

1. Introducción

- **Objetivo del documento:** Este documento tiene como objetivo principal de documentar las pruebas funcionales realizadas a la API “**Gestión de Clientes, Vehículos y Autenticación**”, con esto se busca garantizar la calidad y el correcto funcionamiento de la API, asegurando su correcto funcionamiento.
- **Alcance:** El presente documento abarca las pruebas de los siguientes componentes de la API.
 - **Endpoints:** Se enumeran los endpoints a probar.
 - /clientes.
 - /usuarios
 - /usuarios/login
 - /usuarios/{usuario}
 - /vehículos
 - **Métodos HTTP:**
 - POST
 - GET
 - PUT
 - DELETE
 - **Flujos de autenticación:** Sin autenticación.
 - **Validación de datos:** [Indicar los tipos de datos a validar, por ejemplo: formatos, rangos, tipos de datos]
- **Criterios de aceptación:**

Una prueba es exitosa si cumple con los siguientes criterios:

- **Funcionalidad:** La API responde a las solicitudes de acuerdo con la documentación y los requisitos funcionales.
- **Códigos de estado HTTP:** Los códigos de estado HTTP devueltos por la API son los esperados para cada solicitud.
- **Datos de respuesta:** Los datos devueltos por la API son correctos, completos y están en el formato esperado.
- **Manejo de errores:** La API maneja correctamente los errores y devuelve mensajes de error claros y concisos.
- **Tiempo de respuesta:** La API responde a las solicitudes dentro de los tiempos de respuesta establecidos.
- **Versiones:**
 - **API:** Versión 1.0.0

2. Descripción de la API

- **Propósito:** La API tiene como objetivo principal proporcionar un conjunto de endpoints encargados de gestionar un spa de autos y motos donde se ofrecen diferentes servicios para mejorar la eficiencia y experiencia del cliente.

- **Endpoints:**
 - /clientes
 - GET: Obtiene el listado de los clientes o un solo cliente si se pasa el número de documento.
 - POST: Crea un cliente nuevo.
 - PUT: Actualiza un cliente.
 - DELETE: Elimina un cliente del registro y recibe el número de documento.
 - /usuarios
 - GET: Obtiene el listado de usuarios o uno solo si se pasa el “usuario”.
 - POST: Crea un nuevo usuario
 - /login POST: Inicia sesión.
 - PUT: Actualiza un usuario y se pasa como parámetro el usuario.
 - DELETE: Elimina un usuario y se debe pasar como parámetro el “usuario”
 - /vehículos
 - GET: Obtiene el listado de los vehículos o uno, pero se debe pasar la placa y el documento del cliente.
 - POST: Crea un vehículo nuevo.
 - PUT: Actualiza un vehículo.
 - DELETE: Elimina un vehículo del registro y el parámetro que se pasa es la placa.

3. Casos de Prueba

- **Matriz de trazabilidad:**

Requisito funcional	Caso de prueba
Obtener lista de clientes	TC001
Obtener un cliente en específico	TC002
Registrar un nuevo cliente	TC003
Actualizar un cliente	TC004
Eliminar un cliente	TC005
Obtener lista de los usuarios	TC006
Obtener un usuario en específico	TC007
Registrar un usuario nuevo	TC008
Realizar el login de un usuario	TC009
Actualizar un usuario	TC010
Eliminar un usuario	TC011
Obtener lista de los vehículos	TC012
Obtener un vehículo en específico	TC013
Registrar un vehículo nuevo	TC014
Actualizar un vehículo	TC015
Eliminar un vehículo	TC016
Obtener Configuración	TC017
Actualizar configuración	TC018
Restablecer la configuración	TC019
Obtener las promociones	TC020
Crear una promoción	TC021
Actualizar una promoción	TC022
Eliminar una promoción	TC023
Consultar Factura	TC024
Crear Factura	TC025
Actualizar factura	TC026
Eliminar factura	TC027
Obtener los servicios	TC028

Crear un servicio	TC029
Actualizar un servicio	TC030
Eliminar Servicio	TC031
Reporte por fechas	TC032
Reporte por cliente	TC033
Reporte por medio de pago	TC034
Reporte por número de factura	TC035
Reporte por placa	TC036
Reporte-resumen	TC037

- **Diseño de casos de prueba:**

Caso de Prueba	TC001
Descripción	Obtener el listado de los clientes que existen actualmente.
Precondiciones	En la base de datos existan clientes.
Pasos	Consumir el endpoint /clientes en el método HTTP GET

Caso de Prueba	TC002
Descripción	Obtener un cliente en específico
Precondiciones	En la base de datos deben existir clientes.
Pasos	Consumir el endpoint /clientes en el método HTTP GET enviando el documento del cliente

Caso de Prueba	TC003
Descripción	Registrar un nuevo cliente
Precondiciones	Pueden o no existir clientes en el sistema.
Pasos	Consumir el endpoint /clientes en el método HTTP POST la estructura del json { "tipo_doc": "CC", "documento": "string", "nombre": "string", "apellido": "string", "fec_nacimiento": "2024-11-16", "telefono": "strings", "email": "user@example.com" }

Caso de Prueba	TC004
Descripción	Actualizar un cliente

Precondiciones	Debe existir al menos un cliente
Pasos	Consumir el endpoint /clientes en el método HTTP PUT la estructura del json { "tipo_doc": "CC", "documento": "string", "nombre": "string", "apellido": "string", "fec_nacimiento": "2024-11-16", "telefono": "strings", "email": "user@example.com" }

Caso de Prueba	TC005
Descripción	Eliminar un cliente
Precondiciones	Debe existir al menos un cliente
Pasos	Consumir el endpoint /clientes en el método HTTP DELETE enviando el documento del cliente

Caso de Prueba	TC006
Descripción	Obtener lista de los usuarios
Precondiciones	Consumir el endpoint /usuarios en el método HTTP GET
Pasos	Consumir el endpoint /usuarios en el método HTTP GET

Caso de Prueba	TC007
Descripción	Obtener un usuario en específico
Precondiciones	En la base de datos deben existir usuarios.
Pasos	Consumir el endpoint /usuarios en el método HTTP GET enviando el "username"

Caso de Prueba	TC008
Descripción	Registrar un usuario nuevo
Precondiciones	No es necesario que se encuentren los usuarios creados
Pasos	Consumir el endpoint /usuarios en el método HTTP POST la estructura del json { "usuario": "string", "nombre": "string", "apellido": "string", "clave": "string", "rol": "string" }

Caso de Prueba	TC009
Descripción	Realizar el login de un usuario
Precondiciones	Por lo menos debe existir un usuario

Pasos	Consumir el endpoint /usuarios/login en el método HTTP POST enviando el "username" y la clave, estos datos son obligatorios.
-------	------------------------------------------------------------------------------------------------------------------------------

Caso de Prueba	TC010
Descripción	Actualizar un usuario
Precondiciones	Por lo menos debe existir un usuario
Pasos	Consumir el endpoint /usuarios/{usuario} en el método HTTP PUT la estructura del json { "usuario": "string", "nombre": "string", "apellido": "string", "clave": "string", "rol": "string" }, aparte recibe el suaurio por url

Caso de Prueba	TC011
Descripción	Eliminar un usuario
Precondiciones	Por lo menos debe existir un usuario
Pasos	Consumir el endpoint /usuarios/{usuario} en el método HTTP DELETE, recibe solo el "username"

Caso de Prueba	TC012
Descripción	Obtener lista de los vehículos
Precondiciones	Consumir el endpoint /vehículos en el método HTTP GET
Pasos	Consumir el endpoint /vehículos en el método HTTP GET

Caso de Prueba	TC013
Descripción	Obtener un vehículo en específico
Precondiciones	En la base de datos deben existir vehículos.
Pasos	Consumir el endpoint /vehículos en el método HTTP GET enviando la placa del vehículo

Caso de Prueba	TC014
Descripción	Registrar un vehículo nuevo
Precondiciones	Pueden o no existir vehículos en el sistema.

