**DOCUMENTACION API’s**

**Gestión de vehículos**

El proyecto cuenta con 8 APIs diseñadas bajo el framework FASTAPI para una rápida y sencilla gestión de la información, basadas en el estándar RESTAPI.

**ApiUsuario**

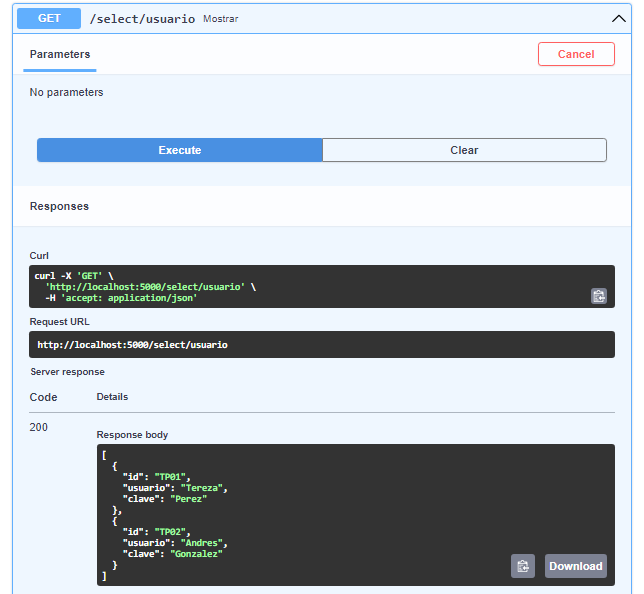
La API “ApiUsuario” tiene implementados los métodos básicos CRUD. El archivo usuario.json temporalmente es usado para obtener y guardar datos.

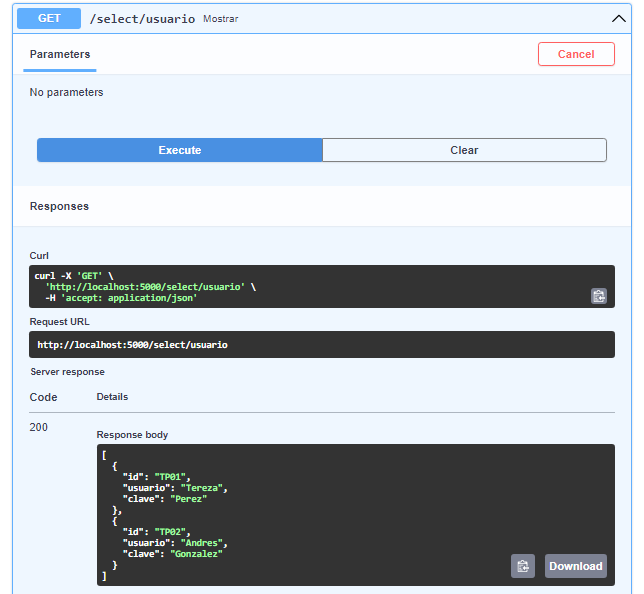
**#GET**--READ

No requiere ningún parámetro y devuelve una lista de objetos de tipo **usuario** declarado dentro de la API

ruta: http://localhost:puerto/select/usuario

puerto: remplazamos por el puerto usado al lanzar el servidor **uvicorn**





**#POST**--CREATE

Requiere en su “Request body” la información a guardar en formato JSON así:

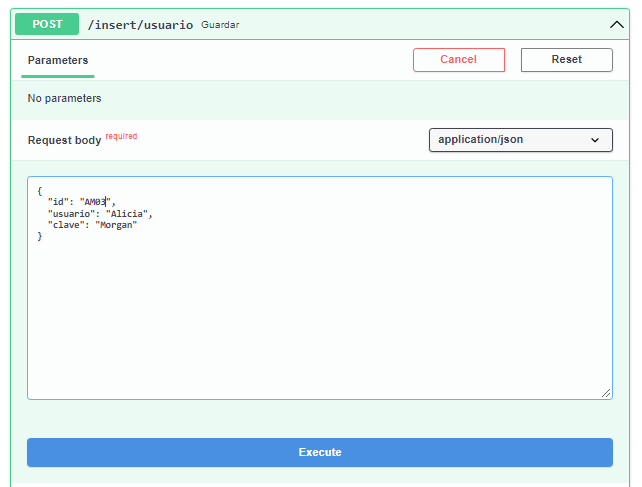
{

"id": "string",

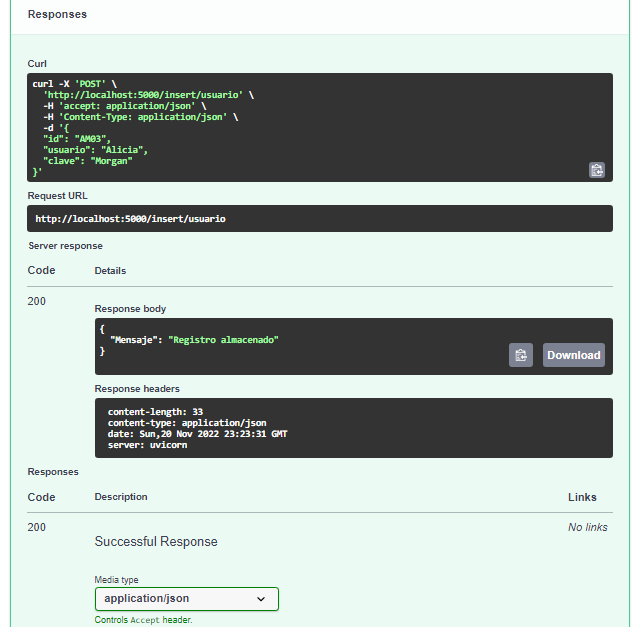
"usuario": "string",

"clave": "string"

}

ruta: http://localhost:puerto/insert/usuario

Y devuelve un mensaje confirmándonos el guardado de la información:



**#PUT**--UPDATE

Requiere en su “Request body” la información a actualizar en formato JSON así:

{

"id": "string",

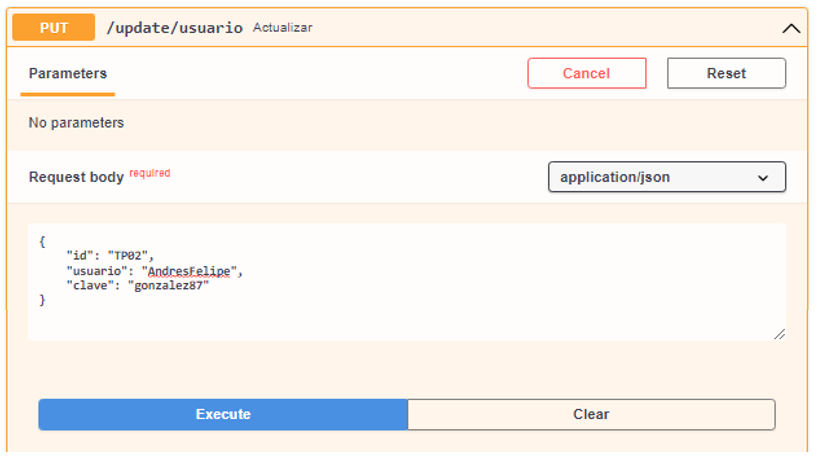
"usuario": "string",

"clave": "string"

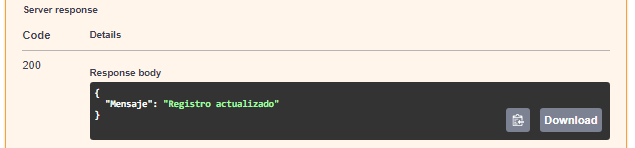
}

Se toma como parámetro de búsqueda el “id” y se actualizan los demás datos.

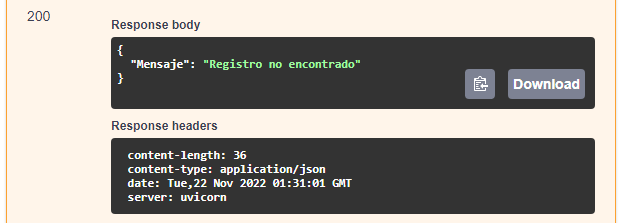
ruta: http://localhost:puerto/update/usuario



Al recibir y actualizar correctamente la información recibiremos un mensaje de confirmación:



Si no se encuentran los datos en el servidor, este nos devolverá un mensaje de error que nos indica que no se ha podido encontrar el registro:



**#DELETE**--DELETE

Requiere en su “Request body” la información a eliminar en formato JSON así:

{

"id": "string",

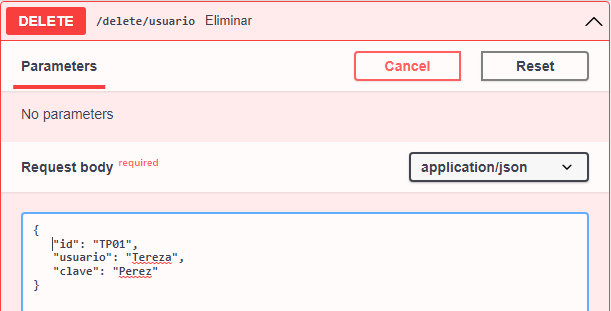
"usuario": "string",

"clave": "string"

