

Laboratorio API REST

Nombre: Teresa Jesús Tapia Soto

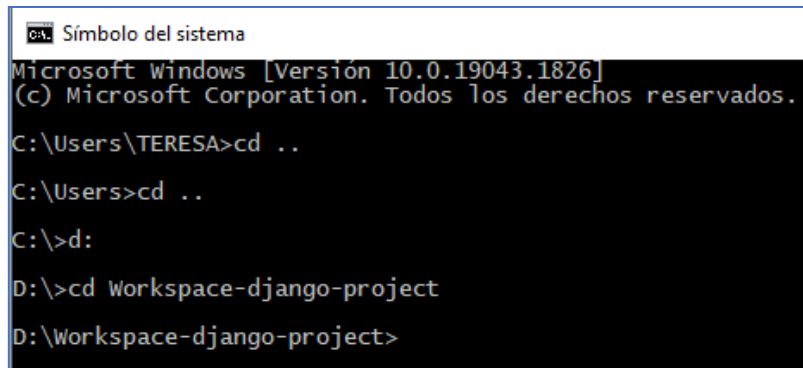
Crear el Proyecto **ProyectoRestBasico** y una aplicación **apiRestDemo**

INSTRUCCIONES DEL PASO A PASO:

1. Abrir CMD
2. Instalar el framework api rest

pip install **django**restframework

3. Ubicarse dentro del espacio de trabajo **workspace-django-project**



```
Símbolo del sistema
Microsoft Windows [Versión 10.0.19043.1826]
(c) Microsoft Corporation. Todos los derechos reservados.
C:\Users\TERESA>cd ..
C:\Users>cd ..
C:\>d:
D:\>cd Workspace-django-project
D:\Workspace-django-project>
```

4. Crear el proyecto **ProyectoRestBasico** ejecutando la siguiente instrucción:

django-admin startproject **ProyectoRestBasico**

```
D:\Workspace-django-project>django-admin startproject ProyectoRestBasico
```

5. Entrar al proyecto **ProyectoRestBasico**

```
D:\Workspace-django-project>cd ProyectoRestBasico
D:\Workspace-django-project\ProyectoRestBasico>
```

6. Crear la aplicación **apiRestDemo** ejecutando la siguiente instrucción:

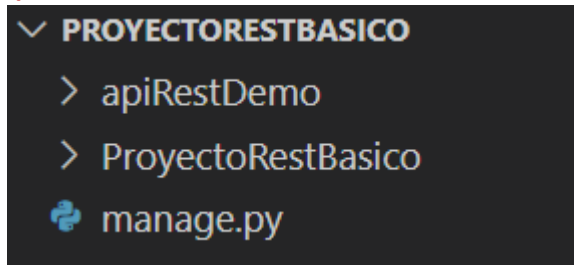
python manage.py startapp **apiRestDemo**

```
D:\Workspace-django-project\ProyectoRestBasico>python manage.py startapp apiRestDemo
D:\Workspace-django-project\ProyectoRestBasico>
```

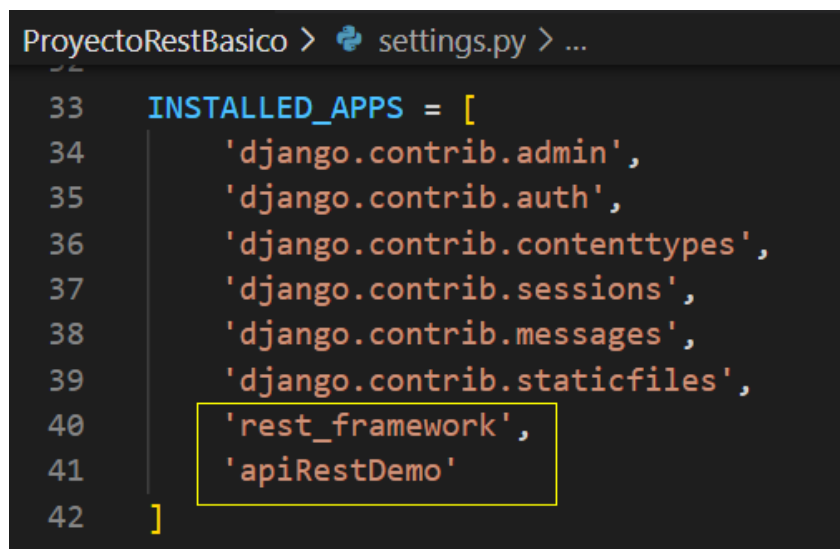
7. Abrir visual code

```
D:\Workspace-django-project\ProyectoRestBasico>code .
```

