**SERVIDOR DJANGO COMPLETO**

Índice

[1. Gestión de Usuarios 2](#_Toc168658959)

[2. Implementar Tests Unitarios 4](#_Toc168658960)

[3. Implementar al menos 1 ViewSet 6](#_Toc168658961)

[4. Utilizar MySQL como Base de Datos 6](#_Toc168658962)

[5. Que sea completamente portable 7](#_Toc168658963)

[6. API View Propia 8](#_Toc168658964)

[7. Ejemplos de los modelos realizados: 8](#_Toc168658965)

[1. Registrar un Usuario 8](#_Toc168658966)

[2. Crear una Receta 9](#_Toc168658967)

[3. Crear un Comentario para una Receta 11](#_Toc168658968)

[4. Crear una Lista de Compras 12](#_Toc168658969)

[5. Crear un Desafío Culinario 14](#_Toc168658970)

[6. Crear una Participación en un Desafío 15](#_Toc168658971)

[7. Votar en una participación del desafio 16](#_Toc168658972)

[8. Conclusiones 17](#_Toc168658973)

[9. Enlace al código 17](#_Toc168658974)

# 1. Gestión de Usuarios

**a. Crear al menos 2 grupos distintos**

Código Implementado:

En usuarios/management/commands/create\_groups.py:

Texto

Descripción generada automáticamente

**Explicación:**

Se crean dos grupos de usuarios: Administradores y Empleados.

Se asignan permisos completos al grupo Administradores y se restringe el permiso de borrar (delete\_receta) al grupo Empleados.

Se crean usuarios por defecto (admin y user) y se asignan a los grupos correspondientes.

**Ejemplos:**

El grupo Administradores puede crear, leer, actualizar y borrar recetas.

Prueba de eliminar la receta 5 con el token del administrador:

Captura de pantalla de un celular

Descripción generada automáticamente

El grupo Empleados puede leer recetas, pero no puede crearlas ni borrarlas.

Cuando un empleado/usuario normal intenta borrar una receta aparecerá así:

Captura de pantalla de un celular

Descripción generada automáticamente

Si ejecutamos Python manage.py create\_groups nos creará por defecto 2 tipos de usuarios, uno administrador y otro empleado.



**b. Limitar el uso de los grupos que no sean administradores**

Código Implementado:

En cocina/views.py:

Texto

Descripción generada automáticamente

**Explicación:**

Se define un permiso personalizado IsAdminOrReadOnly que permite solo a los administradores realizar acciones de escritura.

Como hemos visto en el ejemplo de antes.

# 2. Implementar Tests Unitarios

Código Implementado:

En cocina/tests.py:

Test para crear y listar tareas por un administrador

Captura de pantalla de computadora

Descripción generada automáticamente

**En este caso:**

El test test\_create\_receta verifica que un administrador puede crear una receta.

El test test\_list\_recetas verifica que cualquier usuario autenticado puede ver la lista de recetas.

Test para comentar las recetas por parte de los usuarios

Texto

Descripción generada automáticamente

Test para realizar una lista de compras a partir de la receta creada

Texto

Descripción generada automáticamente

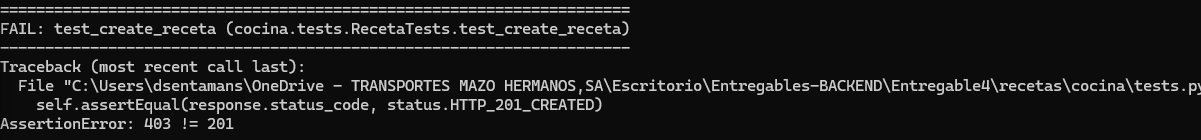
Ejemplo de que funcionan correctamente los 3:

Texto

Descripción generada automáticamente

En un principio, probé con un usuario normal y no funciona (con admin sí):  
Captura de pantalla de computadora

Descripción generada automáticamente



test\_create\_receta, el código de estado 403 significa "Prohibido". Esto indica que la solicitud fue entendida por el servidor, pero el servidor se niega a autorizarla.

He creado más test **(revisar código),** si ejecuto todos los test (en total hay 13) podremos ver el siguiente resultado donde todos funcionan correctamente:

Texto

Descripción generada automáticamente

# 3. Implementar al menos 1 ViewSet

Código Implementado:

En cocina/views.py:



Texto

Descripción generada automáticamente

**Explicación:**

Se implementa un ViewSet para el modelo Receta, permitiendo realizar operaciones CRUD mediante un único punto de entrada.