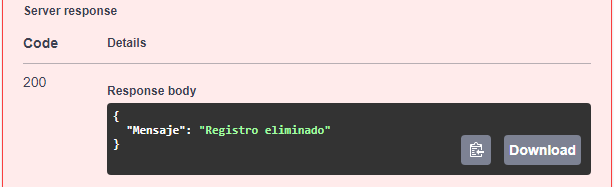
}

Se toma como parámetro de búsqueda el “id” y se eliminan todos los datos.

ruta: http://localhost:puerto/delete/usuario



Al recibir el comando y eliminar correctamente la información recibiremos un mensaje de confirmación:



**ApiEmpleado**

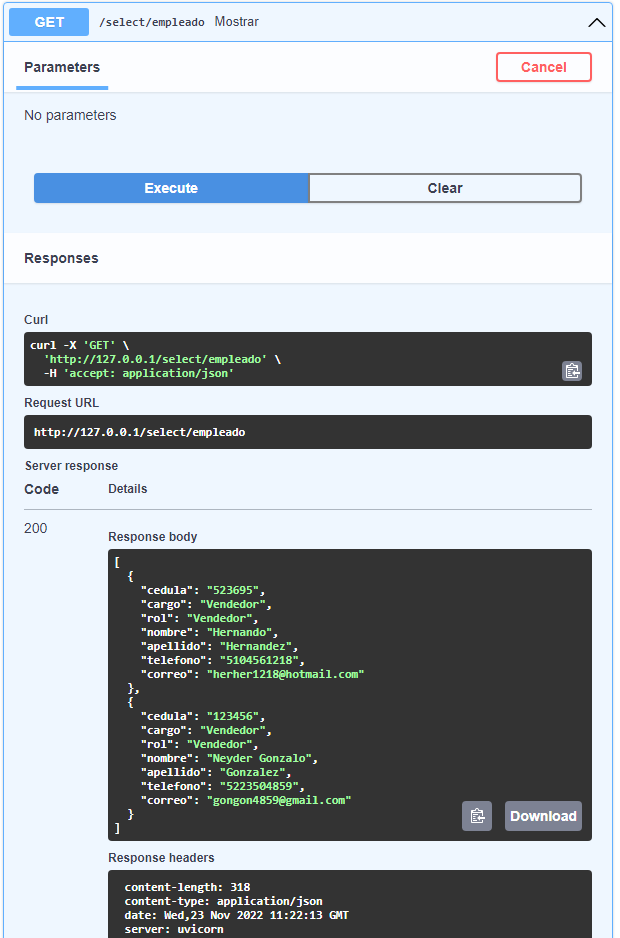
La API “ApiEmpleado” tiene implementados los métodos básicos CRUD. El archivo empleado.json temporalmente es usado para obtener y guardar datos.

**#GET**--READ

No requiere ningún parámetro y devuelve una lista de objetos de tipo **empleado** declarado dentro de la API

ruta: http://localhost:puerto/select/empleado

puerto: remplazamos por el puerto usado al lanzar el servidor **uvicorn**



**#POST**--CREATE

Requiere en su “Request body” la información a guardar en formato JSON así:

{

"cedula": "string",

"cargo": "string",

"rol": "string",

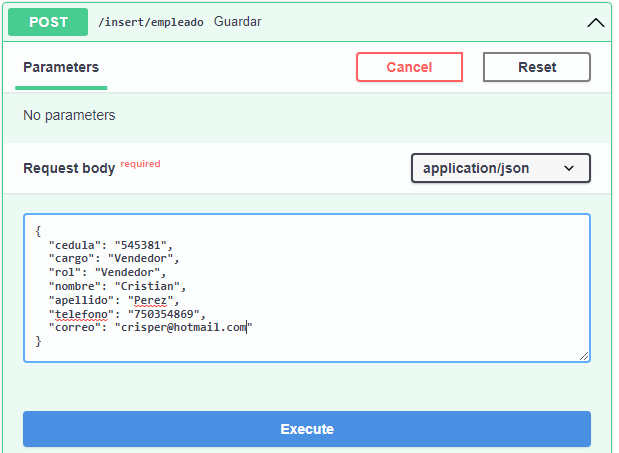
"nombre": "string",

"apellido": "string",

"telefono": "string",

"correo": "string"

}

****

**#PUT**--UPDATE

Requiere en su “Request body” la información a actualizar en formato JSON así:

{

"cedula": "string",

"cargo": "string",

"rol": "string",

"nombre": "string",

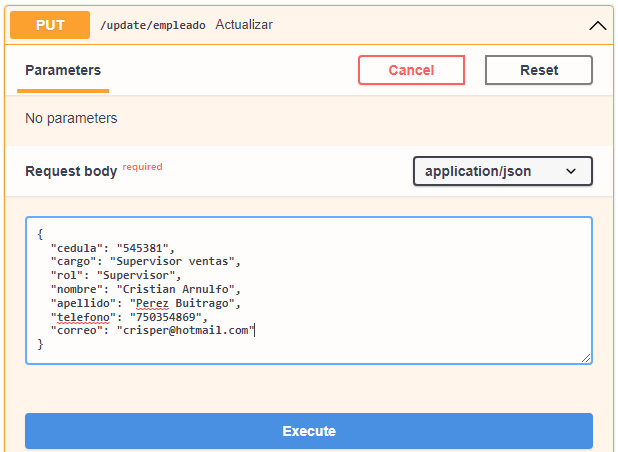
"apellido": "string",

"telefono": "string",

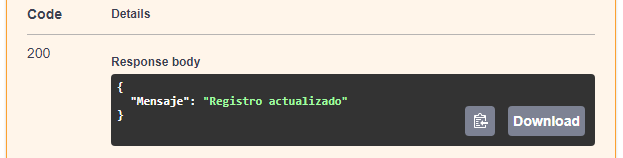
"correo": "string"

}

Se toma como parámetro de búsqueda el campo “cedula” y se actualizan los demás datos.

****

Al recibir y actualizar correctamente la información recibiremos un mensaje de confirmación:



**#DELETE**—DELETE

Requiere en su “Request body” la información a eliminar en formato JSON así:

{

"cedula": "string",

"cargo": "string",

"rol": "string",

"nombre": "string",

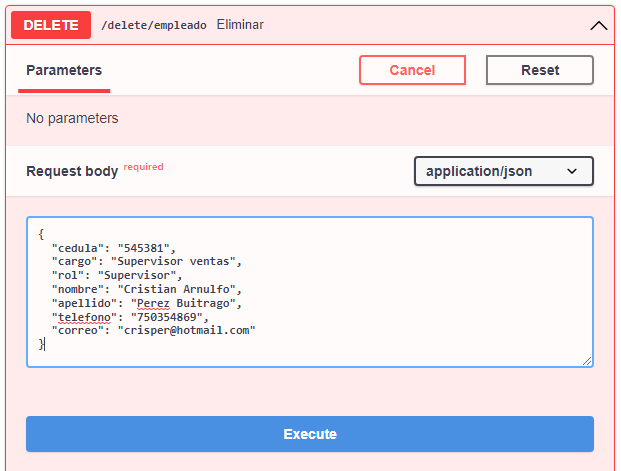
"apellido": "string",

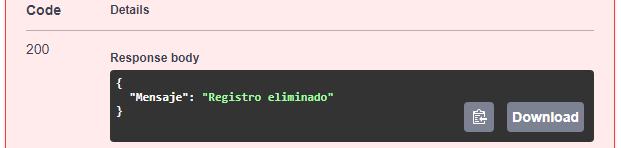
"telefono": "string",

"correo": "string"

}

Se toma como parámetro de búsqueda el campo “cedula” y se eliminan todos los datos.

 Al recibir el comando y eliminar correctamente la información recibiremos un mensaje de confirmación:



**ApiVehiculo**

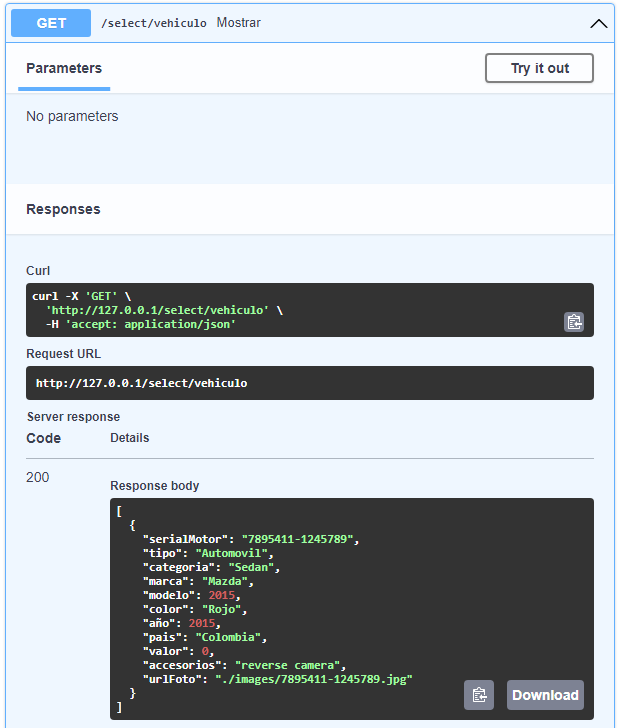
La API “ApiVehiculo” tiene implementados los métodos básicos CRUD. El archivo vehiculo.json temporalmente es usado para obtener y guardar datos.