Pasos	<p>Consumir el endpoint /vehiculos en el método HTTP POST la estructura del json {</p> <pre> "Placa": "string", "Documento cliente": "string", "Segmento": "Moto", "Marca": "string", "Linea": "string", "Modelo": "string", "Cilindrada": 1, "Tipo": "Sport" } </pre>
-------	------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Caso de Prueba	TC015
Descripción	Actualizar un vehículo
Precondiciones	Por lo menos debe existir un vehículo
Pasos	<p>Consumir el endpoint /vehiculos en el método HTTP PUT la estructura del json {</p> <pre> "Placa": "string", "Documento cliente": "string", "Segmento": "Moto", "Marca": "string", "Linea": "string", "Modelo": "string", "Cilindrada": 1, "Tipo": "Sport" } </pre>

Caso de Prueba	TC016
Descripción	Eliminar un vehículo
Precondiciones	Por lo menos debe existir un vehículo
Pasos	Consumir el endpoint /vehiculos en el método HTTP DELETE, recibe solo la placa del vehículo

Caso de Prueba	TC017
Descripción	Obtener configuración
Precondiciones	Configuración precargada en la base de datos.
Pasos	Consumir el endpoint /configuracion en el método HTTP GET, no recibe parámetros

Caso de Prueba	TC018
Descripción	Actualizar configuración
Precondiciones	Configuración precargada en la base de datos.

Pasos	<p>Consumir el endpoint /configuracion en el método HTTP PUT, Parámetros de entrada</p> <pre> { "empresa": { "nombre": "LavApp", "nit": "1234567890", "telefono": "6042223243", "direccion": "Cl 30 # 34-45", "logo": "data:image/png;base64,iVBORw0KGgoAAAANSU UhEUgAAAHcAAAAfCAYAAADdn1K9AAACXBIWXMAAAAsTAAALeEAm pwYAAAAAXNSR0IArs4c6QAAAAARnQU1BAACxjwv8YQUAAASeSURBV HgB7VrdcdswDIZ7fa87QdQJ6g3MTNBsUHWCPBPYncDpBLE3cCawM kGcCaxM4HQCFoioC60AI0QwD7bz3fFskR/BH5AASGngvR8CwAgYD AaDCo4Q0GYa71DjHMXyCaClFwBHChzaxqfh4MDxCU4MQWmFgfoDD hwnp1zETyOvhAPHKSrXGXnDQzfNJ6XcHia5xUGb5s/wjgiT+R12o /EnTDWm04xI10ydiSDuAf1Lh1/izxnDr5G/60RZTXKLC0y/NYLS3 wW2X4fTCLXbzgGN/1aKxpFfAN/PJ6zzN+KU8DLuum2vKyx7tExKx bTyadz75lgS15XqrYS2NgL/pg/Xy3CJsUoY09gwbYXyJw+Uxm0dy w/1V17GrCsq3Kx2tT3xzSsfy1wtkxbI0Xm2DhphMLlP8kxivhp 5cV22LjOwpW+knc0qexgvdQrm8maV+4IG0Y4hj6vmH6dQnx8XdmI dwRK2HjbViBTbnbHjKvSFbugGom5JOfmW0qQMbzDkG/8aTwvneep YcNvYKcwG39+K1QTgVwQX8URp4zyh/2kd1BmcPcyr1g8mpM31Bpv zCd4/+pUDf2vUuj/JHAm8cPXo+Sn9sKAc6TwBnDfrg07TpMa4Vnj cqp9Mg8xc0c8uBFsI4m1n2jTkdM6kbME1S7XnZNG8jjuQCepkv rxOHQk3zPxSU/K9HhtcdmRq3Fm2nUvmFNNdmzDrAdM/TBT0UJBEP o38y9QiC3jTShPULhYnVK+YPCv3TuDtY5p3rE8Y01zgFMDDoi0zA nn3FtnPuWESyH86eBuWggwykWuQTeUcXvenELhrvxutaqaT2qvAj kcm70Ei0840C0BCLZRTnwsmp6/P9c2Rhnan7djIes78Mv52zpYj RjaxQX5xE2U7hVuCf2w7cn/AvtBXBDZ1Bt2wCTRiQpk07QDxTS70 BanXJYPedDXNH918s4kMo73EXRI758LqULOnXsp5JPZcNj5ryFaX oAdXNTcXudxmMcPwT8XkA99omYuAnYCVzPHLaQXGSNRptdvpPQpDe 15RXr427EbL5ujc6xcaXXBR8sznxT3ThgSKjIuIp0W2q768wJ0q3 MtUQLWBNERQd+Mw6kwBuuneAZ1mrFOBzbRWTN6Fwk1Bdw8joS5F/ mcGE0qgca+Qv070BcDoqqBxSaSTC170zxJuc0bLtzC/CgoiJL9dY iBFzV3M4wevm+Q1KuiPUNZG2JIySkxi3Q4KSLuFGuQjmCSzTHAqe kOU0+dqu9eFRIqtweZjLHJb9I2S10pZ6rbKgQ2Vkf961Xd22XS7 RXkvMSowKYIej9qVm7irrKFVY6aZGYhcJAWAJnGM0iDjlnrBGfav qM1QrtyJNBcleliYXr0RaElNDdQNVNcQRM1L6E/UnXm8UPCJFdgw 61SVkIaNNF00rgGvg90cw0cgtL0gR8D5Z3HHygm4J0QJrg9mNfGI OTgQGgPuoRiCxHNx4N2ExX8PfthAtYbRDxycfSWjK5468Tt1gf2g XIU2etNknYUgp44xa8fTwYfyj1ifCj3iPGh3CPGu363fCKYcVn7n g5qMHZQYMF/TDMfb42saGoAAAAASUVORK5CYII=" }, "tema": { "primario": "100 80% 30%", "secundario": "250 80% 70%" } } </pre>
-------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Caso de Prueba	TC019
Descripción	Reestablecer configuración
Precondiciones	Configuración precargada en la base de datos.
Pasos	Consumir el endpoint /configuracion/reset en el método HTTP PUT, no recibe parámetros de entrada

Caso de Prueba	TC020
Descripción	Obtener las promociones
Precondiciones	Promociones precargadas en la base de datos.
Pasos	Consumir el endpoint /Promociones en el método HTTP GET, no recibe parámetros de entrada o se recibe el id de la promoción a consultar

Caso de Prueba	TC021
Descripción	Crear una promoción
Precondiciones	Ninguna.
Pasos	Consumir el endpoint /Promociones en el método HTTP POST, json { " id_promocion": 0, " descripcion": "string", " fecha_inicio": "2024-11-27", " fecha_fin": "2024-11-27", " porcentaje": 0, " estado": true }

Caso de Prueba	TC022
Descripción	Actualizar una promoción
Precondiciones	Promociones en el sistema
Pasos	Consumir el endpoint /Promociones en el método HTTP PUT, json { " id_promocion": 0, " descripcion": "string", " fecha_inicio": "2024-11-27", " fecha_fin": "2024-11-27", " porcentaje": 0, " estado": true }

Caso de Prueba	TC023
Descripción	Eliminar una promoción
Precondiciones	Promociones en el sistema
Pasos	Consumir el endpoint /Promociones?id_promocion=1 en el método HTTP DELETE. Se envía el id de la promoción a eliminar

Caso de Prueba	TC024
Descripción	Consultar Factura o Facturas
Precondiciones	Facturas registradas
Pasos	Consumir el endpoint /facturas en el método HTTP GET. Se pasa vacío o con el id si se quiere consultar un solo registro

Caso de Prueba	TC025
Descripción	Crear factura
Precondiciones	Ninguna
Pasos	Consumir el endpoint /facturas en el método HTTP POST. <pre> Json { "factura": 0, "fecha": "2024-11-27T00:46:34.802Z", "placa": "string", "cliente": "string", "medio_pago": "string", "descuento": 0, "servicios": [{ "servicio": "9999", "cantidad": 1, "descripcion": "string", "valor": 0 }] }</pre>

Caso de Prueba	TC026
Descripción	Actualizar factura
Precondiciones	Facturas registradas
Pasos	Consumir el endpoint /facturas/10002 en el método HTTP PUT. Json { <pre> "numero_factura": 1007, "fecha": "2024-11-27", "placa": "string", "id_cliente": "string", "medio_pago": "string", "descuento": 0,</pre>

	<pre> "servicios": [{ "servicio": 9999, "cantidad": 1, "descripcion": "string", "valor": 0 }] </pre>
--	----------------------------------------------------------------------------------------------------------------------------

Caso de Prueba	TC027
Descripción	Eliminar factura
Precondiciones	Facturas registradas
Pasos	Consumir el endpoint /facturas/10002 en el método HTTP DELETE. Solo recibe el número de factura

Caso de Prueba	TC028
Descripción	Obtener Servicios
Precondiciones	Servicios registrados
Pasos	Consumir el endpoint /servicios/servicios?tipo_servicio=Film%20modificado%20 en el método HTTP GET. Recibe un tipo de servicio en string

Caso de Prueba	TC029
Descripción	Crear servicio
Precondiciones	Ninguna
Pasos	Consumir el endpoint /servicios/servicios en el método HTTP POST. Json { " id_servicio": 0, " nombre": "string", " tipo_servicio": "General", " valores": [{ " categoria": "string", " grupos": [{ " id": 0, " precio": 0