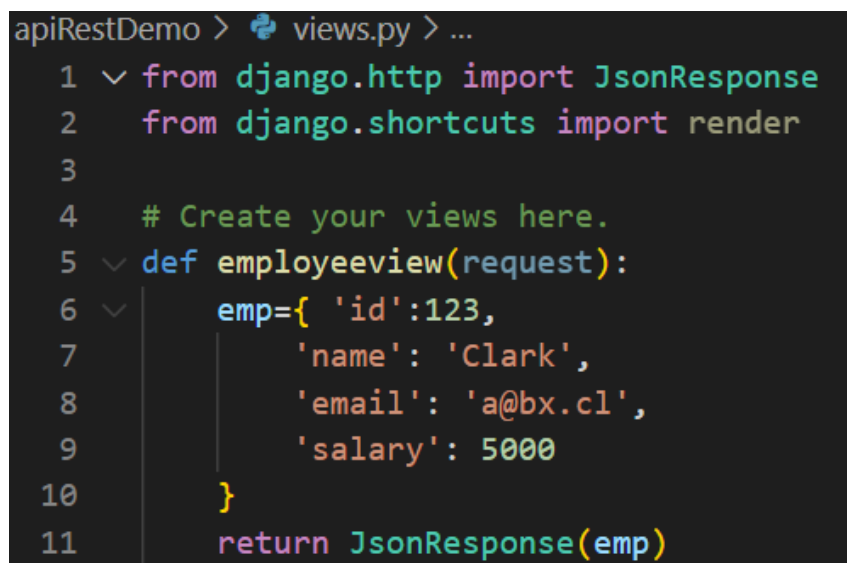
En la siguiente imagen, se puede visualizar el proyecto **ProyectoRestBasico** y la aplicación **apiRestDemo**



8. Configurar **settings.py** agregando la **aplicación** y el **framework rest**



9. Configurar views.py. Se define una vista que retorna una salida json



10. Configurar urls.py

```
ProyectoRestBasico > urls.py > ...
15 """
16 from django.contrib import admin
17 from django.urls import path
18 from apiRestDemo.views import employeeview
19
20 urlpatterns = [
21     path('admin/', admin.site.urls),
22     path('employees/', employeeview)
23 ]
```

11. Subir el servidor

```
PS D:\Workspace-django-project\ProyectoRestBasico> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).

You have 18 unapplied migration(s). Your project may not work properly until you apply the migrations for app(s): admin, auth, contenttypes, sessions.
Run 'python manage.py migrate' to apply them.
December 05, 2022 - 23:01:33
Django version 4.1, using settings 'ProyectoRestBasico.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

12. Acceder a la vista creada. <http://127.0.0.1:8000/employees/>

```
→ 127.0.0.1:8000/employees/

// 20221205230348
// http://127.0.0.1:8000/employees/

{
  "id": 123,
  "name": "Clark",
  "email": "a@bx.cl",
  "salary": 5000
}
```

Ejemplo 2: Acceso Base de datos....

13. Bajar el servidor....
14. Crear Modelo en archivo models.py. Específicamente, crear clase Employee

