

Recelide API operaciones



Definición de la API

Versión 1.0.3

Índice

1. Introducción	7
1.1. Propósito.....	8
1.2. Alcance.....	9
2. Consideraciones sobre la API.....	10
2.1. Métodos HTTP	11
2.2. Respuestas HTTP.....	12
3. Definición de los métodos del servicio web.....	14
3.1. Métodos de identificación	15
3.2. Métodos de operación.....	17
3.3. Método LimpiezaDesinfeccion	18
3.3.1. LIMPIEZADESINFECCION /GET	18
3.3.2. LIMPIEZADESINFECCION /POST	19
3.4. Método SoloLimpieza.....	20
3.4.1. SOLOLIMPIEZA /GET	20
3.4.2. SOLOLIMPIEZA /POST.....	21
3.5. Método SoloDesinfeccion	22
3.5.1. SOLODESINFECCION /GET	22
3.5.2. SOLODESINFECCION /POST.....	23
3.6. Método Vehiculo	24
3.6.1. VEHICULO /GET	24
3.7. Método Conductor.....	25
3.7.1. CONDUCTOR /GET	25
3.8. MétodoCodigo.....	27
3.8.1. METODODESINFECTANTELIST /GET	27
3.8.2. TIPOIDENTIFICACIONVEHICULOLIST /GET	28
3.8.3. TIPOVEHICULOLIST /GET.....	29
4. Modelos de datos.....	31
4.1. CodigoNombreModel	32
4.2. ConductorModel.....	33
4.3. IdentificacionVehiculoModel	34
4.4. LimpiezaDesinfeccionGetModel	35
4.5. LimpiezaDesinfeccionPostModel	37
4.6. ResponsableModel	39

4.7. SoloDesinfeccionGetModel	40
4.8. SoloDesinfeccionPostModel	42
4.9. SoloLimpiezaGetModel.....	44
4.10. SoloLimpiezaPostModel.....	45
4.11. VehiculoGetModel	47

Revisiones

Fecha	Versión	Realizado por	Resumen de revisión
26/01/2024	1.0.0	GSIAP Ttec	Creación del documento.
31/01/2024	1.0.1	GSIAP Ttec	Primera versión
9/02/2024	1.0.2	GSIAP Ttec	Segunda versión
29/02/2024	1.0.2rev01	GSIAP Ttec	Definición de datos e incorporación de nuevos métodos
6/3/2024	1.0.3	GSIAP Ttec	Modificación de la definición SoloLimpiezaGetModel. Aparece un campo nuevo Vehiculo.

1. Introducción

El Real Decreto 638/2019 de 20 de noviembre de 2019 establece las condiciones básicas que deben cumplir los centros de limpieza y desinfección de los vehículos dedicados al transporte por carretera de animales vivos, de productos para la alimentación de animales de producción y de subproductos de origen animal no destinados al consumo humano, y, además, define la creación del Registro nacional de centros de limpieza y desinfección.

Este Registro Nacional de Centros de Limpieza y Desinfección, se constituye como un registro único formado por los datos recogidos en las bases de datos de las comunidades autónomas que incluirá la información de cada centro de limpieza y desinfección de vehículos ubicados en su ámbito territorial, mediante su agregación en un sistema central administrada por el MAPA, contendrá los datos mínimos que se indican en el RD y será puesto a disposición del público. La autoridad competente mantendrá actualizado el registro de todos los centros de limpieza y desinfección conforme a este RD.

1.1. Propósito

Mediante el presente encargo se pretende dar cumplimiento a lo marcado legislativamente en relación a la disposición por parte de la Dirección General de un Registro Nacional de Centros de Limpieza y desinfección.

Una de las tareas del encargo consiste en el desarrollo de una API de servicios web que permitan la grabación de las operaciones de limpieza y desinfección que realizan los centros del registro.

Es importante recalcar que las operaciones de limpieza y limpieza y desinfección que se realicen en cada centro se deberán grabar en RECELIDE, o comunicar a RECELIDE por medio del WS definido en este documento, en caso de usar este Servicio, de forma que cada operación quede registrada en RECELIDE antes de que el vehículo abandone el Centro, dado que dicho registro constituye la única justificación válida de las operaciones realizadas (1).

(1).- Artículo 7 punto 1 RD Real Decreto 208/2023, de 28 de marzo, por el que se modifica el Real Decreto 638/2019, de 8 de noviembre, por el que se establecen las condiciones básicas que deben cumplir los centros de limpieza y desinfección de los vehículos dedicados al transporte por carretera de animales vivos, productos para la alimentación de animales de producción y subproductos de origen animal no destinados al consumo humano, y se crea el Registro nacional de centros de limpieza y desinfección)..

1.2. Alcance

El alcance de este documento es servir de guía para el desarrollo del servicio web.

Este documento recoge el detalle de la versión actual de RECELIDE, definiendo por tanto el protocolo de comunicación en base a la estructura de datos de esta versión y al funcionamiento actual de la aplicación.

2. Consideraciones sobre la API

Una API es una interfaz de programación de aplicaciones (del inglés API: Application Programming Interface).

Las APIs permiten el acceso a características de bajo nivel, detallando solamente la forma en que cada rutina debe ser llevada a cabo y la funcionalidad que brinda, sin otorgar información acerca de cómo se lleva a cabo la tarea.

La API de RECELIDE, con el objeto de ser accesible desde internet, utiliza el protocolo de comunicación HTTP. El protocolo de transferencia de hipertexto (en inglés, Hypertext Transfer Protocol, abreviado HTTP) es un protocolo de comunicación que permite la transferencia de información a través de archivos (XML, HTML...) en la World Wide Web.

En el desarrollo de la API web de RECELIDE se ha utilizado el modelo de arquitectura REST.

La adopción de la arquitectura REST obliga a la utilización de unos métodos definidos por el protocolo HTTP que indican al servidor la acción que se desea realizar sobre uno o varios recursos en concreto. A estos métodos también se les conoce por el nombre de “verbos”.

2.1. Métodos HTTP

A continuación, se detallan los más comunes y su uso.

Método GET

Este método se utiliza cuando se desea conseguir un archivo o recurso que se encuentra en un servidor web.