	<pre> }] }] }</pre>
--	-------------------------------------------

Caso de Prueba	TC030
Descripción	Actualizar un servicio
Precondiciones	Servicios registrados
Pasos	<p>Consumir el endpoint /servicios/servicios/1010?tipo_servicio=General en el método HTTP PUT. Recibe dos datos , id del servicio y el tipo de servicio y el json {</p> <pre> "id_servicio": 0, "nombre": "Plus", "tipo_servicio": "General", "valores": [{ "categoria": "string", "grupos": [{ "id": 0, "precio": 1200 }] }] }</pre>

Caso de Prueba	TC031
Descripción	Eliminar servicio
Precondiciones	Servicios registrados
Pasos	<p>Consumir el endpoint /servicios/servicios/1010?tipo_servicio=General en el método HTTP DELETE. Recibe id del servicio y el tipo</p>

Caso de Prueba	TC032
Descripción	Reporte por fechas
Precondiciones	Información en la BD

Pasos	Consumir el endpoint /reportes en el método HTTP GET. Recibe fecha de inicio y fecha fin
-------	---------------------------------------------------------------------------------------------

Caso de Prueba	TC033
Descripción	Reporte por cliente
Precondiciones	Información en la BD
Pasos	Consumir el endpoint /reportes/cliente en el método HTTP GET. Recibe número documento del cliente

Caso de Prueba	TC034
Descripción	Reporte por medio de pago
Precondiciones	Información en la BD
Pasos	Consumir el endpoint /reportes/medio_pago en el método HTTP GET. Recibe medio de pago

Caso de Prueba	TC035
Descripción	Reporte por número de factura
Precondiciones	Información en la BD
Pasos	Consumir el endpoint /reportes/numero_factura en el método HTTP GET. Recibe número de factura

Caso de Prueba	TC036
Descripción	Reporte por placa
Precondiciones	Información en la BD
Pasos	Consumir el endpoint /reportes/placa en el método HTTP GET. Recibe la placa

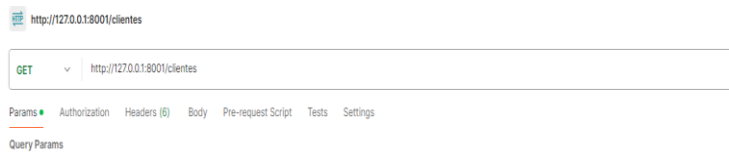
Caso de Prueba	TC037
Descripción	Reporte-resumen
Precondiciones	Información en la BD
Pasos	Consumir el endpoint /reportes/resumen en el método HTTP GET. Recibe fecha de inicio y fecha fin

4. Entorno de Prueba

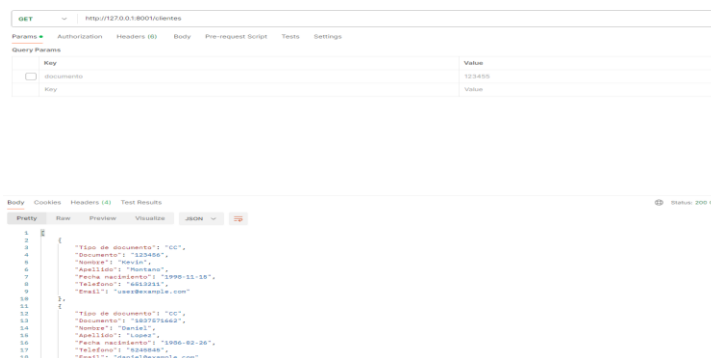
- **Herramientas:** La herramienta utilizada es Postman y el **swagger** de la aplicación.
- **Datos de prueba:** Los datos de prueba son aleatorios para validar cada uno de los endpoints.

5. Ejecución y resultado de Pruebas

- **TC001**
Se consume /clientes en el método GET sin enviar parámetros para obtener el listado de los clientes.



Respuesta exitosa al traer el listado de los clientes.



- **TC002**
Se consume /clientes?documento=123456 en el método GET para traer la información de un solo cliente.

http://127.0.0.1:8001/clientes?documento=123456

GET http://127.0.0.1:8001/clientes?documento=123456

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
<input checked="" type="checkbox"/> documento	123456
Key	Value

Respuesta exitosa al traer los datos de un solo cliente.

GET http://127.0.0.1:8001/clientes?documento=123456

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
<input checked="" type="checkbox"/> documento	123456
Key	Value

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "tipo_documento": "CC",
3   "documento": "123456",
4   "nombre": "Pedro",
5   "apellido": "Montana",
6   "fecha_nacimiento": "1990-11-16",
7   "telefono": "8613331",
8   "email": "user@example.com"
9 }
```

Respuesta cuando un cliente no existe.

GET http://127.0.0.1:8001/clientes?documento=123455

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
<input checked="" type="checkbox"/> documento	123455
Key	Value

Body Cookies Headers (4) Test Results

Pretty Raw Preview Visualize JSON

```
1 {
2   "detail": "Cliente no encontrado"
3 }
```

- **TC003**

Se consume /clientes en el método POST para crear un nuevo cliente.

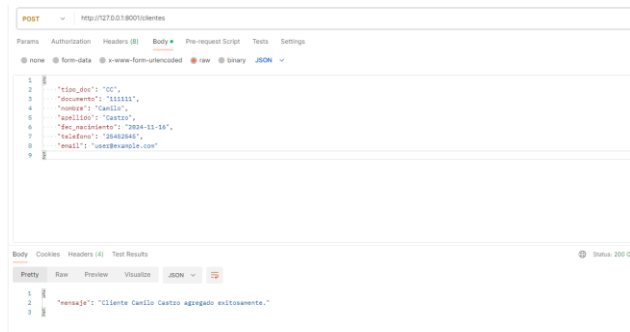
POST http://127.0.0.1:8001/clientes

Params Authorization Headers (8) Body Pre-request Script Tests Settings

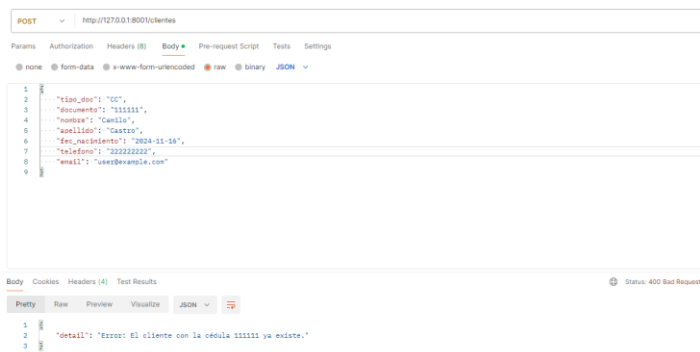
none form-data x-www-form-urlencoded raw binary JSON

```
1 {
2   "tipo_doc": "CC",
3   "documento": "111111",
4   "nombre": "Camilo",
5   "apellido": "Castro",
6   "fec_nacimiento": "2024-11-16",
7   "telefono": "25452545",
8   "email": "user@example.com"
9 }
```

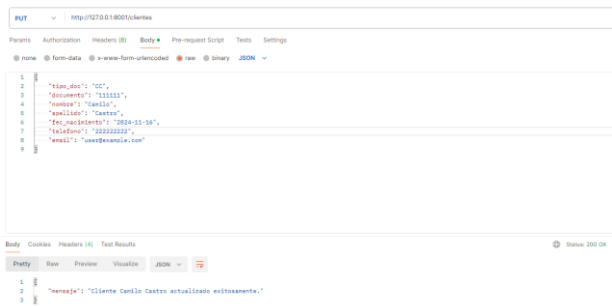
Respuesta exitosa a la creación del cliente nuevo.



Respuesta cuando se trata de crear un cliente que ya existe.



- **TC004**
Se consume /clientes en el método PUT para la actualización de un registro y la respuesta exitosa.



Se consulta /clientes en el método GET para la validación del nuevo cliente.

GET		http://127.0.0.1:8001/clientes/documento=111111
Params • Authorization Headers (0) Body Pre-request Script Tests Settings		
Query Params		
Key		Value
<input checked="" type="checkbox"/> documento		111111
Key		Value

Body		Cookies	Headers (4)	Test Results	Status: 200 OK
Pretty Raw Preview Visualize JSON					
<pre>1 { 2 "files de documentos": "01", 3 "documento": "111111", 4 "nombre": "Carlos", 5 "apellido": "Garcia", 6 "fecha nacimiento": "2020-11-08", 7 "telefono": "333333333", 8 "email": "user@example.com" 9 }</pre>					

- **TC005**

Se consume /clientes en el método DELETE para eliminar un cliente, respuesta exitosa.

DELETE		http://127.0.0.1:8001/clientes/documento=111111
Params • Authorization Headers (0) Body Pre-request Script Tests Settings		
Query Params		
Key		Value
<input checked="" type="checkbox"/> documento		111111
Key		Value

Body		Cookies	Headers (4)	Test Results	Status: 200 OK
Pretty Raw Preview Visualize JSON					
<pre>1 { 2 "mensaje": "Cliente Carlos Garcia eliminado exitosamente." 3 }</pre>					

Consultamos el cliente nuevamente por /clientes en el método GET.

GET		http://127.0.0.1:8001/clientes/documento=111111
Params • Authorization Headers (0) Body Pre-request Script Tests Settings		
Query Params		
Key		Value
<input checked="" type="checkbox"/> documento		111111
Key		Value

Body		Cookies	Headers (4)	Test Results	Status: 404 Not Found
Pretty Raw Preview Visualize JSON					
<pre>1 { 2 "detail": "Cliente no encontrado" 3 }</pre>					

- **TC006**

Se consulta /usuarios en el método GET sin parámetros para obtener el listado de los usuarios.

GET	http://127.0.0.1:8001/usuarios
Params	Authorization Headers (0) Body Pre-request Script Tests Settings
Query Params	
Key	Value
Key	Value