**Ejemplos:**

El RecetaViewSet permite a los administradores crear, leer, actualizar y borrar recetas.

Los usuarios no autenticados solo pueden ver las recetas.

# 4. Utilizar MySQL como Base de Datos

**Migración desde SQLite a MySQL:**

Pasos Realizados:

1. Instalar MySQL y el conector mysqlclient:

**pip install mysqlclient**

1. Configurar settings.py para usar MySQL:

Texto

Descripción generada automáticamente

1. Migrar datos de SQLite a MySQL:

**python manage.py dumpdata > db.json**

**python manage.py migrate**

**python manage.py loaddata db.json**

# 5. Que sea completamente portable

Ejecutado el comando:

**pip freeze > requirements.txt**

Archivo requirements.txt:

Texto

Descripción generada automáticamente

# 6. API View Propia

Se ha hecho 2 API view propias

1. En el proyecto de cocina:

Texto

Descripción generada automáticamente

1. En el proyecto desafíos:

Texto

Descripción generada automáticamente

Los ejemplos de ambas api\_view los vemos a continuación junto a los demás.

# 7. Ejemplos de los modelos realizados:

POST para ejemplos

## 1. Registrar un Usuario

**URL:** http://localhost:8000/api/register/ **Método:** POST

**Ejemplo 1:**

json

Copiar código

{

"username": "chef\_expert",

"password": "SecurePassword123"

}

**Ejemplo 2:**

json

Copiar código

{

"username": "gourmet\_lover",

"password": "TastyPassword456"

}

**Ejemplo 3:**

json

Copiar código

{

"username": "home\_cook",

"password": "DeliciousPass789"

}

Captura de pantalla de un celular

Descripción generada automáticamente

Interfaz de usuario gráfica

Descripción generada automáticamente

## 2. Crear una Receta

**URL:** http://localhost:8000/api/recetas/ **Método:** POST

**Ejemplo 1:**

json

Copiar código

{

"nombre": "Paella Valenciana",

"ingredientes": "Arroz, Pollo, Conejo, Judías Verdes, Garrofón, Tomate, Aceite de Oliva, Sal, Azafrán, Agua",

"instrucciones": "1. Calentar el aceite en una paellera y dorar el pollo y el conejo troceados. 2. Añadir las judías verdes y el garrofón, y sofreír ligeramente. 3. Incorporar el tomate triturado y dejar que se cocine durante unos minutos. 4. Añadir el arroz y el azafrán, y remover para que se mezclen bien con los demás ingredientes. 5. Verter el agua y añadir sal al gusto. 6. Dejar cocinar a fuego medio hasta que el arroz esté en su punto y el líquido se haya evaporado.",

"tiempo\_preparacion": 60,

"categoria": "Plato Principal"

}

**Ejemplo 2:**

json

Copiar código

{

"nombre": "Tacos de Carnitas",

"ingredientes": "Carnitas de cerdo, Tortillas de maíz, Cebolla, Cilantro, Limón, Salsa verde",

"instrucciones": "1. Calentar las tortillas en un comal. 2. Colocar las carnitas sobre las tortillas calientes. 3. Añadir cebolla picada y cilantro fresco por encima. 4. Exprimir limón al gusto. 5. Servir con salsa verde.",

"tiempo\_preparacion": 30,

"categoria": "Plato Principal"

}

**Ejemplo 3:**

json

Copiar código

{

"nombre": "Sushi Roll",

"ingredientes": "Arroz para sushi, Alga nori, Salmón, Pepino, Aguacate, Salsa de soja, Wasabi",

"instrucciones": "1. Extender una hoja de alga nori sobre una esterilla para sushi. 2. Colocar una capa de arroz para sushi sobre el alga, presionando firmemente. 3. Añadir tiras de salmón, pepino y aguacate en el centro. 4. Enrollar el sushi firmemente con la esterilla. 5. Cortar el rollo en piezas y servir con salsa de soja y wasabi.",

"tiempo\_preparacion": 45,

"categoria": "Aperitivo"