Este método devuelve las cabeceras que contienen los metadatos del recurso solicitado y el recurso en sí.

Método POST

Este método se utiliza para enviar información al servidor y que lo enviado sea almacenado como un “hijo” o subelemento de un elemento o recurso ya existentes en el servidor.

Este método se usa para enviar información a un recurso web del servidor, pero no para cargar/crear un elemento nuevo como tal.

Método PUT

Crea un nuevo recurso al servidor, o en caso de que el objeto ya exista en el servidor, reemplaza al recurso existente con el recurso que se está enviando.

Método PATCH

Se utiliza para aplicar modificaciones parciales a un recurso

Método DELETE

Este método le solicita al servidor web que se borre un recurso en específico.

2.2. Respuestas HTTP

Una vez recibida una solicitud el servidor debe enviar la correspondiente respuesta que consta, entre otras, de estas partes:

- El código de respuesta. Indica el resultado del procesamiento de la solicitud
- Las cabeceras. Son metadatos con información adicional sobre la solicitud y la respuesta.
- Cuerpo del mensaje. Generalmente contiene los datos que se intercambian entre el cliente y el servidor, por ejemplo, aquí se incluirían los datos que ha solicitado una petición GET.

Los códigos de respuesta están formados por un número que indica el resultado de la solicitud y un mensaje descriptivo del código. Cada código tiene un significado concreto y se organizan en grupos en los que la centena a la que pertenecen determina el tipo de respuesta que da el servidor.

- **Códigos con centena 1X.** Respuesta informativa. Indica que la petición ha sido recibida y se está procesando.
- **Códigos doscientos:** Respuesta correcta. Indica que la petición ha sido procesada correctamente. Las más corrientes son:
 - o **200 OK.** Respuesta estándar para peticiones correctas.
 - o **201 Created.** La petición ha sido completada y ha resultado en la creación de un nuevo recurso.
 - o **202 Accepted.** La petición ha sido aceptada para procesamiento, pero este no ha sido completado. La petición eventualmente puede no haber sido satisfecha, ya que podría ser no permitida o prohibida cuando el procesamiento tenga lugar. Se estará a la espera de que todo sea correcto cuando se procese.
- **Códigos trescientos:** Respuesta de redirección. Indica que el cliente necesita realizar más acciones para finalizar la petición.
- **Códigos cuatrocientos:** Errores causados por el cliente. Indica que ha habido un error en el procesado de la petición a causa de que el cliente ha hecho algo mal. Los más comunes son:
 - o **400 Bad Request-** El servidor no procesará la solicitud, porque no puede, o no debe, debido a algo que es percibido como un error del cliente (ej: solicitud malformada, sintaxis errónea, etc). La solicitud contiene sintaxis errónea y no debería repetirse.
 - o **401 Authorization Required.** Similar al 403 Forbidden, pero se utiliza específicamente cuando la autenticación del cliente ha fallado (error en usuario o contraseña) o aún no se ha realizado.
 - o **403 Forbidden.** La solicitud fue legal, pero el servidor rehúsa responderla dado que el cliente no tiene los privilegios para realizarla. En contraste a una respuesta 401 No autorizado, autenticarse previamente no va a cambiar la respuesta.

- o **404 Not Found.** Recurso no encontrado. Se utiliza cuando el servidor web no encuentra la página o recurso solicitado.
- Códigos quinientos: Errores causados por el servidor. Indica que ha habido un error en el procesamiento de la petición a causa de un fallo en el servidor.
 - o **500 Internal Server Error.** Es un código comúnmente emitido por aplicaciones web cuando se encuentran con situaciones de error ajenas a la naturaleza del servidor web.
 - o **501 Not Implemented.** El servidor no soporta alguna funcionalidad necesaria para responder a la solicitud del navegador (como por ejemplo el método utilizado para la petición).
 - o **502 Bad Gateway.** El servidor está actuando de proxy o gateway y ha recibido una respuesta inválida del otro servidor, por lo que no puede responder adecuadamente a la petición del cliente.

3. Definición de los métodos del servicio web

El servicio web contará con métodos destinados a la comunicación de las diferentes operaciones que pueden realizar los centros y otros destinados a la identificación del usuario que realiza la operación. Esta identificación permitirá controlar la seguridad y la autorización de uso de cada uno de los servicios de comunicación de operaciones.

Actualmente existen tres tipos de operaciones:

- Operaciones de solo limpieza. Consisten en la eliminación mecánica y química de la suciedad del vehículo de transporte.
- Operaciones de solo desinfección. Son aquellas que producen la desinfección del vehículo de transporte.
- Operaciones de limpieza y desinfección. Son operaciones conjuntas que producen los efectos de las dos anteriores.

Las operaciones de solo limpieza solo se pueden realizar en centros de limpieza y desinfección y exclusivamente sobre vehículos autodesinfectantes (los vehículos autodesinfectantes son aquellos que tienen la capacidad de realizar una desinfección autónoma sin el concurso de un centro de limpieza y desinfección).

Las operaciones de solo desinfección solo pueden llevarse a cabo por vehículos autodesinfectantes y es requisito indispensable que se haya realizado una operación de solo limpieza previamente sobre el vehículo.

Las operaciones de limpieza y desinfección se realizan en los centros de limpieza y desinfección sobre cualquier tipo de vehículo.

Los métodos del servicio web permitirán realizar las comunicaciones de los tres tipos de operaciones e identificar a los usuarios de manera seguras y limitar sus operaciones a las autorizadas.

Los métodos son los siguientes:

- SoloLimpieza: Permitirá comunicar operaciones de solo limpieza
- SoloDesinfeccion: Permitirá comunicar operaciones de solo desinfección
- LimpiezaDesinfeccion: Permitirá comunicar operaciones de limpieza y desinfección.
- Identificacion: Permitirá identificarse al usuario para poder operar con la API.

3.1. Métodos de identificación

La identificación responderá a este esquema:

- 1- Acceder a la API RECELIDE
- 2- Identificarse.
 - a. REGEUS. La API RECELIDE utiliza el usuario y la contraseña de REGEUS para autenticar al usuario.
 - b. Token de seguridad. Una vez autenticado el usuario se le asigna un token para las posteriores comunicaciones. Cada una de las solicitudes debe venir acompañada por el token de seguridad.