```

1  {
2    "users": [
3      {
4        "username": "PRUEBA1",
5        "password": "123456",
6        "email": "prueba1@prueba.com",
7        "phone": "987654321",
8        "role": "user"
9      },
10     {
11       "username": "PRUEBA2",
12       "password": "123456",
13       "email": "prueba2@prueba.com",
14       "phone": "987654321",
15       "role": "user"
16     }
17   ]
18 }

```

- **TC007**

Se consulta /usuarios?usuario=PRUEBA1 en el método GET para obtener un solo usuario.

GET	http://127.0.0.1:8001/usuarios?usuario=PRUEBA1
Params	Authorization Headers (0) Body Pre-request Script Tests Settings
Query Params	
Key	Value
<input checked="" type="checkbox"/> usuario	PRUEBA1
Key	Value

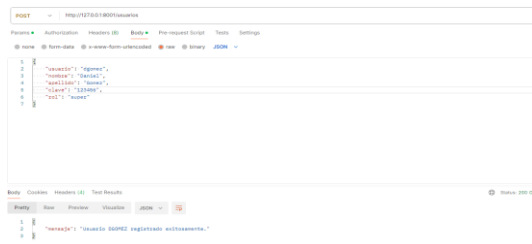
```

1  {
2    "user": {
3      "username": "PRUEBA1",
4      "password": "123456",
5      "email": "prueba1@prueba.com",
6      "phone": "987654321",
7      "role": "user"
8    }
9  }

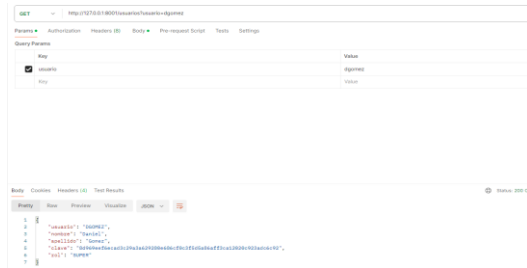
```

- **TC008**

Se consume /usuarios en el método POST para crear un usuario nuevo, respuesta exitosa.

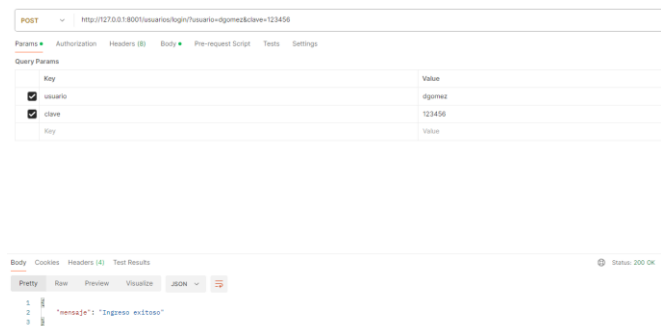


Se consulta /usuarios?usuario=dgomez en el método GET para validar la creación.



- **TC009**

Se consume /usuarios/login?usuario=dgomez&clave=123456 para iniciar sesión, respuesta exitosa.



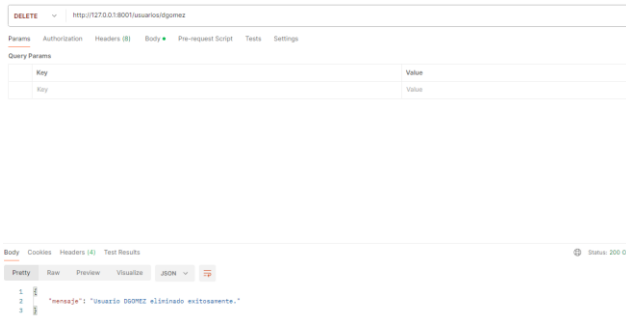
Si se envía un usuario o clave errados, el sistema devuelve error y no permite el ingreso.

POST	http://127.0.0.1:8001/usuarios/login/?usuario=dpomez&clave=123456
Params	Authorization Headers (7) Body Pre-request Script Tests Settings
Query Params	
Key	Value
<input checked="" type="checkbox"/> usuario	dpomez
<input checked="" type="checkbox"/> clave	123456
Key	Value

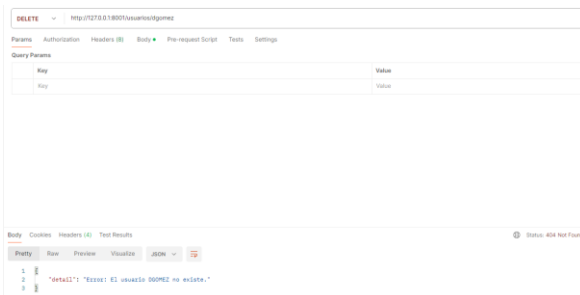
Body	Cookies	Headers (4)	Test Results
Pretty	Raw	Preview	Visualize
JSON			
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			
17			
18			
19			
20			
21			
22			
23			
24			
25			
26			
27			
28			
29			
30			
31			
32			
33			
34			
35			
36			
37			
38			
39			
40			
41			
42			
43			
44			
45			
46			
47			
48			
49			
50			
51			
52			
53			
54			
55			
56			
57			
58			
59			
60			
61			
62			
63			
64			
65			
66			
67			
68			
69			
70			
71			
72			
73			
74			
75			
76			
77			
78			
79			
80			
81			
82			
83			
84			
85			
86			
87			
88			
89			
90			
91			
92			
93			
94			
95			
96			
97			
98			
99			
100			
101			
102			
103			
104			
105			
106			
107			
108			
109			
110			
111			
112			
113			
114			
115			
116			
117			
118			
119			
120			
121			
122			
123			
124			
125			
126			
127			
128			
129			
130			
131			
132			
133			
134			
135			
136			
137			
138			
139			
140			
141			
142			
143			
144			
145			
146			
147			
148			
149			
150			
151			
152			
153			
154			
155			
156			
157			
158			
159			
160			
161			
162			
163			
164			
165			
166			
167			
168			
169			
170			
171			
172			
173			
174			
175			
176			
177			
178			
179			
180			
181			
182			
183			
184			
185			
186			
187			
188			
189			
190			
191			
192			
193			
194			
195			
196			
197			
198			
199			
200			
201			
202			
203			
204			
205			
206			
207			
208			
209			
210			
211			
212			
213			
214			
215			
216			
217			
218			
219			
220			
221			
222			
223			
224			
225			
226			
227			
228			
229			
230			
231			
232			
233			
234			
235			
236			
237			
238			
239			
240			
241			
242			
243			
244			
245			
246			
247			
248			
249			
250			
251			
252			
253			
254			
255			
256			
257			
258			
259			
260			
261			
262			
263			
264			
265			
266			
267			
268			
269			
270			
271			
272			
273			
274			
275			
276			
277			
278			
279			
280			
281			
282			
283			
284			
285			
286			
287			
288			
289			
290			
291			
292			
293			
294			
295			
296			
297			
298			
299			
300			
301			
302			
303			
304			
305			
306			
307			
308			
309			
310			
311			
312			
313			
314			
315			
316			
317			
318			
319			
320			
321			
322			
323			
324			
325			
326			
327			
328			
329			
330			
331			
332			
333			
334			
335			
336			
337			
338			
339			
340			
341			
342			
343			
344			
345			
346			
347			
348			
349			
350			
351			
352			
353			
354			
355			
356			
357			
358			
359			
360			
361			
362			
363			
364			
365			
366			
367			
368			
369			
370			
371			
372			
373			
374			
375			
376			
377			
378			
379			
380			
381			
382			
383			
384			
385			
386			
387			
388			
389			
390			
391			
392			
393			
394			
395			
396			
397			
398			
399			
400			
401			
402			
403			
404			
405			
406			
407			
408			
409			
410			
411			
412			
413			
414			
415			
416			
417			
418			
419			
420			
421			
422			
423			
424			
425			
426			
427			
428			
429			
430			
431			
432			
433			
434			
435			
436			
437			
438			
439			
440			
441			
442			
443			
444			
445			
446			
447			
448			
449			
450			
451			
452			
453			
454			
455			
456			
457			
458			
459			
460			
461			
462			
463			
464			
465			
466			
467			
468			
469			
470			
471			
472			
473			
474			
475			
476			
477			
478			
479			
480			
481			
482			
483			
484			
485			
486			
487			
488			
489			
490			
491			
492			
493			
494			
495			
496			
497			
498			
499			
500			
501			
502			
503			
504			
505			

- **TC011**

Se consume /usuarios/dgomez en el método DELETE para eliminar un usuario, respuesta exitosa.

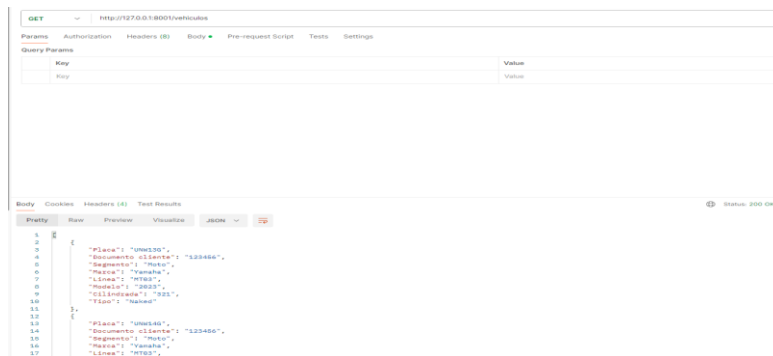


Intentamos eliminar de nuevo el cliente y el sistema nos indica que este ya no existe.



- **TC012**

Se consume /vehículos en el método GET para obtener el listado de los vehículos.



- **TC013**

Se consume /vehículos?placa=BQX80A&documento=123456 para obtener la información de un solo vehículo.

GET	http://127.0.0.1:8001/vehiculos?placa=UNW13G&documento_cliente=123456
Params	Authorization Headers (8) Body Pre-request Script Tests Settings
Query Params	
Key	Value
placa	UNW13G
documento_cliente	123456
Key	Value

