto solo mostrara la temperatura en Arequipa desde el 01 de Julio hasta el 03 de Julio)

- [https://archive-api.open-meteo.com/v1/archive?latitude=-16.39889longitude=-71.535start\\_date=2023-07-01end\\_date=2023-07-10hourly=temperature2mdaily=temperature2mmax,temperature2mintimezone=America%2FNew\\_York](https://archive-api.open-meteo.com/v1/archive?latitude=-16.39889longitude=-71.535start_date=2023-07-01end_date=2023-07-10hourly=temperature2mdaily=temperature2mmax,temperature2mintimezone=America%2FNew_York)

## 8. Pregunta

- ¿Cual fue la mayor dificultad del uso de este framework?. la mayor dificultad que se podria observar es la sintaxis del uso de las ejecuciones, en un caso aparecia error porque faltó un / en la ejecucion ademas que a veces se requiere para modificar un dato se necesita de todos los datos presentes para modificarlo ya que a veces no reconoce al modificar un segundo elemento sin ver el primer elemento

## 9. Entregables

- El informe debe tener un enlace al directorio especifico del laboratorio en su repositorio GitHub privado donde este todo el codigo fuente y otros que sean necesarios. Evitar la presencia de archivos: binarios, objetos, archivos temporales, cache, librerias, entornos virtuales. Si hay configuraciones particulares puede incluir archivos de especificacion como: requirements.txt, o leeme.txt.
- No olvide que el profesor debe ser siempre colaborador a su repositorio (Usuario del profesor @rescobedoq).
- Para ser considerado con la calificacion de maxima nota, el informe debe estar elaborado en LATEX
- Usted debe describir solo los commits mas importantes que marcaron hitos en su trabajo, adjutando capturas de pantalla, del commit, del codigo fuente, de sus ejecuciones y pruebas.
- En el informe siempre se debe explicar las imagenes (codigo fuente, capturas de pantalla, commits, ejecuciones, pruebas, etc.) con descripciones puntuales pero precisas.
- Partes de entrega:
  - Django orientado a Usuarios finales.
  - Tema para clientes, con funcionalidades importantes.
  - Recomendaciones: CRUD en uno de los procesos para el cliente final.
  - Ejemplo 1: Para la WebApp Library (Sistema de Biblioteca virtual), seria .El cliente reserva un ejemplar de un libro determinado”.
  - Ejemplo 2: Para la WebApp Enrollments (Sistema para Inscripciones en laboratorios), seria. El alumno se inscribe en un laboratorio disponible para un curso”.

## 10. Rubricas

### 10.1 Rubrica para entregable Informe

Tabla 1: Rubrica para tipo de Informe

Informe		Cumple	No cumple
<b>Latex</b>	El informe está en formato PDF desde Latex, con un formato limpio (buena presentación) y facil de leer.	20	0
<b>MarkDown</b>	El informe está en formato PDF desde MarkDown README.md, con un formato limpio (buena presentación) y facil de leer.	17	0
<b>MS Word</b>	El informe está en formato PDF desde plantilla MS Word, con un formato limpio (buena presentación) y facil de leer.	15	0
<b>Observaciones</b>	Por cada observación se le descontará puntos.	-	-

### 10.2 Rubrica para el contenido del Informe y demostracion

- El alumno debera marcar o dejar en blanco en las celdas de la columna Checklist, de acuerdo a si cumple o no con el item correspondiente.
- Si un alumno supera la fecha de entrega, su calificación siempre sea sobre la nota mínima aprobada, siempre y cuando cumpla con todos los items.
- El alumno debe autocalificarse en la columna Estudiante de acuerdo a la tabla de calificación de niveles de desempeño:

Tabla 2: Niveles de desempeño

Puntos	Nivel			
	Insatisfactorio 25 %	En Proceso 50 %	Satisfactorio 75 %	Sobresaliente 100 %
<b>2.0</b>	0.5	1.0	1.5	2.0
<b>4.0</b>	1.0	2.0	3.0	4.0

Tabla 3: Rubrica para contenido del Informe y demostración

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 están 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>		<b>20</b>			

## 11. Referencias

- <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>