A pesar de que todos los métodos serán visibles, se necesitará un determinado nivel de privilegios para utilizarlos. Estos privilegios corresponden con el perfil del usuario que se conecte. Por eso, es imprescindible contar con un usuario registrado en la aplicación para poder utilizar los servicios de RECELIDE API.

La API RECELIDE utiliza el usuario y la contraseña de REGEUS para autenticar al usuario.

Identificación solo tendrá un único método que servirá para realizar la identificación del usuario en la aplicación: Identificacion/Login

IDENTIFICACION /LOGIN

Se debe ejecutar el verbo POST del método Identificacion /Login.

Los métodos pueden precisar información adicional para poder ser utilizados:

- Parameters
- Request body

El apartado Parameters está destinado a albergar los parámetros en aquellos métodos en los que se deben proporcionar para poder usarse. En el caso de este método no se pueden proporcionar parámetros.

La información que necesita el método para ejecutarse se proporciona usando el cuerpo del mensaje ("Request body").

El cuerpo del mensaje estará formado por dos variables de tipo alfanumérico (string), una denominada usuario y otra password. Para utilizar este método tendremos que proporcionar el usuario y la contraseña mediante estas dos variables.

Usuario se corresponderá con nuestro usuario de REGEUS (el NIF) y la contraseña será la que está almacenada en REGEUS.

Tras la ejecución del método se obtendrá una respuesta según el resultado del procesado de la información.

La respuesta corresponderá a alguna de las respuestas codificadas HTTP (2.2 Respuestas HTTP):

Se podrá obtener:

- Error 401: Unauthorized, El usuario no está autorizado a ejecutar el método. En caso de que se produzca esta respuesta se acompañará de un mensaje explicativo de la causa del error "Usuario/contraseña incorrectos. Vuelva a intentarlo o consulte con un administrador".
- Error 400: Bad request. indica que el servidor procesará la solicitud porque no puede o no debe procesarla. Ello se debe a algo que entiende que se debe a un error del cliente (por ejemplo, solicitud mal formada, sintaxis errónea, sospecha de un mensaje de solicitud no válido o engañoso o cualquier otro motivo que haga pensar que la solicitud no debe procesarse).
- Mensaje 200: Correcto. Cuando se utiliza el método con un usuario y una contraseña que consiguen la validación se obtendrá un mensaje de ejecución correcta del método y además el token de seguridad que se deberá acompañar en el resto de operaciones de la API.

ADVERTENCIA: El token de seguridad nos identifica como nuestro usuario. Es muy importante no compartir este token con nadie porque cualquiera que lo tuviera podría hacerse pasar por nosotros.

3.2. Métodos de operación

Los métodos de operación se podrán utilizar tras la identificación y permitirán realizar la comunicación de las operaciones de limpieza y desinfección.

La secuencia completa se explica continuación:

- 1- Acceder a la API RECELIDE
- 2- Identificarse.
 - a. REGEUS. La API RECELIDE utiliza el usuario y la contraseña de REGEUS para autenticar al usuario.
 - b. Token de seguridad. Una vez autenticado el usuario se le asigna un token para las posteriores comunicaciones. Cada una de las solicitudes debe venir acompañada por el token de seguridad.
- 3- Operación. Seleccionar la operación que quiere realizar
 - a. Rellenar los parámetros necesarios
 - b. Rellenar el cuerpo de la solicitud (si es necesario)
 - c. Ejecutar
 - d. Obtener la respuesta

Los métodos tendrán diferentes procesos según el verbo utilizado:

- POST. Comunicar nuevos datos o actualizar los datos ya existentes relativos a la operación.
- PUT. Actualizar los datos ya existentes relativos a la operación
- GET. Obtener la información de los datos ya existentes relativos a la operación.

Es decir, en RECELIDE API se utilizan los verbos HTTP para las acciones que define el protocolo.
(2.1 Métodos HTTP)

Una vez tenemos registrado nuestro token de seguridad podemos utilizar los métodos para los que estamos autorizados.

3.3. Método LimpiezaDesinfeccion

Este método se utilizará para las operaciones que realicen la limpieza y la desinfección del vehículo.

Se utilizará con dos verbos:

- GET: para obtener información de operaciones ya comunicadas.
- POST: para comunicar nuevas operaciones o modificaciones sobre operaciones ya comunicadas.

3.3.1. LIMPIEZADESINFECCION /GET

Descripción: Permitirá obtener los datos de una operación de limpieza y desinfección.

URL: [APIOperaciones]/LimpiezaDesinfeccion

Verbo: GET

Parámetros:

- **codigo** (string). El código de la operación de limpieza y desinfección de la que se quieren obtener los datos.

Cuerpo: No es necesario proporcionar un cuerpo

Respuesta:

- **200: OK.** La petición es correcta y se acompañará con los datos del modelo **LimpiezaDesinfeccionGetModel**
- **Error 400: Bad request.** Algo hay en la solicitud que impide que se procese la petición. Revise la solicitud que ha enviado.
- **Error 401: Authorization required.** No se ha identificado correctamente antes de realizar la solicitud. Repita la solicitud tras haber realizado una autenticación correcta.
- **Error 403: Forbidden.** La solicitud es correcta pero el servidor rehúsa la respuesta porque el usuario que la solicita no tiene privilegios suficientes para obtener la información. Al contrario que en una respuesta 401 autenticarse de nuevo con las mismas credenciales no cambiará la respuesta.
- **Error 404: Not Found.** Recurso no encontrado. La operación de Solo limpieza que solicita no existe.
- **Error 500: Internal Server Error.** Ha ocurrido algún error que ha impedido que se procese la solicitud. Puede volver a intentarlo más tarde o informar en el buzón de RECELIDE.

Consideraciones:

- Según el usuario puede haber operaciones sobre las que no se pueda consultar. En tal caso recibirá un error 403.

3.3.2.LIMPIEZADESINFECCION /POST

Descripción: Permitirá añadir una operación de limpieza y desinfección nueva.