```
apiRestDemo > models.py > Employee > __str__
1  from django.db import models
2
3  # Create your models here.
4  class Employee(models.Model):
5      id =models.IntegerField(primary_key=True)
6      name =models.CharField(max_length=50)
7      email =models.CharField(max_length=50)
8      salary=models.DecimalField(max_digits=10,decimal_places=2)
9
10     def __str__(self):
11         return str(self.id) + " " + self.name + "($ " + str(self.salary) + ")"
```

15. Se define nueva vista en views.py. Se agrega import del modelo o clase Employee

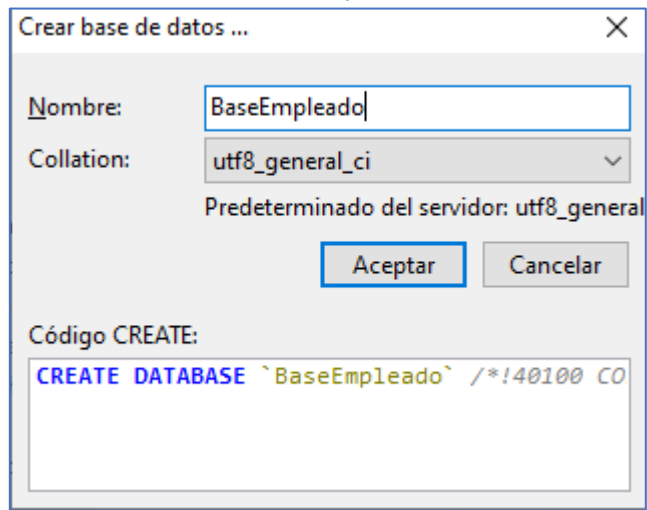
```
from apiRestDemo.models import Employee
```

```
def employeeviewBD(request):
    empleados=Employee.objects.all()
    data={'Employees': list(empleados.values('name', 'salary'))}
    return JsonResponse(data)
```

16. Se define nueva url en urls.py. Se agrega import de la vista employeeviewBD

```
ProyectoRestBasico > urls.py > ...
15  """
16  from django.contrib import admin
17  from django.urls import path
18  from apiRestDemo.views import employeeview, employeeviewBD
19
20  urlpatterns = [
21      path('admin/', admin.site.urls),
22      path('employees/', employeeview),
23      path('employeesBD/', employeeviewBD)
24  ]
```

17. Crear base de datos [BaseEmpleado](#)



Crear base de datos ...

Nombre: BaseEmpleado

Collation: utf8_general_ci
Predeterminado del servidor: utf8_general

Aceptar Cancelar

Código CREATE:

```
CREATE DATABASE `BaseEmpleado` /*!40100 CO
```

18. Configurar la base de datos en [settings.py](#)

```
import pymysql
pymysql.install_as_MySQLdb()

DATABASES = {
    'default': {
        'ENGINE': 'django.db.backends.mysql',
        'NAME': 'BaseEmpleado',
        'USER': 'root',
        'PASSWORD': 'root1502'
    }
}
```

19. Realizar las migraciones ejecutando la siguiente instrucción:

`python manage.py makemigrations`

```
PS D:\Workspace-django-project\ProyectoRestBasico> python manage.py makemigrations

Migrations for 'apiRestDemo':
  apiRestDemo\migrations\0001_initial.py
    - Create model Employee
PS D:\Workspace-django-project\ProyectoRestBasico> |
```

20. Ejecutar las migraciones ejecutando la siguiente instrucción:

```
python manage.py migrate
```

```
PS D:\Workspace-django-project\ProyectoRestBasico> python manage.py migrate
Operations to perform:
  Apply all migrations: admin, apiRestDemo, auth, contenttypes, sessions
Running migrations:
  Applying contenttypes.0001_initial... OK
  Applying auth.0001_initial... OK
  Applying admin.0001_initial... OK
  Applying admin.0002_logentry_remove_auto_add... OK
  Applying admin.0003_logentry_add_action_flag_choices... OK
  Applying apiRestDemo.0001_initial... OK
  Applying contenttypes.0002_remove_content_type_name... OK
  Applying auth.0002_alter_permission_name_max_length... OK
  Applying auth.0003_alter_user_email_max_length... OK
  Applying auth.0004_alter_user_username_opts... OK
  Applying auth.0005_alter_user_last_login_null... OK
  Applying auth.0006_require_contenttypes_0002... OK
  Applying auth.0007_alter_validators_add_error_messages... OK
  Applying auth.0008_alter_user_username_max_length... OK
  Applying auth.0009_alter_user_last_name_max_length... OK
  Applying auth.0010_alter_group_name_max_length... OK
  Applying auth.0011_update_proxy_permissions... OK
  Applying auth.0012_alter_user_first_name_max_length... OK
  Applying sessions.0001_initial... OK
PS D:\Workspace-django-project\ProyectoRestBasico>
```

21. Ingresar datos a la base de datos...manualmente

id	name	email	salary
1	teresa	a@a.d	1.000,00
2	carlos	b@a.d	1.500,00
3	Juan	aaa@ad.d	2.000,00

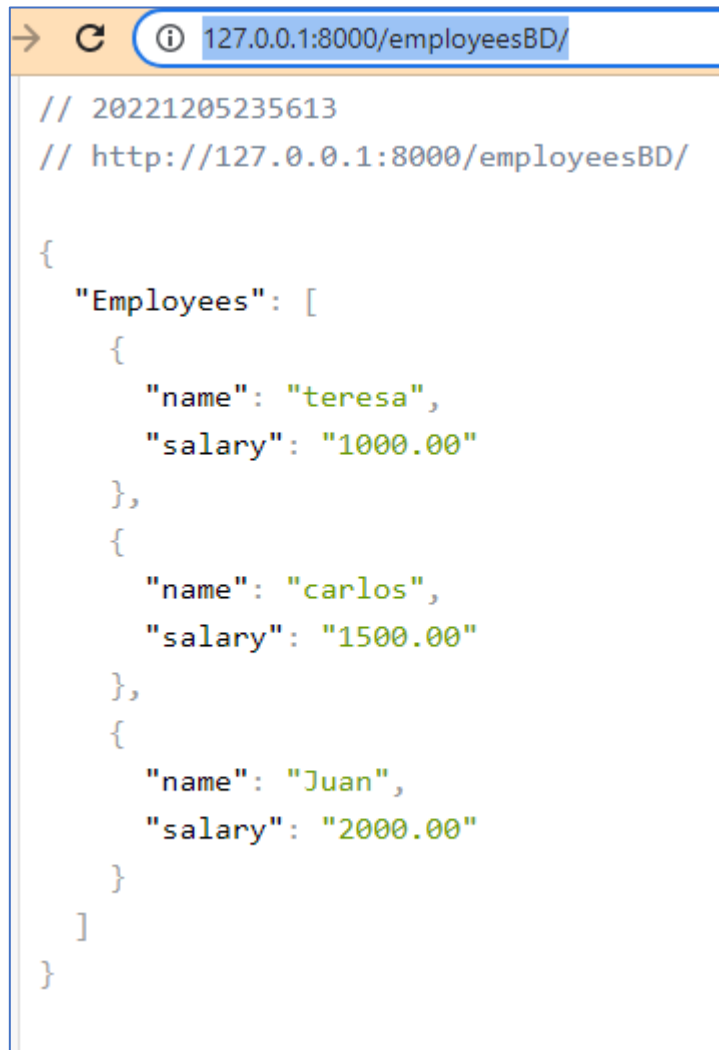
22. Subir el servidor

```
PS D:\Workspace-django-project\ProyectoRestBasico> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 05, 2022 - 23:55:25
Django version 4.1, using settings 'ProyectoRestBasico.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

23. Acceder a la vista creada.

<http://127.0.0.1:8000/employeesBD/>



```
// 20221205235613
// http://127.0.0.1:8000/employeesBD/

{
  "Employees": [
    {
      "name": "teresa",
      "salary": "1000.00"
    },
    {
      "name": "carlos",
      "salary": "1500.00"
    },
    {
      "name": "Juan",
      "salary": "2000.00"
    }
  ]
}
```

Ejemplo 3: Acceso Base de datos....