```
Body Cookies Headers (4) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "Placa": "UNW13G",
3   "Documento_cliente": "123456",
4   "Segmento": "Nota",
5   "Marca": "Yamaha",
6   "Linea": "F188",
7   "Modelo": "2003",
8   "Cilindrada": "393",
9   "Tipo": "Halter"
10 }
```

- **TC014**

Se consume /vehículos en el método POST para crear un vehículo nuevo. Respuesta exitosa.

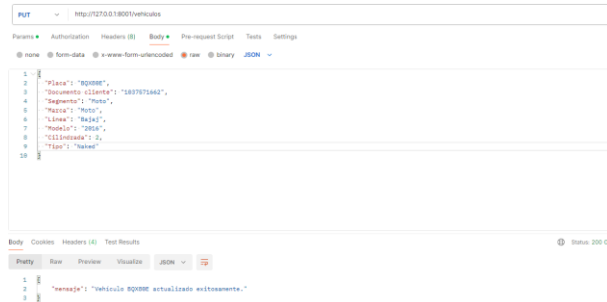
POST	http://127.0.0.1:8001/vehiculos
Params	Authorization Headers (8) Body Pre-request Script Tests Settings
Body	none form-data x-www-form-urlencoded raw binary JSON
Body	raw
1	{
2	"Placa": "BQX80E",
3	"Documento_cliente": "1837971662",
4	"Segmento": "Nota",
5	"Marca": "Nota",
6	"Linea": "Bajaj",
7	"Modelo": "2018",
8	"Cilindrada": "3",
9	"Tipo": "Bajaj"
10	}
Body Cookies Headers (4) Test Results	Status: 200 OK
Body	raw
1	{
2	"mensaje": "vehículo BQX80E agregado exitosamente."
3	}

Consultamos el vehículo creado por el método GET y pasando los parámetros de placa y documento.

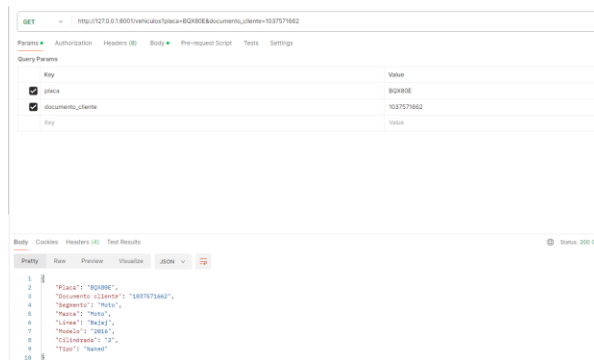
GET	http://127.0.0.1:8001/vehiculos?placa=BQX80E&documento_cliente=103757682
Params	Authorization Headers (8) Body Pre-request Script Tests Settings
Query Params	
Key	Value
placa	BQX80E
documento_cliente	103757682
Key	Value