URL: [APIOperaciones]/LimpiezaDesinfeccion

Verbo: POST

Parámetros: No hay parámetros

Cuerpo: JSON LimpiezaDesinfeccionPostModel

Respuesta:

- **201: Created** La petición ha sido completada y ha resultado en la creación de una nueva operación. Devuelve **LimpiezaDesinfeccionGetModel** para comprobar los datos que se han guardado.
- **Error 400: Bad request**. Algo hay en la solicitud que impide que se procese la petición. Revise la solicitud que ha enviado.
- **Error 401: Authorization required**. No se ha identificado correctamente antes de realizar la solicitud. Repita la solicitud tras haber realizado una autenticación correcta.
- **Error 403: Forbidden**. La solicitud es correcta pero el servidor rehúsa la respuesta porque el usuario que la solicita no tiene privilegios suficientes para obtener la información. Al contrario que en una respuesta 401 autenticarse de nuevo con las mismas credenciales no cambiará la respuesta.
- **Error 404: Not Found**. Recurso no encontrado. La operación de Solo limpieza que solicita no existe.
- **Error 500: Internal Server Error**. Ha ocurrido algún error que ha impedido que se procese la solicitud. Puede volver a intentarlo más tarde o informar en el buzón de RECELIDE.

Consideraciones:

- Según el usuario puede haber datos de la operación que está comunicando sobre los que no se pueda consultar. En tal caso, recibirá un error 403.

3.4. Método SoloLimpieza

Este método se utilizará para la comunicación de operaciones que solo limpien el vehículo, pero no conlleven la desinfección del mismo. Esta operación solo puede ser realizada sobre vehículos autodesinfectantes

Se utilizará con dos verbos:

- GET: para obtener información de operaciones ya comunicadas.
- POST: para comunicar nuevas operaciones o modificaciones sobre operaciones ya comunicadas.

3.4.1. SOLOLIMPIEZA /GET

Descripción: Permitirá obtener los datos de una operación de solo limpieza.

URL: [APIOperaciones]/SoloLimpieza

Verbo: GET

Parámetros:

- **codigo** (string). El código de la operación de solo limpieza de la que se quieren obtener los datos.

Cuerpo: No es necesario proporcionar un cuerpo.

Respuesta:

- **200: OK.** La petición es correcta y se acompañará con los datos del modelo **SoloLimpiezaGetModel**
- **Error 400: Bad request.** Algo hay en la solicitud que impide que se procese la petición. Revise la solicitud que ha enviado.
- **Error 401: Authorization required.** No se ha identificado correctamente antes de realizar la solicitud. Repita la solicitud tras haber realizado una autenticación correcta.
- **Error 403: Forbidden.** La solicitud es correcta pero el servidor rehúsa la respuesta porque el usuario que la solicita no tiene privilegios suficientes para obtener la información. Al contrario que en una respuesta 401 autenticarse de nuevo con las mismas credenciales no cambiará la respuesta.
- **Error 404: Not Found.** Recurso no encontrado. La operación de Solo limpieza que solicita no existe.
- **Error 500: Internal Server Error.** Ha ocurrido algún error que ha impedido que se procese la solicitud. Puede volver a intentarlo más tarde o informar en el buzón de RECELIDE.

Consideraciones:

- Según el usuario puede haber operaciones sobre las que no se pueda consultar. En tal caso recibirá un error 403.

3.4.2.SOLOLIMPIEZA /POST

Descripción: Permitirá añadir una operación de solo limpieza nueva.

URL: [APIOperaciones]/SoloLimpieza

Verbo: POST

Parámetros: No hay parámetros

Cuerpo: JSON **SoloLimpiezaPostModel**

Respuesta:

- **201: Created** La petición ha sido completada y ha resultado en la creación de una nueva operación. Devuelve **SoloLimpiezaGetModel** para comprobar los datos que se han guardado.
- **Error 400: Bad request**. Algo hay en la solicitud que impide que se procese la petición. Revise la solicitud que ha enviado.
- **Error 401: Authorization required**. No se ha identificado correctamente antes de realizar la solicitud. Repita la solicitud tras haber realizado una autenticación correcta.
- **Error 403: Forbidden**. La solicitud es correcta pero el servidor rehúsa la respuesta porque el usuario que la solicita no tiene privilegios suficientes para obtener la información. Al contrario que en una respuesta 401 autenticarse de nuevo con las mismas credenciales no cambiará la respuesta.
- **Error 500: Internal Server Error**. Ha ocurrido algún error que ha impedido que se procese la solicitud. Puede volver a intentarlo más tarde o informar en el buzón de RECELIDE.

3.5. Método SoloDesinfeccion

Este método se utilizará para la comunicación de operaciones que solo realicen la desinfección del vehículo. Esta operación solo puede ser realizada sobre vehículos autodesinfectantes

Se utilizará con dos verbos:

- GET: para obtener información de operaciones ya comunicadas.
- POST: para comunicar nuevas operaciones o modificaciones sobre operaciones ya comunicadas.

3.5.1. SOLODESINFECCION /GET

Descripción: Permitirá obtener los datos de una operación de solo desinfeccion.

URL: [APIOperaciones]/SoloDesinfeccion

Verbo: GET

Parámetros:

- **codigo** (string). El código de la operación de solo desinfección de la que se quieren obtener los datos.

Cuerpo: No es necesario proporcionar un cuerpo.

Respuesta:

- **200: OK**. La petición es correcta y se acompañará con los datos del modelo **SoloDesinfeccionGetModel**
- **Error 400: Bad request**. Algo hay en la solicitud que impide que se procese la petición. Revise la solicitud que ha enviado.
- **Error 401: Authorization required**. No se ha identificado correctamente antes de realizar la solicitud. Repita la solicitud tras haber realizado una autenticación correcta.
- **Error 403: Forbidden**. La solicitud es correcta pero el servidor rehúsa la respuesta porque el usuario que la solicita no tiene privilegios suficientes para obtener la información. Al contrario que en una respuesta 401 autenticarse de nuevo con las mismas credenciales no cambiará la respuesta.
- **Error 404: Not Found**. Recurso no encontrado. La operación de Solo limpieza que solicita no existe.
- **Error 500: Internal Server Error**. Ha ocurrido algún error que ha impedido que se procese la solicitud. Puede volver a intentarlo más tarde o informar en el buzón de RECELIDE.