24. Crear aplicación serialApp

```
D:\Workspace-django-project\ProyectoRestBasico>python manage.py startapp serialApp
D:\Workspace-django-project\ProyectoRestBasico>
```

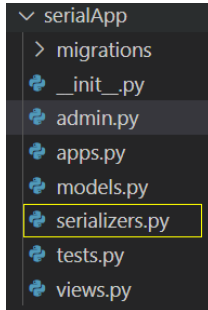
25. Configurar `settings.py` agregando la aplicación `serialApp`

```
INSTALLED_APPS = [
    'django.contrib.admin',
    'django.contrib.auth',
    'django.contrib.contenttypes',
    'django.contrib.sessions',
    'django.contrib.messages',
    'django.contrib.staticfiles',
    'rest_framework',
    'apiRestDemo',
    'serialApp'
]
```

26. Creamos el `modelo` (en `model.py`) en la aplicación `serialApp`

```
serialApp > models.py > ...
1  from django.db import models
2
3  # Create your models here.
4  class Student(models.Model):
5      id =models.IntegerField(primary_key=True)
6      name =models.CharField(max_length=50)
7      email =models.CharField(max_length=50)
8      score=models.DecimalField(max_digits=5,decimal_places=2)
9
10     def __str__(self):
11         return str(self.id) + " " + self.name + "(SCORE: " + str(self.score) + ")"
12
```


27. Creamos el archivo `serializers` (`serializers.py`) en la aplicación `serialApp`



```
serialApp > serializers.py > ...
1  from rest_framework import serializers
2  from serialApp.models import Student
3
4  class StudentSerializer(serializers.ModelSerializer):
5      class Meta:
6          model = Student
7          fields = "__all__"
```

28. Creamos la vista `student_list` para implementar los métodos `GET` y `POST`

```
serialApp > views.py > reservas_detail
1  from django.shortcuts import render
2  from serialApp.models import Student
3  from serialApp.serializers import StudentSerializer
4  from rest_framework.response import Response
5  from rest_framework import status
6  from rest_framework.decorators import api_view
7
8  # Create your views here.
9  @api_view(['GET', 'POST'])
10 def student_list(request):
11     if request.method == 'GET':
12         students = Student.objects.all()
13         ser = StudentSerializer(students, many=True)
14         return Response(ser.data)
15
16     if request.method == 'POST':
17         ser = StudentSerializer(data=request.data)
18         if ser.is_valid():
19             ser.save()
20             return Response(ser.data, status=status.HTTP_201_CREATED)
21         return Response(ser.errors, status=status.HTTP_400_BAD_REQUEST)
```

29. Creamos la vista `student_detail` para implementar los métodos `GET`, `PUT` y `DELETE`

```
serialApp > views.py > student_list
23 @api_view(['GET', 'PUT', 'DELETE'])
24 def student_detail(request, pk):
25     try:
26         student = Student.objects.get(pk=pk)
27     except:
28         return Response(status=status.HTTP_404_NOT_FOUND)
29
30     if request.method == 'GET':
31         ser = StudentSerializer(student)
32         return Response(ser.data)
33
34     if request.method == 'PUT':
35         ser = StudentSerializer(student, data=request.data)
36         if ser.is_valid():
37             ser.save()
38             return Response(ser.data)
39         return Response(ser.errors, status=status.HTTP_400_BAD_REQUEST)
40
41     if request.method == 'DELETE':
42         student.delete()
43         return Response(status=status.HTTP_204_NO_CONTENT)
```

30. Configurar las rutas en el archivo `urls.py`

```
ProyectoRestBasico > urls.py > ...
15 """
16 from django.contrib import admin
17 from django.urls import path
18 from apiRestDemo.views import employeeview, employeeviewBD
19 from serialApp.views import student_list, student_detail
20
21 urlpatterns = [
22     path('admin/', admin.site.urls),
23     path('employees/', employeeview),
24     path('employeesBD/', employeeviewBD),
25     #api rest
26     path('apistudents/', student_list),
27     path('apistudents/<int:pk>', student_detail),
28
29 ]
```