**#GET**--READ

No requiere ningún parámetro y devuelve una lista de objetos de tipo **vehiculo** declarado dentro de la API

ruta: http://localhost:puerto/select/vehiculo

puerto: remplazamos por el puerto usado al lanzar el servidor **uvicorn**

****

**#POST**--CREATE

Requiere en su “Request body” la información a guardar en formato JSON así:

{

"serialMotor": "string",

"tipo": "string",

"categoria": "string",

"marca": "string",

"modelo": int,

"color": "string",

"año": int,

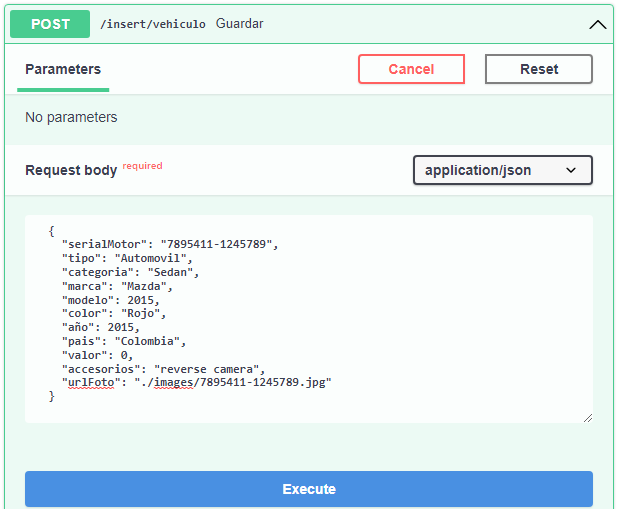
"pais": "string",

"valor": int,

"accesorios": "string",

"urlFoto": "string"

}



**#PUT**--UPDATE

Requiere en su “Request body” la información a actualizar en formato JSON así:

{

"serialMotor": "string",

"tipo": "string",

"categoria": "string",

"marca": "string",

"modelo": int,

"color": "string",

"año": int,

"pais": "string",

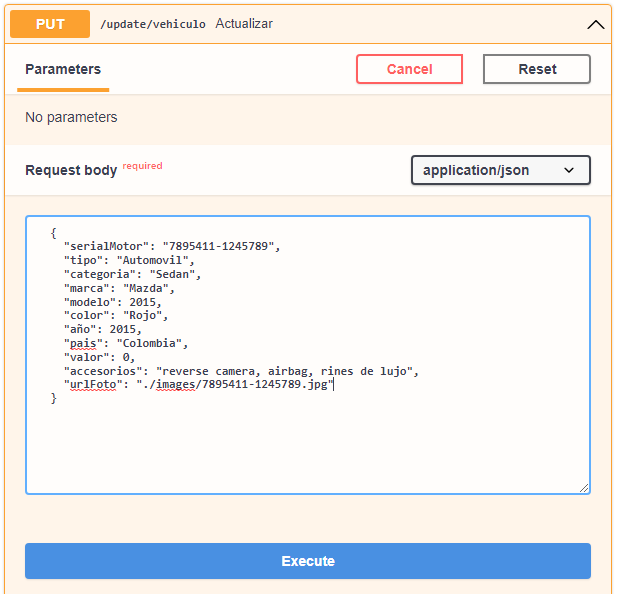
"valor": int,

"accesorios": "string",

"urlFoto": "string"

}

Se toma como parámetro de búsqueda el campo “serialMotor” y se actualizan los demás datos.