Consideraciones:

- Según el usuario puede haber operaciones sobre las que no se pueda consultar. En tal caso recibirá un error 403.

3.5.2.SOLODESINFECCION /POST

Descripción: Permitirá añadir una operación de solo desinfección nueva o modificar los datos de una ya existente

URL: [APIOperaciones]/SoloDesinfeccion

Verbo: POST

Parámetros: No hay parámetros

Cuerpo: JSON SoloDesinfeccionPostModel

Respuesta:

- **201: Created** La petición ha sido completada y ha resultado en la creación de una nueva operación. Devuelve **SoloDesinfeccionGetModel** para comprobar los datos que se han guardado.
- **Error 400: Bad request**. Algo hay en la solicitud que impide que se procese la petición. Revise la solicitud que ha enviado.
- **Error 401: Authorization required**. No se ha identificado correctamente antes de realizar la solicitud. Repita la solicitud tras haber realizado una autenticación correcta.
- **Error 403: Forbidden**. La solicitud es correcta pero el servidor rehúsa la respuesta porque el usuario que la solicita no tiene privilegios suficientes para obtener la información. Al contrario que en una respuesta 401 autenticarse de nuevo con las mismas credenciales no cambiará la respuesta.
- **Error 500: Internal Server Error**. Ha ocurrido algún error que ha impedido que se procese la solicitud. Puede volver a intentarlo más tarde o informar en el buzón de RECELIDE.

3.6. Método Vehículo

Este método se utilizará para la consulta de vehículos registrados en la aplicación

Se utilizará con un verbo:

- GET: para obtener información de vehículos registrados.

3.6.1. VEHICULO /GET

Descripción: Permite obtener los datos de un vehículo.

URL: [APIOperaciones]/Vehiculo

Verbo: GET

Parámetros:

- **matricula** (string). La matrícula del vehículo del que se quieren obtener los datos.

Cuerpo: No es necesario proporcionar un cuerpo.

Respuesta:

- **200: OK.** La petición es correcta y se acompañará con los datos del modelo **VehiculoGetModel**
- **Error 400: Bad request.** Algo hay en la solicitud que impide que se procese la petición. Revise la solicitud que ha enviado.
- **Error 401: Authorization required.** No se ha identificado correctamente antes de realizar la solicitud. Repita la solicitud tras haber realizado una autenticación correcta.
- **Error 403: Forbidden.** La solicitud es correcta pero el servidor rehúsa la respuesta porque el usuario que la solicita no tiene privilegios suficientes para obtener la información. Al contrario que en una respuesta 401 autenticarse de nuevo con las mismas credenciales no cambiará la respuesta.
- **Error 404: Not Found.** Recurso no encontrado. La operación de Solo limpieza que solicita no existe.
- **Error 500: Internal Server Error.** Ha ocurrido algún error que ha impedido que se procese la solicitud. Puede volver a intentarlo más tarde o informar en el buzón de RECELIDE.

3.7. Método Conductor

Este método se utilizará para la consulta de los conductores registrados en la aplicación

Se utilizará con un verbo:

- GET: para obtener información de conductores registrados.

3.7.1. CONDUCTOR /GET

Descripción: Permite obtener los datos de un conductor.

URL: [APIOperaciones]/Conductor

Verbo: GET

Parámetros:

- **nif** (string). El NIF/NIE del conductor del que se quiere comprobar que existe en la aplicación.

Cuerpo: No es necesario proporcionar un cuerpo.

Respuesta:

- **200: OK.** La petición es correcta y se acompañará con los datos del modelo **CodigoNombreModel**

El modelo se utiliza para mostrar listas formadas por un código y un nombre.

Nombre	Tipo	Observaciones
Codigo	Int	
Nombre	String(50)	

- ConductorModel
- **Error 400: Bad request.** Algo hay en la solicitud que impide que se procese la petición. Revise la solicitud que ha enviado.
- **Error 401: Authorization required.** No se ha identificado correctamente antes de realizar la solicitud. Repita la solicitud tras haber realizado una autenticación correcta.
- **Error 403: Forbidden.** La solicitud es correcta pero el servidor rehúsa la respuesta porque el usuario que la solicita no tiene privilegios suficientes para obtener la

información. Al contrario que en una respuesta 401 autenticarse de nuevo con las mismas credenciales no cambiará la respuesta.

- **Error 404: Not Found**. Recurso no encontrado. La operación de Solo limpieza que solicita no existe.
- **Error 500: Internal Server Error**. Ha ocurrido algún error que ha impedido que se procese la solicitud. Puede volver a intentarlo más tarde o informar en el buzón de RECELIDE.

3.8. Método Código

Este método se utilizará para la consulta de los diferentes conjuntos de códigos que utiliza la aplicación.

Se utilizará con un verbo:

- GET: para obtener la información de cada una de las acciones posibles.

3.8.1. **METODODESINFECTANTELIST /GET**

Descripción: Permite obtener la lista de los posibles métodos desinfectantes.

URL: [APIOperaciones]/Codigo/MetodoDesinfectanteList

Verbo: GET

Parámetros: No hay parámetros

Cuerpo: No es necesario proporcionar un cuerpo.

Respuesta:

3.9.200: **OK**. La petición es correcta y se acompañará con una lista del modelo **CodigoNombreModelCodigoNombreModel**

El modelo se utiliza para mostrar listas formadas por un código y un nombre.

Nombre	Tipo	Observaciones
Codigo	Int	
Nombre	String(50)	

- ConductorModel con los métodos de desinfección.
- **Error 400: Bad request**. Algo hay en la solicitud que impide que se procese la petición. Revise la solicitud que ha enviado.
- **Error 401: Authorization required**. No se ha identificado correctamente antes de realizar la solicitud. Repita la solicitud tras haber realizado una autenticación correcta.
- **Error 403: Forbidden**. La solicitud es correcta pero el servidor rehúsa la respuesta porque el usuario que la solicita no tiene privilegios suficientes para obtener la información. Al contrario que en una respuesta 401 autenticarse de nuevo con las mismas credenciales no cambiará la respuesta.
- **Error 404: Not Found**. Recurso no encontrado. La operación de Solo limpieza que solicita no existe.
- **Error 500: Internal Server Error**. Ha ocurrido algún error que ha impedido que se procese la solicitud. Puede volver a intentarlo más tarde o informar en el buzón de RECELIDE.