31. Realizar las migraciones ejecutando la siguiente instrucción:

```
python manage.py makemigrations
```

```
PS D:\Workspace-django-project\ProyectoRestBasico> python manage.py makemigrations

Migrations for 'serialApp':
  serialApp\migrations\0001_initial.py
    - Create model Student
PS D:\Workspace-django-project\ProyectoRestBasico>
```

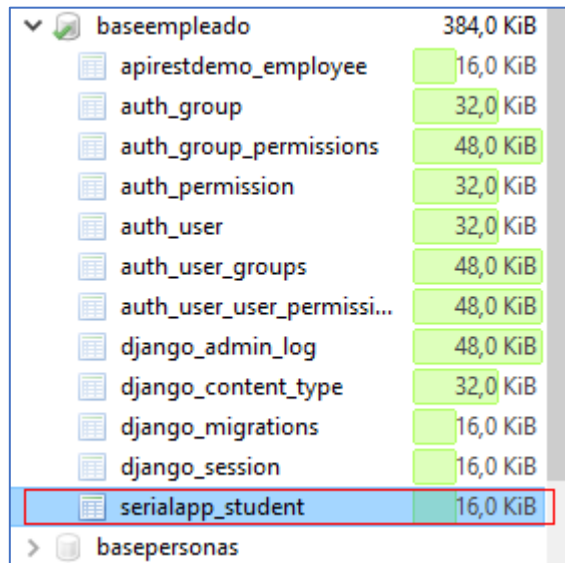
32. Ejecutar las migraciones ejecutando la siguiente instrucción:

```
python manage.py migrate
```

```
PS D:\Workspace-django-project\ProyectoRestBasico> python manage.py migrate

Operations to perform:
  Apply all migrations: admin, apiRestDemo, auth, contenttypes, serialApp, session
s
Running migrations:
  Applying serialApp.0001_initial... OK
PS D:\Workspace-django-project\ProyectoRestBasico>
```

33. Ingresar datos a la base de datos...manualmente



A screenshot of a database management tool showing a table with 3 rows of data. The table has columns: id, name, email, and score. The data is as follows:

id	name	email	score
1	teresa	a@a.d	100,00
2	carlos	b@b.d	200,00
3	mario	c@c.d	300,00

34. Subir el servidor

```
PS D:\Workspace-django-project\ProyectoRestBasico> python manage.py runserver
Watching for file changes with StatReloader
Performing system checks...

System check identified no issues (0 silenced).
December 06, 2022 - 10:21:08
Django version 4.1, using settings 'ProyectoRestBasico.settings'
Starting development server at http://127.0.0.1:8000/
Quit the server with CTRL-BREAK.
```

35. Acceder a la vista creada.

<http://127.0.0.1:8000/apistudents/>

Browser view of the Django REST framework API endpoint `/apistudents/`.

URL: `127.0.0.1:8000/apistudents/`

Method: `GET`

Response:

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

[
  {
    "id": 1,
    "name": "teresa",
    "email": "a@a.cl",
    "score": "100.00"
  },
  {
    "id": 2,
    "name": "carlos",
    "email": "b@b.cl",
    "score": "200.00"
  },
  {
    "id": 3,
    "name": "mario",
    "email": "c@c.cl",
    "score": "300.00"
  }
]
```

127.0.0.1:8000/apistudents/2

Django REST framework

Student List / Student Detail

Student Detail

DELETEOPTIONSGET

GET /apistudents/2

HTTP 200 OK

Allow: OPTIONS, PUT, GET, DELETE

Content-Type: application/json

Vary: Accept

```
{
  "id": 2,
  "name": "carlos",
  "email": "b@b.cl",
  "score": "200.00"
}
```

Media type: application/json

Content:

PUT