**#DELETE**—DELETE

Requiere en su “Request body” la información a eliminar en formato JSON así:

{

"serialMotor": "string",

"tipo": "string",

"categoria": "string",

"marca": "string",

"modelo": int,

"color": "string",

"año": int,

"pais": "string",

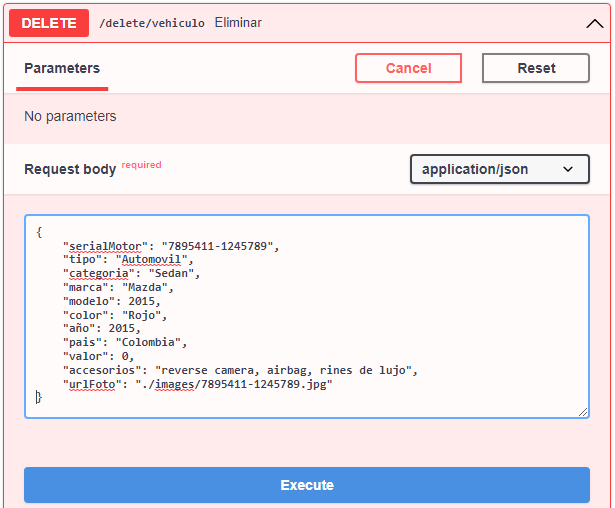
"valor": int,

"accesorios": "string",

"urlFoto": "string"

}

Se toma como parámetro de búsqueda el campo “serialMotor” y se eliminan todos los datos.



**ApiEnvio**

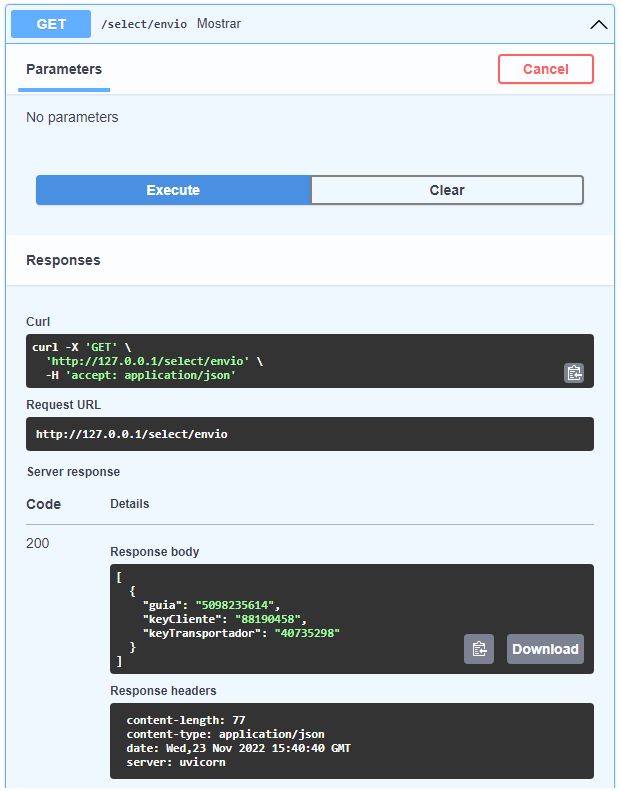
La API “ApiEnvio” tiene implementados los métodos básicos CRUD. El archivo envio.json temporalmente es usado para obtener y guardar datos.

**#GET**--READ

No requiere ningún parámetro y devuelve una lista de objetos de tipo **envio** declarado dentro de la API

ruta: http://localhost:puerto/select/envio

puerto: remplazamos por el puerto usado al lanzar el servidor **uvicorn**

****

**#POST**--CREATE

Requiere en su “Request body” la información a guardar en formato JSON así:

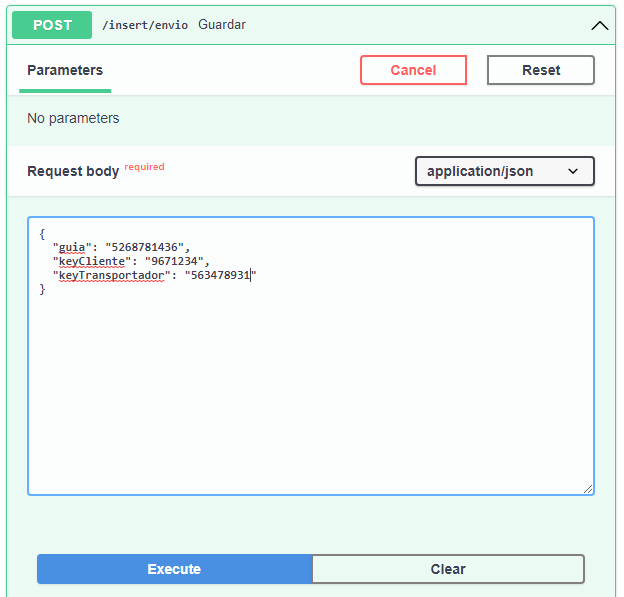
{

"guia": "string",

"keyCliente": "string",

"keyTransportador": "string"

}



**#PUT**--UPDATE

Requiere en su “Request body” la información a actualizar en formato JSON así:

{

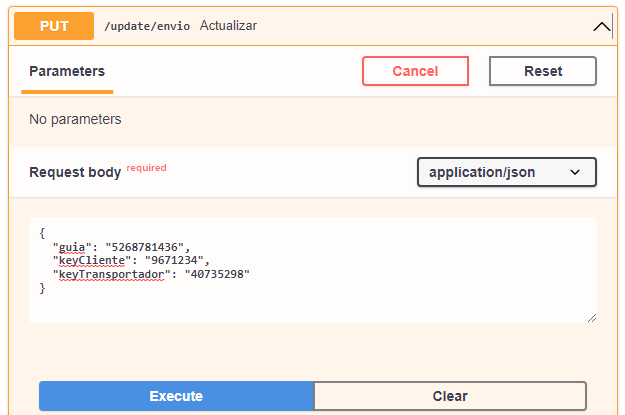
"guia": "string",

"keyCliente": "string",

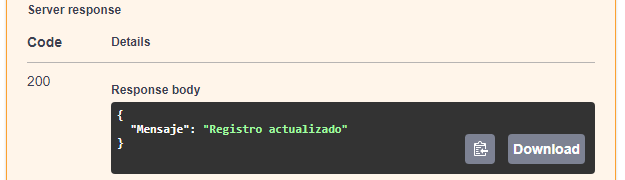
"keyTransportador": "string"

}

Se toma como parámetro de búsqueda el campo “guia” y se actualizan los demás datos.



Al recibir y actualizar correctamente la información recibiremos un mensaje de confirmación:



**#DELETE**—DELETE

Requiere en su “Request body” la información a eliminar en formato JSON así:

{

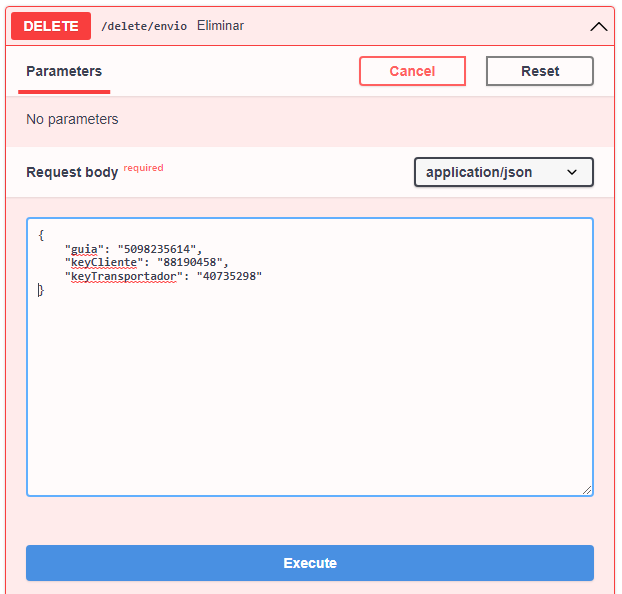
"guia": "string",

"keyCliente": "string",

"keyTransportador": "string"

}

Se toma como parámetro de búsqueda el campo “guia” y se eliminan todos los datos.



**ApiTranspotador**

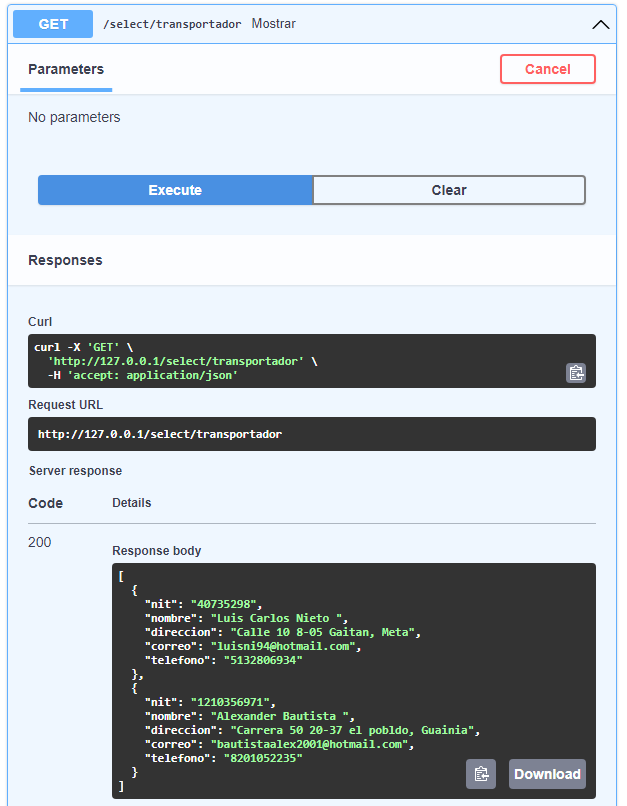
La API “ApiTransportador” tiene implementados los métodos básicos CRUD. El archivo transportador.json temporalmente es usado para obtener y guardar datos.