3.9.1. **TIPOIDENTIFICACIONVEHICULOLIST /GET**

Descripción: Permite obtener la lista de los posibles tipos de identificación que tiene un vehículo.

URL: [APIOperaciones]/Codigo/TipoIdentificacionVehiculoList

Verbo: GET

Parámetros: No hay parámetros

Cuerpo: No es necesario proporcionar un cuerpo.

Respuesta:

3.10. 200: OK. La petición es correcta y se acompañará con una lista del modelo **CodigoNombreModelCodigoNombreModel**

El modelo se utiliza para mostrar listas formadas por un código y un nombre.

Nombre	Tipo	Observaciones
Codigo	Int	
Nombre	String(50)	

- ConductorModel con los tipos de identificación que puede tener un vehículo.
- **Error 400: Bad request**. Algo hay en la solicitud que impide que se procese la petición. Revise la solicitud que ha enviado.
- **Error 401: Authorization required**. No se ha identificado correctamente antes de realizar la solicitud. Repita la solicitud tras haber realizado una autenticación correcta.
- **Error 403: Forbidden**. La solicitud es correcta pero el servidor rehúsa la respuesta porque el usuario que la solicita no tiene privilegios suficientes para obtener la información. Al contrario que en una respuesta 401 autenticarse de nuevo con las mismas credenciales no cambiará la respuesta.
- **Error 404: Not Found**. Recurso no encontrado. La operación de Solo limpieza que solicita no existe.
- **Error 500: Internal Server Error**. Ha ocurrido algún error que ha impedido que se procese la solicitud. Puede volver a intentarlo más tarde o informar en el buzón de RECELIDE.

3.10.1. TIPOVEHICULOLIST /GET

Descripción: Permite obtener la lista de los posibles tipos de vehículo que contempla la aplicación.

URL: [APIOperaciones]/Codigo/TipoVehiculoList

Verbo: GET

Parámetros: No hay parámetros

Cuerpo: No es necesario proporcionar un cuerpo.

Respuesta:

3.11. 200: OK. La petición es correcta y se acompañará con una lista del modelo `CodigoNombreModelCodigoNombreModel`

El modelo se utiliza para mostrar listas formadas por un código y un nombre.

Nombre	Tipo	Observaciones
Codigo	Int	
Nombre	String(50)	

- ConductorModel con los tipos de vehiculo.
- **Error 400: Bad request**. Algo hay en la solicitud que impide que se procese la petición. Revise la solicitud que ha enviado.
- **Error 401: Authorization required**. No se ha identificado correctamente antes de realizar la solicitud. Repita la solicitud tras haber realizado una autenticación correcta.
- **Error 403: Forbidden**. La solicitud es correcta pero el servidor rehúsa la respuesta porque el usuario que la solicita no tiene privilegios suficientes para obtener la información. Al contrario que en una respuesta 401 autenticarse de nuevo con las mismas credenciales no cambiará la respuesta.
- **Error 404: Not Found**. Recurso no encontrado. La operación de Solo limpieza que solicita no existe.
- **Error 500: Internal Server Error**. Ha ocurrido algún error que ha impedido que se procese la solicitud. Puede volver a intentarlo más tarde o informar en el buzón de RECELIDE.

4. Modelos de datos

A continuación, se presentan, por orden alfabético, cada uno de los modelos de datos que se utilizan en la API.

- CodigoNombreModel
- ConductorModel
- IdentificacionVehiculoModel
- LimpiezaDesinfeccionGetModel
- LimpiezaDesinfeccionPostModel
- ResponsableModel
- SoloDesinfeccionGetModel
- SoloDesinfeccionPostModel
- SoloLimpiezaGetModel
- SoloLimpiezaPostModel
- VehiculoGetModel

4.1. CódigoNombreModel

El modelo se utiliza para mostrar listas formadas por un código y un nombre.

Nombre	Tipo	Observaciones
Código	Int	
Nombre	String(50)	

4.2. ConductorModel

El modelo sirve para definir un conductor registrado en la aplicación.

Nombre	Tipo	Obligatorio	Observaciones
Apellido1	String (50)	No/Sí	Es obligatorio cuando el conductor no existe en RECELIDE
Apellido2	String (50)	No/Sí	Es obligatorio cuando el conductor no existe en RECELIDE
Nif	String (9)	Sí	Debe pasar la validación de NIF de persona física
Nombre	String (50)	No/Sí	Es obligatorio cuando el conductor no existe en RECELIDE

4.3. IdentificacionVehiculoModel

El modelo define las identificaciones posibles que puede tener un vehículo.

Nombre	Tipo	Obligatorio	Observaciones
TipoidentificacionId	Int	<i>Sí</i>	<i>Identificación del tipo de identificación del vehículo. Corresponderá con el campo codigo de los códigos que se obtienen de TIPOIDENTIFICACIONVEHICULOLIST/GET</i>
TipoidentificacionNombre	String(50)	<i>No</i>	<i>Nombre del tipo de identificación del vehículo. Corresponderá con el campo nombre de los códigos que se obtienen de TIPOIDENTIFICACIONVEHICULOLIST/GET</i>
Identificacion	String(20)	<i>Sí</i>	<i>La identificación validará según el VehiculoPrincipalTipoidentificacion aportado.</i>

4.4. LimpiezaDesinfeccionGetModel

Este modelo de datos define la información de una operación de Limpieza y Desinfección:

Nombre	Tipo	Observaciones
AltaSistemaFechaHora	DateTime	Fecha de grabación de la operación en el sistema. El formato es 'aaaa-mm-dd hh:mm:ss'
CentroCodigo	String (50)	Código RECELIDE del centro en el que se ha realizado la operación de solo limpieza
CentroNombre	String(255)	Nombre del centro en el que se ha realizado la operación de solo limpieza
Conductor	ConductorModel	Datos del conductor del vehículo implicado en la operación. Consulte la definición de ¡Error! No se encuentra el origen de la referencia.
FinFecha	Date	Fecha de finalización de la operación. El formato es 'aaaa-mm-dd'
FinHora	Time	Hora de finalización de la operación. El formato es 'hh:mm'
InicioFecha	Date	Fecha de inicio de la operación. El formato es 'aaaa-mm-dd'
InicioHora	Time	Hora de inicio de la operación. El formato es 'hh:mm'
Identificador	String(20)	Identificador de la operación de limpieza y desinfección
MetodoDesinfeccionId	Int	Identificador del método de desinfección, según los códigos de METODODESINFECTANTELIST /GET
MetodoDesinfeccionNombre	String(255)	Nombre del método de desinfección, según los códigos de METODODESINFECTANTELIST /GET
MetodoDesinfeccionBiocida	String(255)	Nombre del biocida empleado. Solo estará relleno si se ha aplicado un método de desinfección de tipo biocida.
Movimientos	Array de String(50)	Lista de los códigos REMO relacionados con la operación.

Recelide

Precintos	Array de String(50)	<i>Lista de precintos que se han empleado en la operación</i>
RemolqueMatricula	String(20)	<i>Matrícula del remolque (si es que existe)</i>
Responsable	ResponsableModel	<i>Datos del responsable que ha realizado la operación. Consulte la definición de ResponsableModel</i>
VehiculoPrincipal	VehiculoModel	<i>Datos del vehículo implicado en la operación. Consulte VehiculoGetModel</i>

4.5. LimpiezaDesinfeccionPostModel

Este modelo de datos define la información que se debe aportar para añadir una nueva operación de Limpieza y Desinfección:

Nombre	Tipo	Obligatorio	Observaciones
CentroCodigo	String (50)	Sí	Código RECELIDE del centro en el que se ha realizado la operación de solo limpieza
Conductor	ConductorModel	Sí	Datos del conductor del vehículo implicado en la operación. Consulte la definición de ¡Error! No se encuentra el origen de la referencia.
FinFecha	Date	Sí	Fecha de finalización de la operación. El formato es 'aaaa-mm-dd'
FinHora	Time	Sí	Hora de finalización de la operación. El formato es 'hh:mm'
InicioFecha	Date	Sí	Fecha de inicio de la operación. El formato es 'aaaa-mm-dd'
InicioHora	Time	Sí	Hora de inicio de la operación. El formato es 'hh:mm'
MetodoDesinfeccionId	Int	Sí	Identificador del método de desinfección, según los códigos de METODODESINFECTANTELIST/GET
MetodoDesinfeccionBiocida	String(255)	No/Sí	Nombre del biocida empleado. Es obligatorio en caso de que se haya aplicado un método de desinfección de tipo biocida. (Determinado por el atributo MetodoDesinfeccion)
Movimientos	Array de String(50)	No	Lista de los códigos REMO relacionados con la operación.

Precintos	Array de String(50)	<i>Sí</i>	<i>Lista de precintos que se han empleado en la operación. Es obligatorio incluir al menos uno.</i>
RemolqueMatricula	String(20)	<i>No/Sí</i>	<i>Si el vehículo no lleva remolque no es obligatorio incluirla. Si se incluye la matrícula validará según las validaciones de matrícula.</i>
Responsable	ResponsableModel	<i>Sí</i>	<i>Datos del responsable que ha realizado la operación. Consulte la definición de ResponsableModel</i>
VehiculoPrincipal	Vehiculo	<i>Sí</i>	<i>Datos del vehículo implicado en la operación. Consulte VehiculoGetModel</i>

Consideraciones importantes:

- El instante compuesto por la fecha y la hora de finalización de la operación (FinFecha + FinHora) debe ser posterior al instante compuesto por la fecha y la hora de finalización de la operación (InicioFecha + InicioHora).
- En el caso del conductor, la aportación de los datos correspondientes al nombre y los apellidos de un conductor existente el sistema no implicará la modificación de los mismos. Esos datos son necesarios (según su obligatoriedad) cuando el responsable no existe previamente en el sistema.
- En el caso del responsable, la aportación de los datos correspondientes al nombre, los apellidos, la dirección de correo electrónico y el teléfono de un responsable existente el sistema no implicará la modificación de los mismos. Esos datos son necesarios (según su obligatoriedad) cuando el responsable no existe previamente en el sistema.

4.6. ResponsableModel

Nombre	Tipo	Obligatorio	Observaciones
Apellido1	String (50)	No/Sí	Es obligatorio cuando el responsable no existe en RECELIDE
Apellido2	String (50)	No/Sí	Es obligatorio cuando el responsable no existe en RECELIDE
CorreoE	String(255)	No/Sí	Es obligatorio cuando el responsable no existe en RECELIDE
Nif	String (9)	Sí	Debe pasar la validación de NIF de persona física
Nombre	String (50)	No/Sí	Es obligatorio cuando el responsable no existe en RECELIDE
Telefono	String (15)	No	Debe validar según la recomendación E.164 de la Unión Internacional de Telecomunicaciones

4.7. SoloDesinfeccionGetModel

Este modelo de datos define la información de una operación de Solo Desinfección:

Nombre	Tipo	Observaciones
AltaSistemaFechaHora	DateTime	Fecha de grabación de la operación en el sistema. El formato es 'aaaa-mm-dd hh:mm:ss'
Conductor	ConductorModel	Datos del conductor del vehículo implicado en la operación. Consulte la definición de ¡Error! No se encuentra el origen de la referencia.
FinFecha	Date	Fecha de finalización de la operación. El formato es 'aaaa-mm-dd'
FinHora	Time	Hora de finalización de la operación. El formato es 'hh:mm'
Identificador	String(20)	Identificador de la operación de limpieza y desinfección
InicioFecha	Date	Fecha de inicio de la operación. El formato es 'aaaa-mm-dd'
InicioHora	Time	Hora de inicio de la operación. El formato es 'hh:mm'
Matricula	String(20)	Matrícula del vehículo autodesinfectante. La matrícula validará según las validaciones de matrícula.
MetodoDesinfeccionId	Int	Identificador del método de desinfección, según los códigos de METODODESINFECTANTELIST /GET
MetodoDesinfeccionBiocida	String(255)	Nombre del biocida empleado. Es obligatorio en caso de que se haya aplicado un método de desinfección de tipo biocida. (Determinado por el atributo MetodoDesinfeccion)
OperacionSoloLimpieza	SoloLimpiezaModel	Operación de solo limpieza anterior a esta operación de solo desinfección. Consulte la definición de SoloLimpiezaGetModel

Responsable	ResponsableModel	<i>Datos del responsable que ha realizado la operación. Consulte la definición de ResponsableModel</i>
--------------------	-------------------------	--

4.8. SoloDesinfeccionPostModel

Este modelo de datos define la información que se debe aportar para añadir una nueva operación de Solo Desinfección:

Nombre	Tipo	Obligatorio	Observaciones
Conductor	ConductorModel	Sí	Datos del conductor del vehículo implicado en la operación. Consulte la definición de ¡Error! No se encuentra el origen de la referencia.
FinFecha	Date	Sí	Fecha de finalización de la operación. El formato es 'aaaa-mm-dd'
FinHora	Time	Sí	Hora de finalización de la operación. El formato es 'hh:mm'
Identificador	String(20)	Sí	Identificador de la operación de limpieza y desinfección
InicioFecha	Date	Sí	Fecha de inicio de la operación. El formato es 'aaaa-mm-dd'
InicioHora	Time	Sí	Hora de inicio de la operación. El formato es 'hh:mm'
Matricula	String(20)	Sí	Matrícula del vehículo autodesinfectante. La matrícula validará según las validaciones de matrícula.
MetodoDesinfeccionId	Int	Sí	Identificador del método de desinfección, según los códigos de METODODESINFECTANTELIST /GET
MetodoDesinfeccionBiocida	String(255)	No/Sí	Nombre del biocida empleado. Es obligatorio en caso de que se haya aplicado un método de desinfección de tipo biocida. (Determinado por el atributo MetodoDesinfeccion)

OperacionLimpiezaId	String(20)	<i>Sí</i>	<i>Identificador de la operación de solo limpieza anterior a esta operación de solo desinfección.</i>
Responsable	ResponsableModel	<i>Sí</i>	<i>Datos del responsable que ha realizado la operación. Consulte la definición de ResponsableModel</i>

4.9. SoloLimpiezaGetModel

Este modelo de datos define la información de una operación de Solo Limpieza:

Nombre	Tipo	Observaciones
AltaSistemaFechaHora	DateTime	Fecha de grabación de la operación en el sistema. El formato es 'aaaa-mm-dd hh:mm:ss'
CentroCodigo	String (50)	Código RECELIDE del centro en el que se ha realizado la operación de solo limpieza
Conductor	ConductorModel	Datos del conductor del vehículo implicado en la operación. Consulte la definición de ¡Error! No se encuentra el origen de la referencia.
FinFecha	Date	Fecha de finalización de la operación. El formato es 'aaaa-mm-dd'
FinHora	Time	Hora de finalización de la operación. El formato es 'hh:mm'
Identificador	String(20)	Identificador de la operación de limpieza y desinfección
InicioFecha	Date	Fecha de inicio de la operación. El formato es 'aaaa-mm-dd'
InicioHora	Time	Hora de inicio de la operación. El formato es 'hh:mm'
Matricula	String(20)	Matrícula del vehículo autodesinfectante.
Movimientos	Array de String(50)	Lista de los códigos REMO relacionados con la operación.
Precintos	Array de String(50)	Lista de precintos que se han empleado en la operación. Es obligatorio incluir al menos uno.
Responsable	ResponsableModel	Datos del responsable que ha realizado la operación. Consulte la definición de ResponsableModel
Vehiculo	VehiculoModel	Vehículo autodesinfectante sobre el que se ha realizado la operación.

4.10. SoloLimpiezaPostModel

Este modelo de datos define la información que se debe aportar para añadir una nueva operación de Solo Limpieza:

Nombre	Tipo	Obligatorio	Observaciones
CentroCodigo	String (50)	Sí	Código RECELIDE del centro en el que se ha realizado la operación de solo limpieza
Conductor	ConductorModel	Sí	Revisar la definición de ¡Error! No se encuentra el origen de la referencia.
FinFecha	Date	Sí	Fecha de finalización de la operación. El formato es 'aaaa-mm-dd'
FinHora	Time	Sí	Hora de finalización de la operación. El formato es 'hh:mm'
InicioFecha	Date	Sí	Fecha de inicio de la operación. El formato es 'aaaa-mm-dd'
InicioHora	Time	Sí	Hora de inicio de la operación. El formato es 'hh:mm'
Matricula	String(20)	Sí	Matrícula del vehículo autodesinfectante. La matrícula validará según las validaciones de matrícula.
Movimientos	Array de MovimientoModel	No	Revisar la definición de ¡Error! No se encuentra el origen de la referencia..
Precintos	Array de PrecintoModel	Sí	Es obligatorio incluir al menos un precinto. Revisar la definición de ¡Error! No se encuentra el origen de la referencia.
Responsable	ResponsableModel	Sí	Revisar la definición de ResponsableModel

Consideraciones importantes:

- El instante compuesto por la fecha y la hora de finalización de la operación (FinFecha + FinHora) debe ser posterior al instante compuesto por la fecha y la hora de finalización de la operación (InicioFecha + InicioHora).

- En el caso del conductor, la aportación de los datos correspondientes al nombre y los apellidos de un conductor existente el sistema no implicará la modificación de los mismos. Esos datos son necesarios (según su obligatoriedad) cuando el responsable no existe previamente en el sistema.
- En el caso del responsable, la aportación de los datos correspondientes al nombre, los apellidos, la dirección de correo electrónico y el teléfono de un responsable existente el sistema no implicará la modificación de los mismos. Esos datos son necesarios (según su obligatoriedad) cuando el responsable no existe previamente en el sistema.
- Como las operaciones de solo limpieza únicamente pueden llevarse a cabo en un vehículo autodesinfectante se comprobará que el vehículo está registrado como autodesinfectante por medio de su matrícula.

4.11. VehiculoGetModel

Nombre	Tipo	Observaciones
Identificacion	IdentificacionVehiculoModel	
TipoVehiculoId	Int	
TipoVehiculoNombre	String(50)	