```
Body Cookies Headers (4) Test Results
Pretty Raw Preview Visualize JSON
1 {
2   "Placa": "BQX80E",
3   "Documento_cliente": "1837971662",
4   "Segmento": "Nota",
5   "Marca": "Nota",
6   "Linea": "Bajaj",
7   "Modelo": "2018",
8   "Cilindrada": "3",
9   "Tipo": "Bajaj"
10 }
```

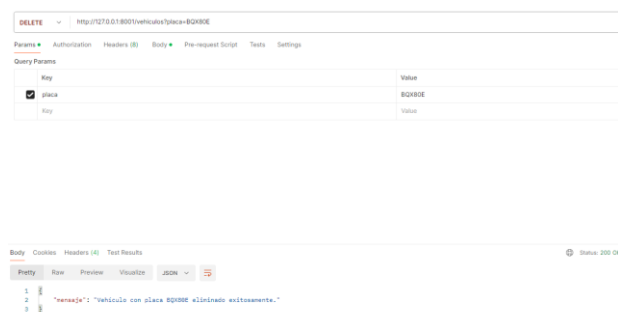
- TC015**
 Se consume /vehículos en el método PUT para actualizar la información de un vehículo. Respuesta exitosa.



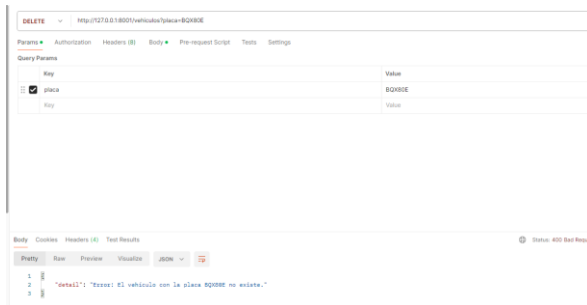
Se consulta por el método GET y se valida el vehículo creado.



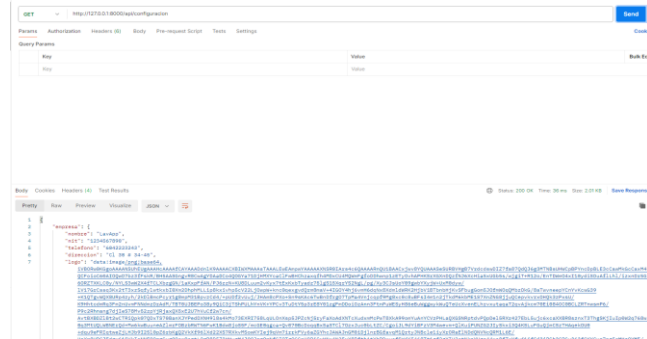
- TC016**
 Se consume /vehículos en el método DELETE para eliminar un registro. Respuesta exitosa.



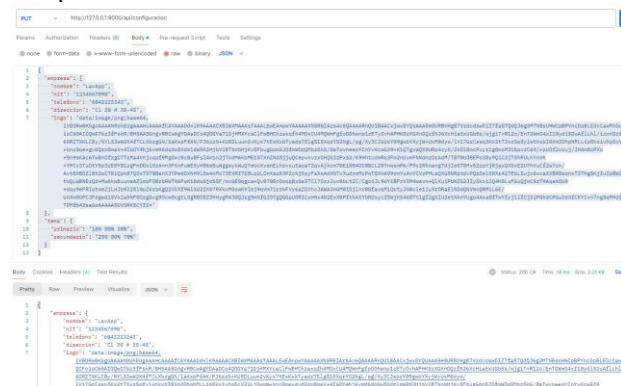
Si intentamos eliminar el mismo vehículo, el sistema indica que este no existe.



- TC017**
 Se consume /configuración en el método GET para consultar la configuración del sistema.
 Respuesta exitosa.

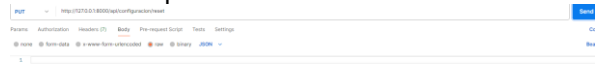


- TC018**
 Se consume /configuración en el método PUT para actualizar la configuración del sistema.
 Respuesta exitosa.



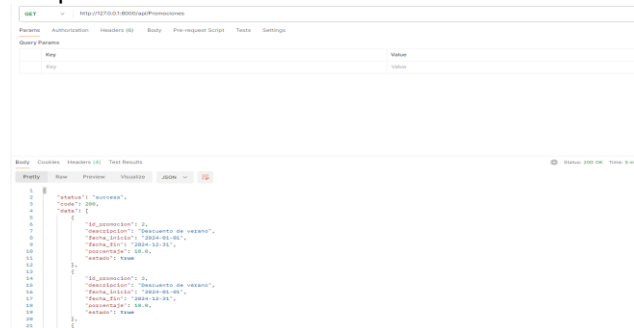
- **TC019**

Se consume / configuracion/reset en el método PUT para resetear la configuración del sistema. Respuesta exitosa.

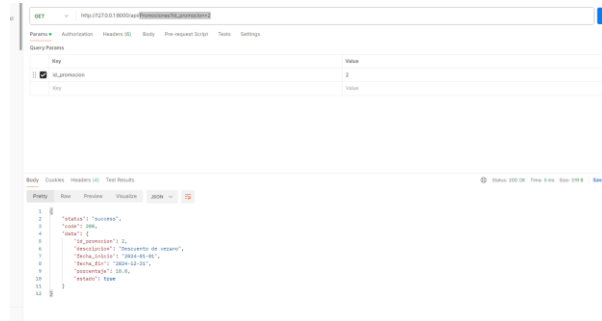


- **TC020**

Se consume / Promociones en el método GET para obtener las promociones.
Respuesta exitosa.

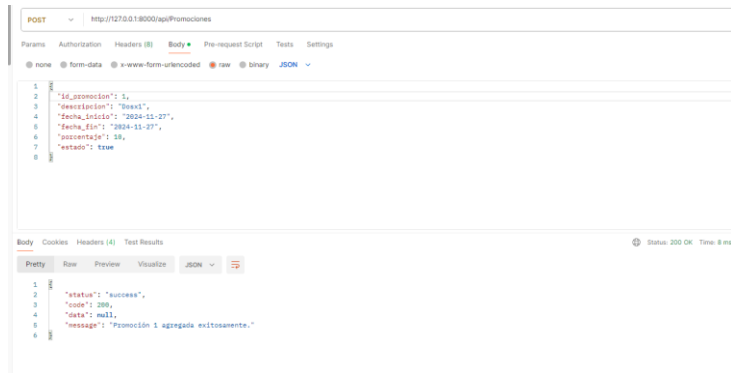


Quando se consulta solo una promoción. "Promociones?id_promocion=2"



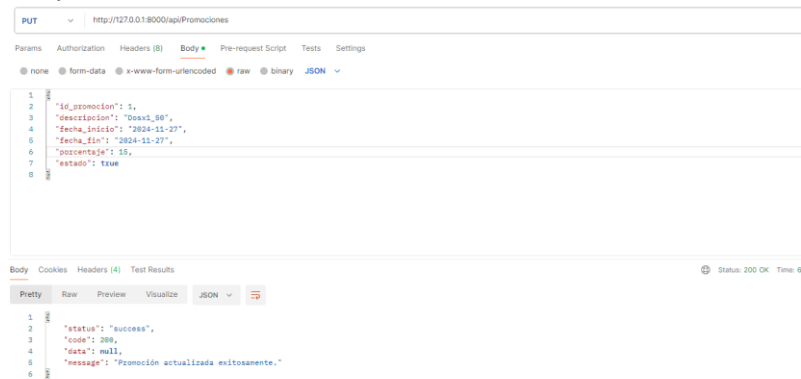
- **TC021**

Se consume / Promociones en el método POST para obtener las promociones.
Respuesta exitosa.



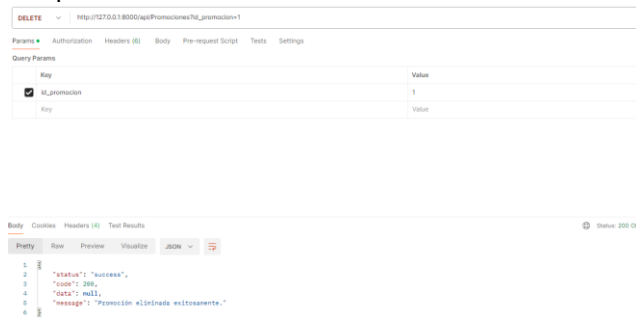
- **TC022**

Se consume / Promociones en el método PUT para obtener las promociones.
Respuesta exitosa.



- **TC023**

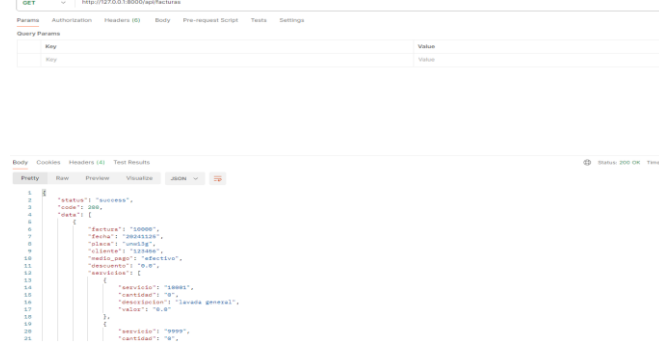
Se consume / Promociones?id_promocion=1 en el método DELETE para Eliminar una promoción.
Respuesta exitosa.



- **TC024**

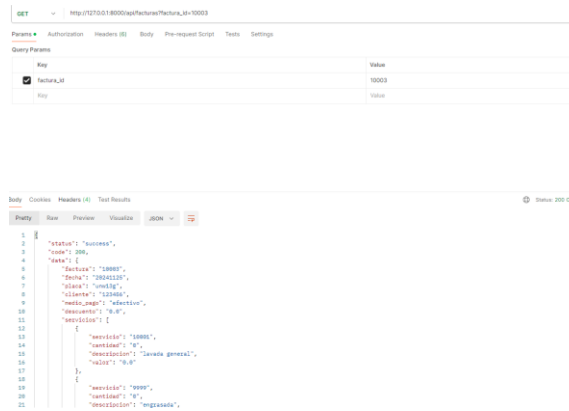
Se consume / facturas en el método GET para consultar el listado de facturas.

Respuesta exitosa.



Cuando solo se quiere consultar una factura se paso la URL de la siguiente forma.

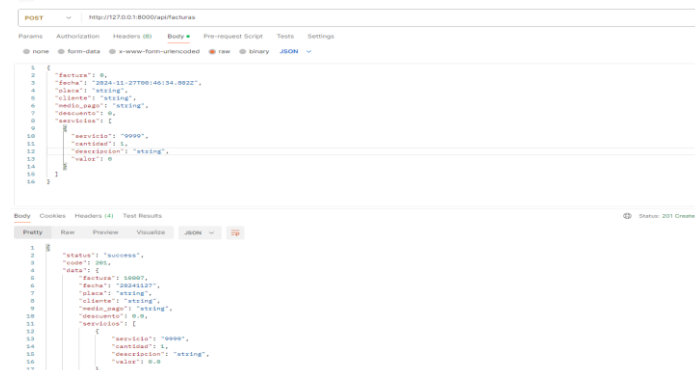
“facturas?factura_id=10003”



- **TC025**

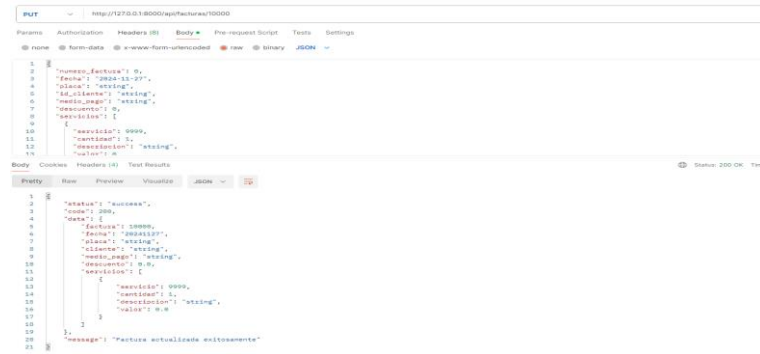
Se consume / facturas en el método POST para crear o registrar una factura nueva.

Respuesta exitosa.



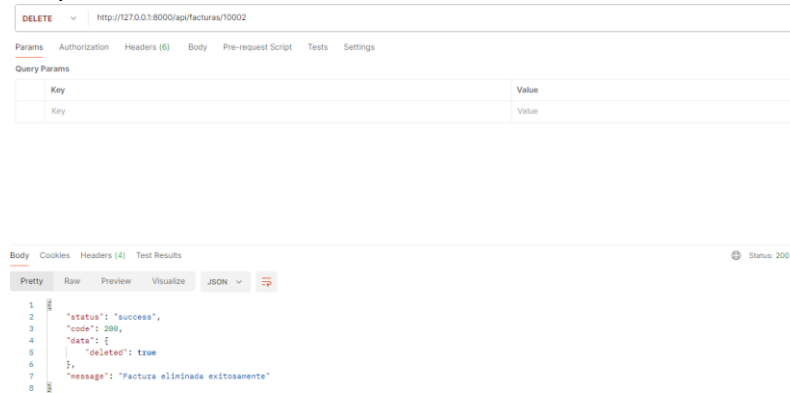
- **TC026**

Se consume / facturas/10002 en el método PUT para actualizar una factura.
Respuesta exitosa.



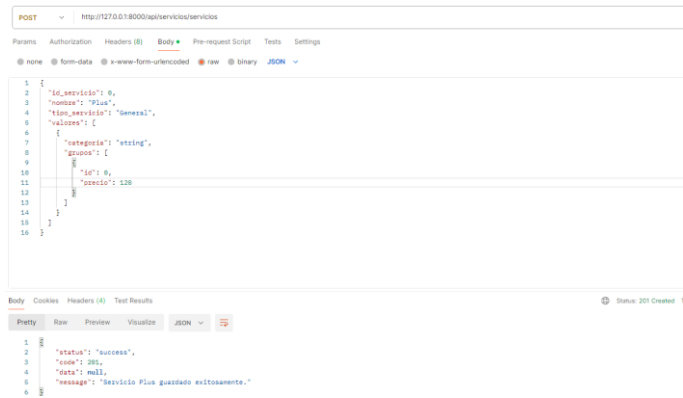
- **TC027**

Se consume / facturas/10002 en el método DELETE para eliminar una factura.
Respuesta exitosa.



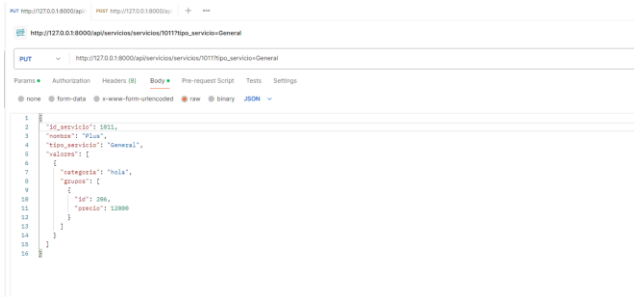
- **TC029**

Se consume /servicios/servicios en el método POST para crear un servicio.
Respuesta exitosa.



- **TC030**

Se consume /servicios/servicios en el método PUT para actualizar un servicio.
Respuesta exitosa.



```

{
  "status": "success",
  "code": 200,
  "data": {
    "id_servicio": 0,
    "nombre": "aaaaaa",
    "tipo_servicio": "General",
    "valores": [
      {
        "categoria": "hola",
        "grupos": [
          {
            "id": 0,
            "precio": 0
          }
        ]
      }
    ]
  },
  "message": "Servicio actualizado exitosamente"
}

```

- **TC031**

Se consume /servicios/servicios/1010?tipo_servicio=General en el método DELETE para eliminar un servicio.

Respuesta exitosa.

DELETE	http://127.0.0.1:8000/api/servicios/servicios/1010?tipo_servicio=General
Params	Authorization Headers (6) Body Pre-request Script Tests Settings
Query Params	
Key	Value
<input checked="" type="checkbox"/> tipo_servicio	General
Key	Value

Body	Cookies	Headers (4)	Test Results
Pretty	Raw	Preview	Visualize
1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16
17	18	19	20

- **TC032**

Consumir el endpoint /reportes en el método HTTP GET. Recibe fecha de inicio y fecha fin.

Respuesta exitosa.

GET http://127.0.0.1:8000/api/reportes?fecha_inicio=2024-11-01&fecha_fin=2024-11-29

GET

http://127.0.0.1:8000/api/reportes?fecha_inicio=2024-11-01&fecha_fin=2024-11-29

Params

Authorization

Headers (6)

Body

Pre-request Script

Tests

Settings

Query Params

Key

Value

☒

fecha_inicio

2024-11-01

☒

fecha_fin

2024-11-29

Body

Cookies

Headers (4)

Test Results

Pretty

Raw

Preview

Visualize

JSON

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

51

52

53

54

55

56

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

73

74

75

76

77

78

79

80

81

82

83

84

85

86

87

88

89

90

91

92

93

94

95

96

97

98

99

100

101

102

103

104

105

106

107

108

109

110

111

112

113

114

115

116

117

118

119

120

121

122

123

124

125

126

127

128

129

130

131

132

133

134

135

136

137

138

139

140

141

142

143

144

145

146

147

148

149

150

151

152

153

154

155

156

157

158

159

160

161

162

163

164

165

166

167

168

169

170

171

172

173

174

175

176

177

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

200

201

202

203

204

205

206

207

208

209

210

211

212

213

214

215

216

217

218

219

220

221

222

223

224

225

226

227

228

229

230

231

232

233

234

235

236

237

238

239

240

241

242

243

244

245

246

247

248

249

250

251

252

253

254

255

256

257

258

259

260

261

262

263

264

265

266

267

268

269

270

271

272

273

274

275

276

277

278

279

280

281

282

283

284

285

286

287

288

289

290

291

292

293

294

295

296

297

298

299

300

301

302

303

304

305

306

307

308

309

310

311

312

313

314

315

316

317

318

319

320

321

322

323

324

325

326

327

328

329

330

331

332

333

334

335

336

337

338

339

340

341

342

343

344

345

346

347

348

349

350

351

352

353

354

355

356

357

358

359

360

361

362

363

364

365

366

367

368

369

370

371

372

373

374

375

376

377

378

379

380

381

382

383

384

385

386

387

388

389

390

391

392

393

394

395

396

397

398

399

400

401

402

403

404

405

406

407

408

409

410

411

412

413

414

415

416

417

418

419

420

421

422

423

424

425

426

427

428

429

430

431

432

433

434

435

436

437

438

439

440

441

442

443

444

445

446

447

448

449

450

451

452

453

454

455

456

457

458

459

460

461

462

463

464

465

466

467

468

469

470

471

472

473

474

475

476

477

478

479

480

481

482

483

484

485

486

487

488

489

490

491

492

493

494

495

496

497

498

499

500

501

502

503

504

505

506

507

508

509

510

511

512

513

514

515

516

517

518

519

520

521

522

523

524

525

526

527

528

529

530

531

532

533

534

535

536

537

538

539

540

541

542

543

544

545

546

547

548

549

550

551

552

553

554

555

556

557

558

559

560

561

562

563

564

565

566

567

568

569

570

571

572

573

574

575

576

577

578

579

580

581

582

583

584

585

586

587

588

589

590

591

592

593

594

595

596

597

598

599

600

601

602

603

604

605

606

607

608

609

610

611

612

613

614

615

616

617

618

619

620

621

622

623

624

625

626

627

628

629

630

631

632

633

634

635

636

637

638

639

640

641

642

643

644

645

646

647

648

649

650

651

652

653

654

655

656

657

658

659

660

661

662

663

664

665

666

667

668

669

670

671

672

673

674

675

676

677

678

679

680

681

682

683

684

685

686

687

688

689

690

691

692

693

694

695

696

697

698

699

700

701

702

703

704

705

706

707

708

709

710

711

712

713

714

715

716

717

718

719

720

721

722

723

724

725

726

727

728

729

730

731

732

733

734

735

736

737

738

739

740

741

742

743

744

745

746

747

748

749

750

751

752

753

754

755

756

757

758

759

760

761

762

763

764

765

766

767

768

769

770

771

772

773

774

775

776

777

778

779

780

781

782

783

784

785

786

787

788

789

790

791

792

793

794

795

796

797

798

799

800

801

802

803

804

805

806

807

808

809

810

811

812

813

814

815

816

817

818

819

820

821

822

823

824

825

826

827

828

829

830

831

832

833

834

835

836

837

838

839

840

841

842

843

844

845

846

847

848

849

850

851

852

853

854

855

856

857

858

859

860

861

862

863

864

865

866

867

868

869

870

871

872

873

874

875

876

877

878

879

880

881

882

883

884

885

886

887

888

889

890

891

892

893

894

895

896

897

898

899

900

901

902

903

904

905

906

907

908

909

910

911

912

913

914

915

916

917

918

919

920

921

922

923

924

925

926

927

928

929

930

931

932

933

934

935

936

937

938

939

940

941

942

943

944

945

946

947

948

949

950

951

952

953

954

955

956

957

958

959

960

961

962

963

964

965

966

967

968

969

970

971

972

973

974

975

976

977

978

979

980

981

982

983

984

985

986

987

988

989

990

991

992

993

994

995

996

997

998

999

1000

1001

1002

1003

1004

1005

1006

1007

1008

1009

1010

1011

1012

1013

1014

1015

1016

1017

1018

1019

1020

1021

1022

1023

1024

1025

1026

1027

1028

1029

1030

1031

1032

1033

1034

1035

1036

1037

1038

1039

1040

1041

1042

1043

1044

1045

1046

1047

1048

1049

1050

1051

1052

1053

1054

1055

1056

1057

1058

1059

1060

1061

1062

1063

1064

1065

1066

1067

1068

1069

1070

1071

1072

1073

1074

1075

1076

1077

1078

1079

1080

1081

1082

1083

1084

1085

1086

1087

1088

1089

1090

1091

1092

1093

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

1143

1144

1145

1146

1147

1148

1149

1150

1151

1152

1153

1154

1155

1156

1157

1158

1159

1160

1161

1162

1163

1164

1165

1166

1167

1168

1169

1170

1171

1172

1173

1174

1175

1176

1177

1178

1179

1180

1181

1182

1183

1184

1185

1186

1187

1188

1189

1190

1191

1192

1193

1194

1195

1196

1197

1198

1199

1200

1201

1202

1203

1204

1205

1206

1207

1208

1209

1210

1211

1212

1213

1214

1215

1216

1217

1218

1219

1220

1221

1222

1223

1224

1225

1226

1227

1228

1229

1230

1231

1232

1233

1234

1235

1236

1237

1238

1239

1240

1241

1242

1243

1244

1245

1246

1247

1248

1249

1250

1251

1252

1253

1254

1255

1256

1257

1258

1259

1260

1261

1262

1263

1264

1265

1266

1267

1268

1269

1270

1271

1272

1273

1274

1275

1276

1277

1278

1279

1280

1281

1282

1283

1284

1285

1286

1287

1288

1289

1290

1291

1292

1293

1294

1295

1296

1297

1298

1299

1300

1301

1302

1303

1304

1305

1306

1307

1308

1309

1310

1311

1312

1313

1314

1315

1316

1317

1318

1319

1320

1321

1322

1323

1324

1325

1326

1327

1328

1329

1330

1331

1332

1333

1334

1335

1336

1337

1338

1339

1340

1341

1342

1343

1344

1345

1346

1347

1348

1349

1350

1351

1352

1353

1354

1355

1356

1357

1358

1359

1360

1361

1362

1363

1364

1365

1366

1367

1368

1369

1370

1371

1372

1373

1374

1375

1376

1377

1378

1379

1380

1381

1382

1383

1384

1385

1386

1387

1388

1389

1390

1391

1392

1393

1394

1395

1396

1397

1398

1399

1400

1401

1402

1403

1404

1405

1406

1407

1408

1409

1410

1411

1412

1413

1414

1415

1416

1417

1418

1419

1420

1421

1422

1423

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439

1440

1441

1442

1443

1444

1445

1446

1447

1448

1449

1450

1451

1452

1453

1454

1455

1456

1457

1458

1459

1460

1

- **TC033**

Consumir el endpoint /reportes/cliente en el método HTTP GET. Recibe número documento del cliente.

Respuesta exitosa.

GET http://127.0.0.1:8000/api/reportes/cliente?cedula_cliente=1234546

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
cedula_cliente	1234546

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 30 ms

```
1 {
2   "status": "success",
3   "code": 200,
4   "data": {
5     "factura": "10000",
6     "fecha": "2024-11-20T23:48:52.416000+00:00",
7     "place": "DEF456",
8     "categoria": "General",
9     "cliente": "1234546",
10    "medio_pago": "TD",
11    "descuento": "0.0",
12    "vlt_descuento": "0.0",
13    "subtotal": "0.0",
14    "total": "0.0",
15    "servicios": [
16      {
17        "servicio": "9999",
18        "cantidad": "1"
19      }
20    ]
21  }
22 }
```

- **TC034**

Consumir el endpoint /reportes/medio_pago en el método HTTP GET. Recibe medio de pago.

Respuesta exitosa.

GET http://127.0.0.1:8000/api/reportes/medio_pago?medio_pago=TD

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
medio_pago	TD

Body Cookies Headers (4) Test Results

Status: 200 OK Time: 9 ms

```
1 {
2   "status": "success",
3   "code": 200,
4   "data": {
5     "factura": "10000",
6     "fecha": "2024-11-20T23:48:52.416000+00:00",
7     "place": "DEF456",
8     "categoria": "General",
9     "cliente": "1234546",
10    "medio_pago": "TD",
11    "descuento": "0.0",
12    "vlt_descuento": "0.0",
13    "subtotal": "0.0",
14    "total": "0.0",
15    "servicios": [
16      {
17        "servicio": "9999",
18        "cantidad": "1"
19      }
20    ]
21  }
22 }
```

- **TC035**

Consumir el endpoint /reportes/numero_factura en el método HTTP GET. Recibe número de factura.

Respuesta exitosa.

GET http://127.0.0.1:8000/api/reportes/numero_factura?numero_factura=10000

Params Authorization Headers (6) Body Pre-request Script Tests Settings

Query Params

Key	Value
numero_factura	10000

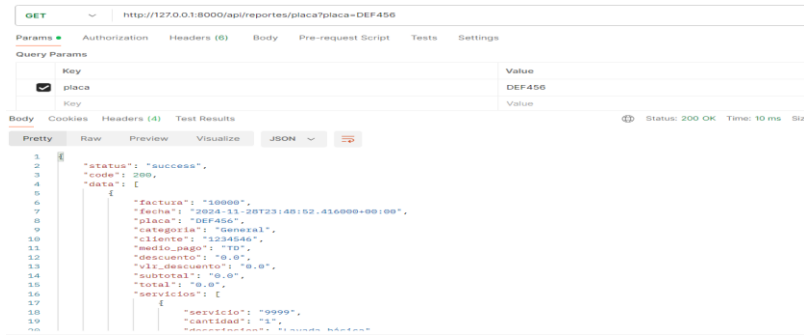
Body Cookies Headers (4) Test Results

Status: 200 OK Time: 9 ms

```
1 {
2   "status": "success",
3   "code": 200,
4   "data": {
5     "factura": "10000",
6     "fecha": "2024-11-20T23:48:52.416000+00:00",
7     "place": "DEF456",
8     "categoria": "General",
9     "cliente": "1234546",
10    "medio_pago": "TD",
11    "descuento": "0.0",
12    "vlt_descuento": "0.0",
13    "subtotal": "0.0",
14    "total": "0.0",
15    "servicios": [
16      {
17        "servicio": "9999",
18        "cantidad": "1"
19      }
20    ]
21  }
22 }
```

- **TC036**

Consumir el endpoint /reportes/placa en el método HTTP GET. Recibe la placa.
Respuesta exitosa.



- **TC037**

Consumir el endpoint /reportes/resumen en el método HTTP GET. Recibe fecha de inicio y fecha fin.
Respuesta exitosa.

6. Informes de Pruebas

Se adjunta informe en PDF "[Casos e informes de pruebas.pdf](#)".