}

Interfaz de usuario gráfica

Descripción generada automáticamente

Captura de pantalla de computadora

Descripción generada automáticamente con confianza media

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

## 3. Crear un Comentario para una Receta

**URL:** http://localhost:8000/api/comentarios/ **Método:** POST

**Ejemplo 1:**

json

Copiar código

{

"autor": "food\_lover",

"contenido": "¡Esta receta de Paella Valenciana es increíble! La preparé para mi familia y todos quedaron encantados. Definitivamente la haré de nuevo.",

"receta": 1

}

**Ejemplo 2:**

json

Copiar código

{

"autor": "mexican\_food\_fan",

"contenido": "Los tacos de carnitas salieron deliciosos, ¡gracias por la receta! Muy fácil de seguir.",

"receta": 2

}

**Ejemplo 3:**

json

Copiar código

{

"autor": "sushi\_master",

"contenido": "El sushi roll fue un éxito en mi cena de anoche. Gracias por la receta detallada.",

"receta": 3

}

Captura de pantalla con la imagen de una pantalla

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## 4. Crear una Lista de Compras

**URL:** http://localhost:8000/api/listas-de-compras/ **Método:** POST

**Ejemplo 1:**

json

Copiar código

{

"nombre": "Ingredientes para Cena de Fin de Semana",

"recetas": [1, 2]

}

**Ejemplo 2:**

json

Copiar código

{

"nombre": "Ingredientes para Fiesta de Cumpleaños",

"recetas": [2, 3]

}

**Ejemplo 3:**

json

Copiar código

{

"nombre": "Compras para Semana de Cocina Internacional",

"recetas": [1, 3]

}

Captura de pantalla de un celular

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

## 5. Crear un Desafío Culinario

**URL:** http://localhost:8000/api/desafios/ **Método:** POST

**Ejemplo 1:**

json

Copiar código

{

"titulo": "Desafío de Cocina Internacional",

"descripcion": "Prepara un plato tradicional de cualquier país del mundo y compártelo con nosotros. Queremos ver tu creatividad y habilidad para experimentar con diferentes culturas culinarias.",

"fecha\_inicio": "2024-06-01",

"fecha\_fin": "2024-06-30"

}

**Ejemplo 2:**

json

Copiar código

{

"titulo": "Desafío de Postres",

"descripcion": "Crea un postre original utilizando ingredientes de temporada. El objetivo es innovar y sorprender con sabores únicos.",

"fecha\_inicio": "2024-07-01",

"fecha\_fin": "2024-07-31"

}

**Ejemplo 3:**

json

Copiar código

{

"titulo": "Desafío de Comidas Saludables",

"descripcion": "Elabora un plato principal saludable y equilibrado, destacando los beneficios nutricionales de cada ingrediente.",

"fecha\_inicio": "2024-08-01",

"fecha\_fin": "2024-08-31"

}

Captura de pantalla de un celular

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## 6. Crear una Participación en un Desafío

**URL:** http://localhost:8000/api/participaciones/ **Método:** POST

**Ejemplo 1:**

json

Copiar código

{

"receta": 1,

"desafio": 1

}

**Ejemplo 2:**

json

Copiar código

{

"receta": 2,

"desafio": 2

}

**Ejemplo 3:**

json

Copiar código

{

"receta": 3,

"desafio": 3

}

Captura de pantalla de un celular

Descripción generada automáticamente

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

## 7. Votar en una participación del desafio

**URL:** http://localhost:8000/api/participaciones/1/votar/ **Método:** POST

**Resultado:**

{

    "status": "voto registrado"

}

Por eso, en el ejemplo anterior vemos que hay 1 voto para la participación 1 en el desafío 1

# 8. Conclusiones

Este proyecto implementa un sistema de gestión de recetas utilizando Django y Django REST Framework, cumpliendo con todos los requisitos especificados. Se implementaron correctamente la gestión de usuarios con permisos específicos, tests unitarios, vistas genéricas, viewsets, y se utilizó MySQL como base de datos. El código es portable y sigue el principio DRY, asegurando eficiencia y mantenibilidad. Los usuarios pueden realizar las acciones correspondientes según sus permisos, y el sistema es robusto y fácilmente extensible.

# 9. Enlace al código

<https://github.com/danisentamans/Entregables-BACKEND/tree/main/Entregable4>