**#GET**--READ

No requiere ningún parámetro y devuelve una lista de objetos de tipo **transportador** declarado dentro de la API

ruta: http://localhost:puerto/select/transportador

puerto: remplazamos por el puerto usado al lanzar el servidor **uvicorn**



**#POST**--CREATE

Requiere en su “Request body” la información a guardar en formato JSON así:

{

"nit": "string",

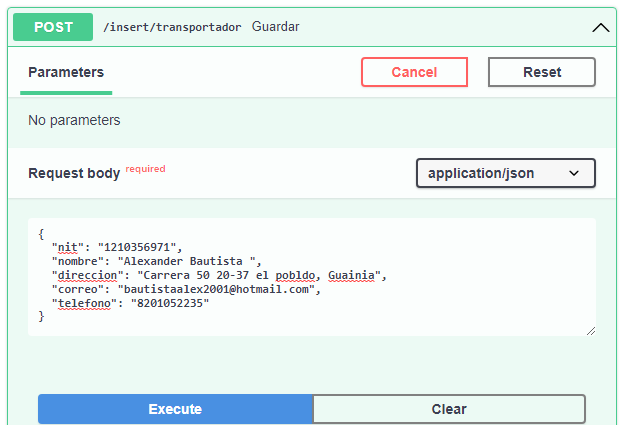
"nombre": "string",

"direccion": "string",

"correo": "string",

"telefono": "string"

}



**#PUT**--UPDATE

Requiere en su “Request body” la información a actualizar en formato JSON así:

{

"nit": "string",

"nombre": "string",

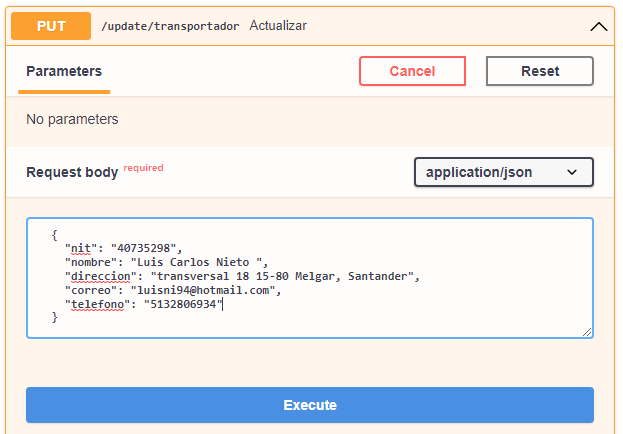
"direccion": "string",

"correo": "string",

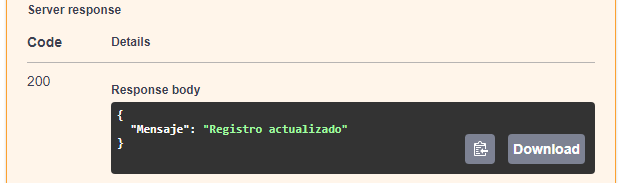
"telefono": "string"

}

Se toma como parámetro de búsqueda el campo “nit” y se actualizan los demás datos.



Al recibir y actualizar correctamente la información recibiremos un mensaje de confirmación:



**#DELETE**—DELETE

Requiere en su “Request body” la información a eliminar en formato JSON así:

{

"nit": "string",

"nombre": "string",

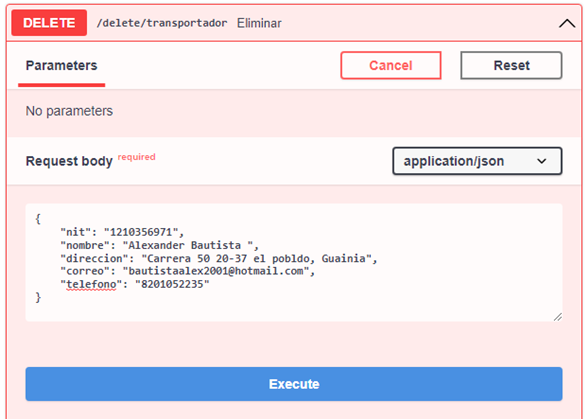
"direccion": "string",

"correo": "string",

"telefono": "string"

}

Se toma como parámetro de búsqueda el campo “nit” y se eliminan todos los datos.



Al recibir el comando y eliminar correctamente la información recibiremos un mensaje de confirmación:

