



REGISTRO DE
CONTRATOS
ALIMENTARIOS

RCA API

Manual de integración para desarrolladores

Versión 0.44

Mayo 2025

Índice

1. Introducción.....	5
1.1. Propósito del documento	5
1.2. Audiencia del documento.....	5
1.3. Glosario	5
2. Visión general	7
2.1. Referencias	8
2.1.1. Aplicación web RCA	8
2.1.2. Real Decreto 1028/2022.....	10
3. ¿Cómo obtener las credenciales de acceso a RCA API?	11
3.1. Precondiciones.....	11
3.2. ¿Cómo obtenerlas en el entorno de pruebas ?	11
3.3. ¿Cómo obtenerlas en producción ?	15
3.4. ¿Y una vez firmado el acuerdo?.....	20
3.4.1. Consulta o incidente con la API	20
3.4.2. Planificación de pruebas de carga o estrés.....	20
4. Contrato de la API a alto nivel	21
4.1. URLs	21
4.2. RCA API.....	21
4.2.1. WADL de los servicios web REST.....	21
4.2.2. Obtener <i>token</i>	22
4.2.3. Nuevo contrato alimentario	22
4.2.4. Actualizar un contrato alimentario	22
4.2.5. Agregar productores primarios.....	23
4.2.6. Modificar productores primarios.....	23
5. Seguridad de la API	24
5.1. Protocolo seguro.....	24
5.2. Obtención de <i>token</i>	24
5.2.1. Ejemplo de obtención y uso de token en cliente JAVA.....	25
5.2.2. Ejemplo de obtención y uso de Token con Postman	26
6. Descripción de <i>endpoints</i> detallada.....	28

6.1.1.	Nuevo contrato.....	28
6.1.1.1.	Ejemplo de llamada a <i>nuevocontrato</i> (test Java)	30
6.1.1.2.	Ejemplo de llamada a <i>nuevocontrato</i> con Postman.....	30
6.1.1.3.	Respuesta éxito	32
6.1.1.4.	Respuesta error	32
6.1.2.	Actualización de un contrato.....	33
6.1.2.1.	Ejemplo de llamada a <i>actualizarcontrato</i> (test Java)	35
6.1.2.2.	Ejemplo de llamada a <i>actualizarcontrato</i> con Postman	35
6.1.2.3.	Respuesta éxito	37
6.1.2.4.	Respuesta error	37
6.1.3.	Agregar productores primarios.....	38
6.1.3.1.	Ejemplo de llamada a <i>agregarproductores</i> (test Java).....	40
6.1.3.2.	Ejemplo de llamada a <i>agregarproductores</i> con Postman	41
6.1.3.3.	Respuesta	44
6.1.4.	Modificar productores primarios.....	45
6.1.4.1.	Ejemplo de llamada a <i>modificarproductores</i> (test Java).....	47
6.1.4.2.	Ejemplo de llamada a <i>modificarproductores</i> con Postman	48
6.1.4.3.	Respuesta	51
7.	Roadmap.....	52
8.	Anexos	53
8.1.	Ejemplo de justificante de inscripción de nuevo contrato o actualización.....	53
8.2.	Errores.....	55
8.2.1.	Caracteres no permitidos	57

Control de versiones

VERSIÓN	FECHA	AUTOR	CAMBIOS RESPECTO DE LA VERSIÓN ANTERIOR
DRAFT	12/04/2023	Alejandro Gómez (AICA) Fco. Javier Granado (TTEC)	Borrador
0.1	25/05/2023	Alejandro Gómez (AICA)	<ul style="list-style-type: none"> ▶ Corrección de errores. ▶ Modificar nombre y contenido del capítulo 3. ¿Cómo obtener las credenciales de acceso a RCA API?
0.2	05/06/2023	Alejandro Gómez	<ul style="list-style-type: none"> ▶ Mejora en la provisión de las credenciales de acceso.
0.3	29/06/2023	M ^a José Menéndez (TTEC)	<ul style="list-style-type: none"> ▶ Corrección correo CAU.
0.4	13/12/2023	Fco. Javier Granado (TTEC)	<ul style="list-style-type: none"> ▶ Añadir métodos de alta y modificación de productores
0.41	26/12/2023	M ^a José Menéndez (TTEC)	<ul style="list-style-type: none"> ▶ Actualización web AICA
0.42	23/01/2024	TTEC	<ul style="list-style-type: none"> ▶ Aclaración juego caracteres métodos productores primarios
0.43	17/07/2024	TTEC	<ul style="list-style-type: none"> ▶ Actualización web Registro de Contratos
0.44	29/05/2025	TTEC	<ul style="list-style-type: none"> ▶ Referencia versión TLS ▶ Acceso producción por web AICA

1. Introducción

1.1. Propósito del documento

El propósito del presente documento es detallar la API de RCA diseñada para lograr el intercambio de información entre aquellas aplicaciones o soluciones clientes destinadas a consumir los servicios publicados en este sistema.

1.2. Audiencia del documento

El presente documento está dirigido a integradores de sistemas y áreas TIC de grandes operadores de la cadena alimentaria.

1.3. Glosario

ACRÓNIMO/ABREVIATURA	DEFINICIÓN
AICA	Agencia de Información y Control Alimentarios, O.A.
Cliente o consumidor de la API de servicios	Es un trozo de código fuente compilado o interpretado capaz de realizar una llamada a la API de servicios que publica RCA.
<i>Endpoint</i>	Un endpoint o “punto final” de un API es el lugar donde un cliente realiza las solicitudes de un servicio a otra aplicación, en este caso RCA (conocidas como llamadas API).
<i>Files</i>	Documentos asociados al contrato.
JSON	JSON (acrónimo de JavaScript Object Notation , 'notación de objeto de JavaScript') es un formato de texto sencillo para el intercambio de datos. Se trata de un subconjunto de la notación literal de objetos de JavaScript , aunque, debido a su amplia adopción como alternativa a XML , se considera un formato independiente del lenguaje.
<i>JSON Schema</i>	Un JSON Schema es un JSON que describe la estructura que tendrá otro JSON, pudiendo definir qué información puede contener, con qué tipos de datos, que datos son obligatorios, valores admitidos, etc.
<i>Metadata</i>	Datos asociados al comprador y al productor primario.
<i>Multipart/form-data</i>	Es un tipo de codificación que permite enviar archivos a través de un método POST. Este tipo de contenido permite dividir varios tipos de datos y procesar estos datos de manera diferente en una misma llamada.

ACRÓNIMO/ABREVIATURA	DEFINICIÓN
RCA	Registro de Contratos Alimentarios.
WADL	Acrónimo en inglés de “Web Application Description Language”

2. Visión general

Identificación

RCA API

Título

API del Registro de Contratos Alimentarios (RCA, en adelante).

¿Qué es RCA?

RCA es una solución informática creada por la **Agencia de Información y Control Alimentarios, O.A.** (en adelante, AICA) y regulado por el Real Decreto 1028/2022, del 20 de diciembre, por el que se desarrolla el Registro de Contratos Alimentarios.

Como solución de la Administración Pública, su misión es proporcionar los canales o medios electrónicos necesarios a los operadores de la cadena alimentaria que estén obligados por este R.D. a inscribir los contratos alimentarios (y sus modificaciones).

Uno de esos canales es RCA API.

¿Qué es RCA API?

RCA API es una interfaz programática para los servicios implementados en la solución RCA. Es una parte clave de RCA, pues permite a terceros e integradores agregar dichos servicios a sus propias aplicaciones o soluciones empresariales.

NOTA: El lanzamiento de RCA API (v1.0) en producción se realizó la **última semana de junio de 2023** con los “endpoints” que permiten al integrador automatizar la inscripción de nuevos contratos alimentarios y su actualización.

El lanzamiento de RCA API (v1.2) en producción está programado para las **últimas semanas de diciembre de 2023** con los “endpoints” que permiten al integrador automatizar el alta de nuevos productores primarios y su actualización.

[Ver Roadmap.](#)

¿Cuál es su objetivo?

El objetivo principal de RCA API es proporcionar un canal electrónico **seguro, robusto y eficaz** para aquellos operadores de la cadena (compradores) que deseen integrar los servicios expuestos por el sistema RCA.

¿Cómo funciona?

RCA API se expone como un conjunto de servicios REST a los clientes a través de Internet y proporciona un interfaz JSON para el intercambio de datos. Los clientes pueden enviar solicitudes HTTPs a dichos servicios (previa [obtención de credenciales de acceso](#)), cumpliendo las especificaciones técnicas que en este documento se detallan en materia de seguridad, protocolos de comunicación y estructuras de datos requeridas.

NOTA: *Es responsabilidad del cliente o consumidor de la API de servicios la adaptación de sus aplicaciones o soluciones informáticas para llevar a cabo la integración con RCA. Queda fuera del alcance de AICA, cualquier desarrollo o adaptación en la parte cliente.*

2.1. Referencias

2.1.1. Aplicación web RCA

El **22 de diciembre de 2022**, AICA lanzó a producción la aplicación web RCA como primer canal destinado a los operadores y que se encuentra accesible en el portal de Internet del organismo:

[AICA: Agencia de Información y Control Alimentarios](#)

Registro de contratos Alimentarios

La inscripción de los contratos alimentarios formalizados con productores primarios y agrupaciones de productores primarios es obligatoria desde el 30 de junio de 2023.

Descubre cómo



Control de la Ley de la Cadena Alimentaria Más Información

Registro de Contratos Alimentarios Más Información

¿Cómo denunciar un incumplimiento? Más Información

Informes de actividad de AICA Más Información

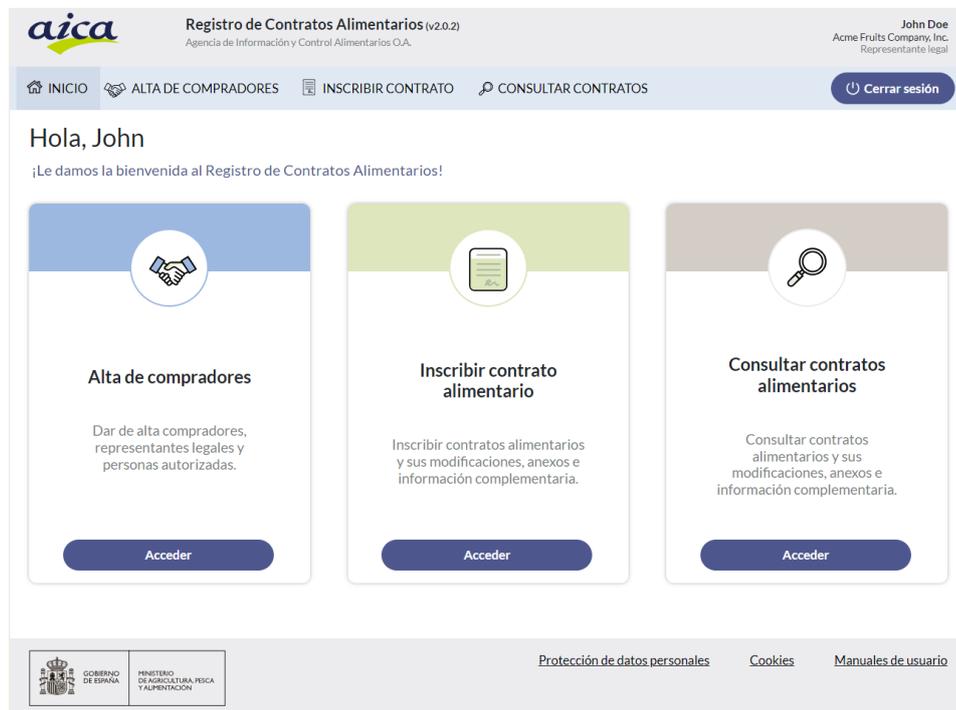
Registro de Contratos Alimentarios

Acceder

Preguntas frecuentes

API de servicios RCA

Desde el **31 de enero de 2023**, fecha en la que entró en vigor el R.D. 1028/2022, el usuario de la aplicación web tiene disponibles los servicios de “Alta de compradores”, “Inscribir contrato alimentario” y “Consultar contratos alimentarios”, a los cuales puede acceder en función de su perfil de seguridad.



En la primera versión de la API de RCA (ver [Roadmap](#)), se exponen públicamente los endpoints de inscribir un nuevo contrato alimentario y actualizar un contrato existente para permitir a los equipos integradores automatizar el mismo proceso que el usuario final de la aplicación web hace de manera “uno a uno”.

En la segunda versión de la API de RCA (ver [Roadmap](#)), se exponen públicamente los endpoints de agregar nuevos productores primarios y actualizar los datos del para permitir a los equipos integradores automatizar el mismo proceso que el usuario final de la aplicación web.

NOTA: En el pie de la aplicación web RCA se muestra un enlace para descargar los manuales de usuario y que quizás puedan servir para ayudar a entender el sistema antes de proceder a la integración.

2.1.2. Real Decreto 1028/2022

El **21 de diciembre de 2022** se publicó en el BOE el Real Decreto 1028/2022 por el que se desarrolla el Registro de Contratos Alimentarios.

Para más información, ir al [Real Decreto](#).

3. ¿Cómo obtener las credenciales de acceso a RCA API?

Para obtener las credenciales de acceso a la API de RCA, es necesario **firmar el acuerdo de uso adecuado y responsable de la misma** en los entornos disponibles (*pruebas y producción*).

3.1. Precondiciones

El comprador debe haber solicitado **la petición de acceso** a la API de servicios de RCA mediante el formulario destinado a ello y haber recibido una respuesta afirmativa por parte de AICA.

3.2. ¿Cómo obtenerlas en el **entorno de pruebas**?

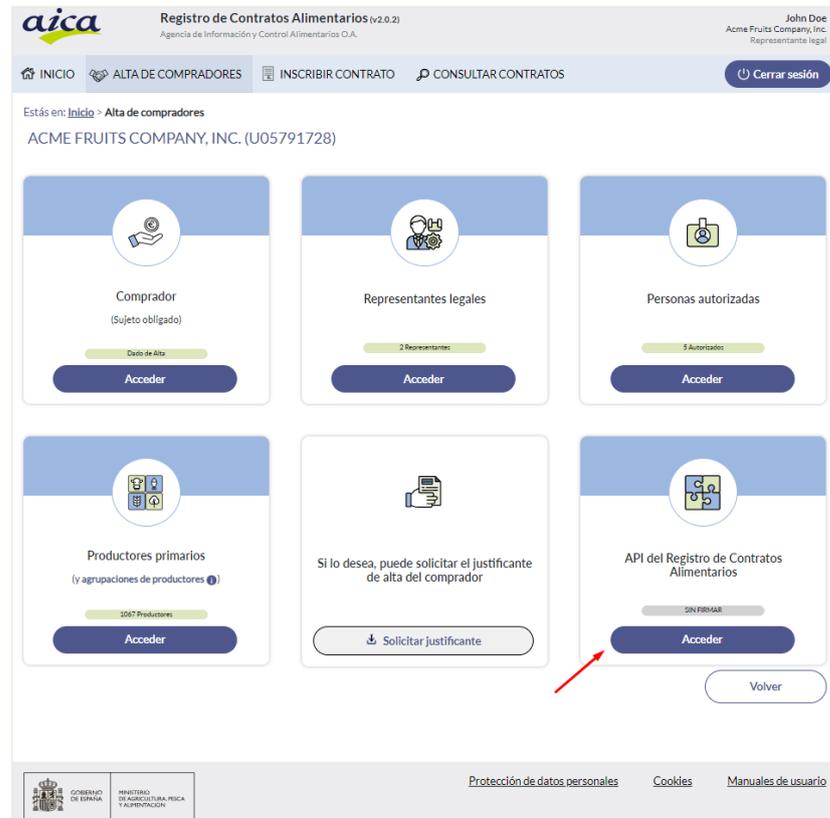
NOTA: RCA proporciona un **entorno de pruebas** que facilitará la integración antes de pasar a producción o hacer cambios sin usar datos reales.

Antes de enumerar cómo obtener las credenciales de acceso en el entorno de pruebas, si el comprador no está dado de alta en dicho entorno, debe hacerlo en la aplicación web de RCA desplegada en: <https://integra-servicio.mapa.gob.es/regcontratosaic>

Una vez dado de alta, se han de dar los siguientes pasos:

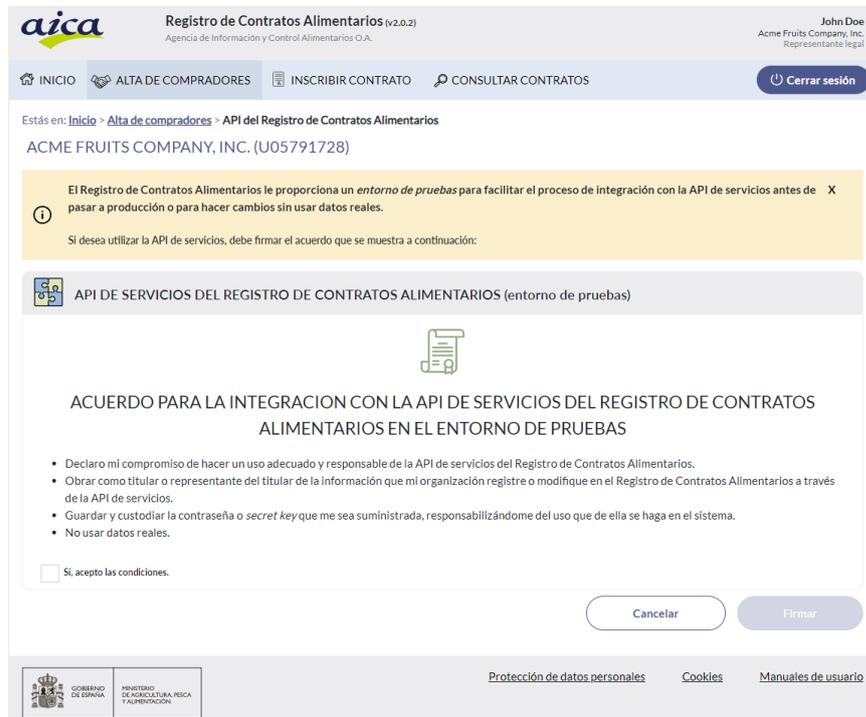
1. El representante legal del comprador debe acceder a la aplicación web de RCA desplegada en el *entorno de pruebas* haciendo uso de su certificado electrónico de representación e ir a la tarjeta "**Alta de compradores**".

A continuación, debe pulsar sobre el botón "Acceder" de la tarjeta "**API del Registro de Contrato Alimentarios**", la cual le dará acceso al proceso de firma electrónica del acuerdo de uso responsable de la API de servicios:



NOTA: Hasta que el acuerdo de uso no sea firmado, el estado de la tarjeta “API del Registro de Contratos Alimentarios” será “SIN FIRMAR”. Una vez firmado, el estado será “FIRMADO”.

2. Seguidamente el representante debe leer detenidamente el acuerdo, aceptar las condiciones y firmarlo electrónicamente:



Para activar el botón “Firmar”, solo tiene que marcar la casilla de verificación “Sí, acepto las condiciones”.

NOTA: Al tratarse de un **entorno de pruebas**, es importante cumplir con el cuarto punto y **no usar datos reales**, acorde a las directrices marcadas por el RGPD y el ENS.

3. Al pulsar en el botón “Firmar”, se muestra una ventana emergente para confirmar el proceso de firma electrónica:



4. Si quiere seguir con el proceso de firma, el representante solo tiene que pulsar sobre el botón “Firmar”.

El cliente de firma utilizado por RCA es **Autofirma**. Mediante este cliente de escritorio, el representante legal puede firmar el justificante con el mismo certificado electrónico con el cual ha accedido a la aplicación web.

El proceso termina con la confirmación de la firma del acuerdo en el **entorno de pruebas**, donde el representante legal puede descargar el justificante firmado electrónicamente:

The screenshot displays the 'API DE SERVICIOS DEL REGISTRO DE CONTRATOS ALIMENTARIOS (entorno de pruebas)' page. It features a green confirmation banner stating 'El acuerdo ya está firmado.' Below this, there is a section for 'CREDENCIALES DE ACCESO' with fields for 'USUARIO:' (containing 'DBDNS1') and 'CONTRASEÑA:' (containing 'IBK#WB#1U0Hqorf3'). A central area contains the title 'ACUERDO PARA LA INTEGRACION CON LA API DE SERVICIOS DEL REGISTRO DE CONTRATOS ALIMENTARIOS EN EL ENTORNO DE PRUEBAS' and a list of terms and conditions. At the bottom, there is a 'Firmado por: John Doe - 99999999R' and 'Fecha y hora: 17/07/2024 08:19'. Two buttons are visible: 'Descargar justificante' and 'Volver'.

Las credenciales de acceso se muestran en pantalla (el usuario y la contraseña serán generados por el sistema). La contraseña se muestra oculta con asteriscos por motivos de seguridad (solo debe pulsar sobre el icono  para mostrarla o  para copiarla en el portapapeles).

En cualquier momento, el representante legal del comprador tendrá la posibilidad de volver a entrar en la tarjeta “API del Registro de Contratos Alimentarios” para descargar el justificante o ver las credenciales de acceso de nuevo.

A continuación, se muestra un ejemplo de un justificante de la firma del acuerdo:

The image shows a digital document titled "Registro de Contratos Alimentarios" with the subtitle "ACUERDO PARA LA INTEGRACIÓN CON LA API DE SERVICIOS DEL REGISTRO DE CONTRATOS ALIMENTARIOS EN EL ENTORNO DE PRUEBAS". The document is signed by JOHN DOE, representing Acme Fruits Company, Inc. It includes a list of terms and conditions, a digital signature box from the Ministry of Agriculture, Fisheries and Food, and a timestamp of 17/07/2024 09:31:00.

FIRMADO POR:	EN REPRESENTACIÓN DE:
Núm. de doc. identificativo: ***9999**	Núm. de doc. identificativo: U05791728
Nombre y apellidos: JOHN DOE	Razón social: Acme Fruits Company, Inc.
Rol en el comprador: Representante legal	

- Declaro mi compromiso de hacer un uso adecuado y responsable de la API de servicios del Registro de Contratos Alimentarios.
- Obrar como titular o representante del titular de la información que mi organización registre o modifique en el Registro de Contratos Alimentarios a través de la API de servicios.
- Guardar y custodiar la contraseña o *secret key* que me sea suministrada, responsabilizándome del uso que de ella se haga en el sistema.
- No usar datos reales.

MINISTERIO DE AGRICULTURA, PESCA Y ALIMENTACIÓN
AGENCIA DE INFORMACIÓN Y CONTROL ALIMENTARIOS, O.A.
17/07/2024 09:31:00

Firmado por JOHN DOE - ***9999** el día 17/07/2024 con un certificado emitido por AC FNMT Usuarios

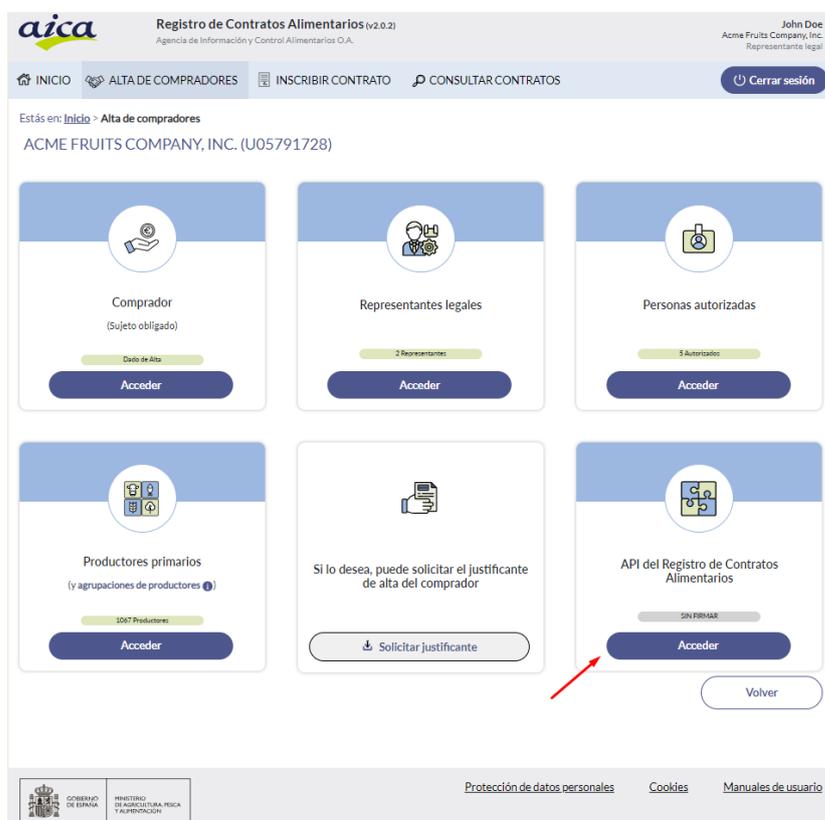
3.3. ¿Cómo obtenerlas en **producción**?

Antes de enumerar cómo obtener las credenciales de acceso en **producción**, si el comprador no está dado de alta en dicho entorno, debe hacerlo en la aplicación web de RCA desplegada en: [AICA: Agencia de Información y Control Alimentarios](#).

Una vez dado de alta, se han de dar los siguientes pasos:

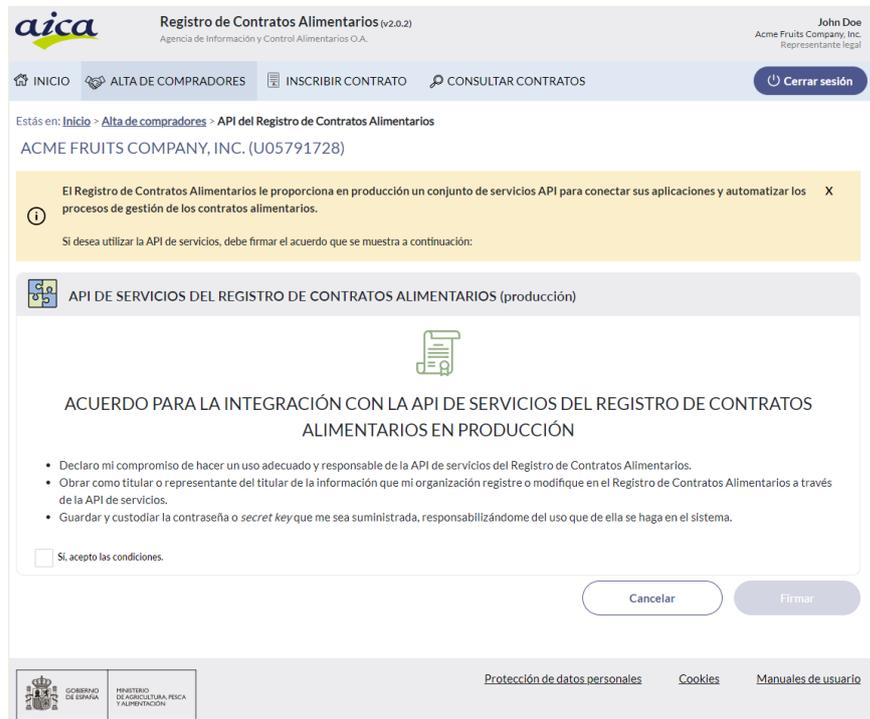
1. El representante legal del comprador debe acceder a la aplicación web de RCA desplegada en **producción**, haciendo uso de su certificado electrónico de representación e ir a la tarjeta **“Alta de compradores”**.

A continuación, debe pulsar sobre el botón **“Acceder”** de la tarjeta **“API del Registro de Contrato Alimentarios”**, la cual le dará acceso al proceso de firma electrónica del acuerdo de uso responsable de la API de servicios:



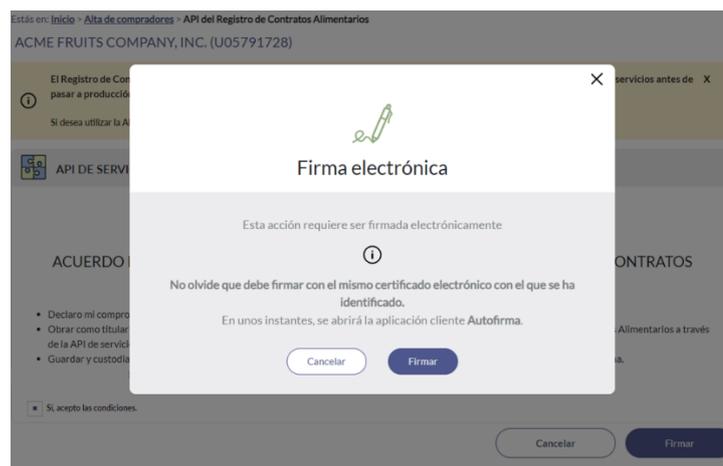
NOTA: *Hasta que el acuerdo de uso no sea firmado, el estado de la tarjeta “API del Registro de Contratos Alimentarios” será “SIN FIRMAR”. Una vez firmado, el estado será “FIRMADO”.*

2. Seguidamente el representante debe leer detenidamente el acuerdo, aceptar las condiciones y firmarlo electrónicamente:



Para activar el botón “Firmar”, solo tiene que marcar la casilla de verificación “Sí, acepto las condiciones”.

3. Al pulsar en el botón “Firmar”, se muestra una ventana emergente para confirmar el proceso de firma electrónica:



4. Si quiere seguir con el proceso de firma, el representante solo tiene que pulsar sobre el botón “Firmar”.

El cliente de firma utilizado por RCA es **Autofirma**. Mediante este cliente de escritorio, el representante legal puede firmar el justificante con el mismo certificado electrónico con el cual ha accedido a la aplicación web.

El proceso termina con la confirmación de la firma del acuerdo en **producción**, donde el representante legal puede descargar el justificante firmado electrónicamente:

The screenshot displays the 'API DE SERVICIOS DEL REGISTRO DE CONTRATOS ALIMENTARIOS (producción)' page. It features a green confirmation banner stating 'El acuerdo ya está firmado.' and provides instructions on using the API. Below this is a 'CREDENCIALES DE ACCESO' section with fields for 'USUARIO:' (containing 'D8DNS1') and 'CONTRASEÑA:' (masked with asterisks). A central area contains the title 'ACUERDO PARA LA INTEGRACION CON LA API DE SERVICIOS DEL REGISTRO DE CONTRATOS ALIMENTARIOS EN PRODUCCIÓN' and a list of terms and conditions. At the bottom, it shows the signature 'Firmado por: John Doe - 99999999R' and the timestamp 'Fecha y hora: 17/07/2024 09:31'. Two buttons are present: 'Descargar justificante' and 'Volver'. The footer includes the government logo and links for 'Protección de datos personales', 'Cookies', and 'Manuales de usuario'.

NOTA: La contraseña de acceso a la API de servicios en **producción** es distinta al entorno de pruebas.

Las credenciales de acceso se muestran en pantalla (el usuario y la contraseña serán generados por el sistema). La contraseña se muestra oculta con asteriscos por motivos de

seguridad (solo debe pulsar sobre el icono  para mostrarla o  para copiarla en el portapapeles).

En cualquier momento, el representante legal del comprador tendrá la posibilidad de volver a entrar en la tarjeta “API del Registro de Contratos Alimentarios” para descargar el justificante o ver las credenciales de acceso de nuevo.

A continuación, se muestra un ejemplo de un justificante de la firma del acuerdo:

 GOBIERNO DE ESPAÑA
MINISTERIO DE AGRICULTURA, PESCA Y ALIMENTACIÓN

AGENCIA DE INFORMACIÓN Y CONTROL ALIMENTARIOS, O.A.

Registro de Contratos Alimentarios

ACUERDO PARA LA INTEGRACIÓN CON LA API DE SERVICIOS DEL REGISTRO DE CONTRATOS ALIMENTARIOS EN PRODUCCIÓN

FIRMADO POR:	EN REPRESENTACIÓN DE:
Núm. de doc. identificativo: ***9999** <small>DNI o NIE</small>	Núm. de doc. identificativo: U05791728 <small>NIF, DNI o NIE</small>
Nombre y apellidos: JOHN DOE	Razón social: ACME FRUITS COMPANY,
Rol en el comprador: Representante legal	

- Declaro mi compromiso de hacer un uso adecuado y responsable de la API de servicios del Registro de Contratos Alimentarios.
- Obrar como titular o representante del titular de la información que mi organización registre o modifique en el Registro de Contratos Alimentarios a través de la API de servicios.
- Guardar y custodiar la contraseña o `secret key` que me sea suministrada, responsabilizándome del uso que de ella se haga en el sistema.

MINISTERIO DE AGRICULTURA, PESCA Y ALIMENTACIÓN
AGENCIA DE INFORMACIÓN Y CONTROL ALIMENTARIOS, O.A.
17/07/2024 09:31:00

Firmado por JOHN DOE - ***9999** el día 17/07/2024 con un certificado emitido por AC FNMT Usuarios

3.4. ¿Y una vez firmado el acuerdo?

Una vez firmado el acuerdo (*sea en el entorno de pruebas o producción*), el comprador dispondrá de las credenciales de usuario necesarias para acceder a la API de servicios de RCA, cuyas URL se proporcionan en el [capítulo 4](#).

A continuación, se enumeran las solicitudes que el comprador puede enviar al CAU de RCA para realizar la integración:

3.4.1. Consulta o incidente con la API

El equipo de integración del comprador puede consultar o reportar un incidente con la API, enviando un mensaje de correo electrónico a cau.rca@tragsa.es cuyo asunto debe ser “RCA API | Consulta o incidente | %Razón social del comprador%” donde debe indicar como mínimo los siguientes datos en el cuerpo del mensaje:

- ▶ Número de documento identificativo del comprador.
- ▶ Entorno donde ocurre el incidente o se requiere la consulta:
 - a. Pruebas.
 - b. Producción.
- ▶ Descripción del incidente o la consulta.
- ▶ En caso de un incidente, indicar fecha y hora de la ocurrencia de este.
- ▶ En caso de un incidente, indicar el código de error devuelto, pantallazo o similar.

El CAU responderá mediante el mismo canal a la consulta o incidente.

3.4.2. Planificación de pruebas de carga o estrés

Solo en el entorno de pruebas.

El equipo de integración del comprador puede solicitar la planificación de pruebas de carga o estrés sobre la API en el entorno de **pruebas**, enviando un mensaje de correo electrónico a cau.rca@tragsa.es cuyo asunto debe ser “RCA API | Pruebas de carga | %Razón social del comprador%” donde debe indicar como mínimo los siguientes datos en el cuerpo del mensaje:

- ▶ Número de documento identificativo del comprador.
- ▶ Descripción de las pruebas que se desean hacer.
- ▶ Volumen de transacciones por minuto que se desean lanzar.
- ▶ Enumeración de fechas y horas candidatas para realizarlas.

El CAU responderá por el mismo canal con la fecha y hora planificada.

4. Contrato de la API a alto nivel

En este apartado se especifican los métodos públicos disponibles a través de la API RCA.

4.1. URLs

Las URLs de RCA API son:

ENTORNO	URL
INTEGRACIÓN	https://integra-servicio.mapa.gob.es/wsregcontratosaica/
PRODUCCIÓN	https://servicio.mapa.gob.es/wsregcontratosaica/

4.2. RCA API

RCA API es una interfaz de servicios REST que permite a sus clientes (compradores) realizar la inscripción de contratos alimentarios con sus productores primarios.

Para poder consumir cualquiera de los recursos, previamente se debe obtener un “Access Token” que autorice su consumo, tal y como se describe en el capítulo [5. Seguridad de la API](#).

La API ofrece los siguientes recursos:

- ▶ WADL - Web Application Description Language.
- ▶ Obtener token.
- ▶ Inscripción de un nuevo contrato alimentario.
- ▶ Actualizar un contrato alimentario.

4.2.1. WADL de los servicios web REST

El WADL modela los recursos provistos por un servicio y las relaciones entre ellos.

La especificación completa del servicio web y sus métodos se encuentra accesible a través de la siguiente URL:

<https://integra-servicio.mapa.gob.es/wsregcontratosaica/servicioweb/application.wadl>

NOTA: Esta URL también está protegida por un token de seguridad.

4.2.2. Obtener *token*

Para obtener un *token* de un usuario autorizado, el método del servicio es:

URI: <https://integra-servicio.mapa.gob.es/wsregcontratosaic/oauth/token>

Método: POST

Toda petición enviada desde un cliente a cualquiera de los métodos disponibles en el servicio web deberá incluir en la cabecera un *token* válido. De lo contrario, la petición será rechazada por el servicio.

La obtención del *token* se realiza a través de un método incluido en el catálogo de métodos públicos del servicio, y requiere credenciales de usuario y contraseña (obtenidas tal y como se indica en el [capítulo 3](#)). Por tanto, sólo una aplicación cliente autorizada con credenciales válidas podrá obtener el *Token* de seguridad que permite a su vez utilizar el resto de los métodos del servicio.

4.2.3. Nuevo contrato alimentario

Para inscribir un nuevo contrato, el método del servicio es:

URI: <https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/nuevocontrato>

Método: POST

Este método está preparado para recibir una petición de tipo multipart/form-data, en el que una parte del contenido (los documentos asociados al contrato) se transportan en un array de bytes y otra parte (los metadatos del contrato) se transportan en formato JSON.

El método devuelve una respuesta en formato JSON con el número de contrato (identifica unívocamente al contrato y todos sus documentos inscritos), un mensaje de confirmación y el justificante de la inscripción de este codificado en Base64.

4.2.4. Actualizar un contrato alimentario

Para actualizar un contrato ya existente, el método del servicio es:

URI: <https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/actualizarcontrato>

Método: POST

Este método está preparado para recibir un número de contrato junto a una petición de tipo multipart/form-data, en el que una parte del contenido (los documentos asociados al contrato) se transportan en un array de bytes y otra parte (los metadatos del contrato) se transportan en formato JSON.

El método devuelve una respuesta en formato JSON con el número de contrato, un mensaje de confirmación y el justificante de la inscripción de este codificado en Base64.

4.2.5. Agregar productores primarios

Para agregar nuevos productores primarios, el método del servicio es:

URI: <https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/agregarproductores>

Método: POST

Este método está preparado para recibir uno o varios productores primarios en formato JSON.

El método devuelve una respuesta en formato JSON con el resultado de cada una de las altas de los productores primarios enviados, con un "OK" para cada uno de los productores agregados correctamente y con la descripción de los errores en caso de los productores que no se hayan podido agregar.

4.2.6. Modificar productores primarios

Para modificar productores primarios, el método del servicio es:

URI: <https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/modificarproductores>

Método: POST

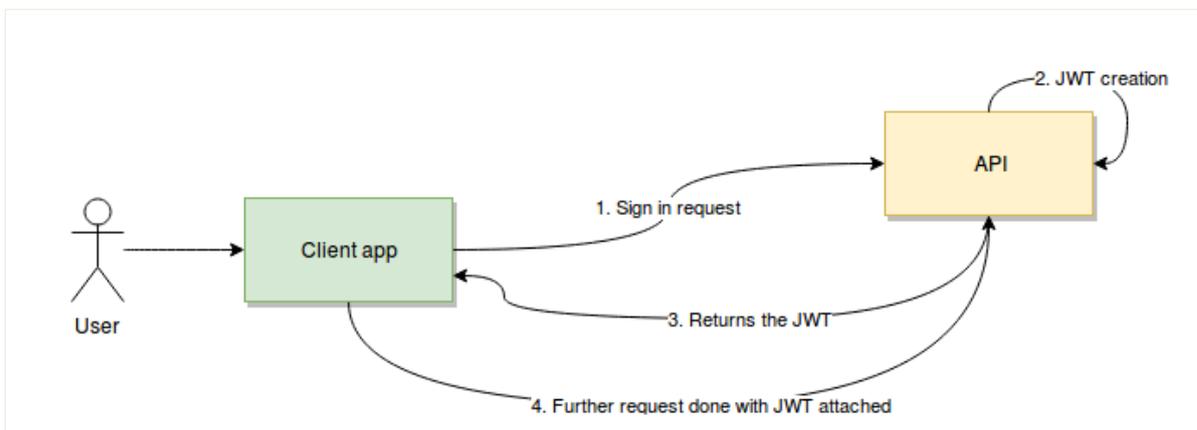
Este método está preparado para recibir uno o varios productores primarios en formato JSON.

El método devuelve una respuesta en formato JSON con el resultado de cada una de las modificaciones de los productores primarios enviados, con un "OK" para cada uno de los productores modificados correctamente y con la descripción de los errores en caso de los productores que no se hayan podido modificar.

5. Seguridad de la API

La API de RCA se ha desarrollado como un conjunto de servicios REST con autorización **OAuth2** (*Token Bearer*).

Toda petición enviada desde un cliente a cualquiera de los recursos disponibles en la API deberá incluir en la cabecera un **token válido**. De lo contrario, la petición será rechazada por el *endpoint*.



Hay que destacar que la contraseña del usuario que llama al método para obtener dicho *token* se genera internamente en el sistema acorde a una política muy restrictiva de seguridad. Una vez generada se almacena encriptada.

NOTA: *Un token solo será válido durante un tiempo limitado, concretamente durante 1.800 segundos. Si éste caduca, el cliente de la API deberá solicitar uno nuevo.*

5.1. Protocolo seguro

RCA API utiliza **TLS** (Transport Layer Security) el cual protege la información que viaja entre el cliente y la propia API, cifrando el mensaje y haciendo uso del protocolo HTTPS.

NOTA: *La versión, mínima, aceptada del protocolo TLS será la v1.2.*

5.2. Obtención de *token*

El *token* de seguridad es requisito indispensable para el consumo de cualquiera de los otros métodos disponibles en el servicio. El *token* se solicitará mediante el método público y disponible en la siguiente URL:

URI: <https://integra-servicio.mapa.gob.es/wsregcontratosaic/oauth/token>

Método: POST

Este método recibe los siguientes parámetros de entrada:

NOMBRE	VALOR	¿OBLIGATORIO?
<i>grant_type</i>	client_credentials	S
<i>client_id</i>	%Usuario del comprador%	S
<i>client_secret</i>	%Contraseña del comprador%	S

El método devuelve una estructura JSON con los siguientes elementos:

RESPUESTA EXITO	VALOR
<i>access_token</i>	Token generado por el servicio, válido para utilizar en otras llamadas a métodos del servicio mientras no expire su validez.
<i>token_type</i>	<i>bearer</i>
<i>expires_in</i>	Tiempo de validez del token (expresado en segundos).
<i>scope</i>	<i>all</i>

RESPUESTA ERROR	VALOR
<i>error</i>	<i>Invalid_client</i>
<i>error_description</i>	Descripción del error.

5.2.1. Ejemplo de obtención y uso de token en cliente JAVA

A continuación, se muestra un ejemplo de implementación en código java para obtener el *token* de seguridad.

```

CloseableHttpClient client = HttpClients.custom(). setSSLContextFactory (new SSLConnectionSocketFactory
(SSLContexts.custom(). loadTrustMaterial (null, new TrustSelfSignedStrategy ()). build(),
NoopHostnameVerifier.INSTANCE)).build();

String tokenEndpoint = "https://integra-servicio.mapa.gob.es/wsregcontratosaic/oauth/token";

System.out.println("Getting an access token...");

HttpPost tokenRequest = new HttpPost(tokenEndpoint);

List<NameValuePair> data = new ArrayList<NameValuePair>();
  
```

```

data.add(new BasicNameValuePair("grant_type", "client_credentials"));
data.add(new BasicNameValuePair("client_id", "%Usuario del comprador%"));
data.add(new BasicNameValuePair("client_secret", "%Contraseña del comprador%"));
tokenRequest.setEntity(new UrlEncodedFormEntity(data));
CloseableHttpResponse tokenResponse = client.execute(tokenRequest);
JsonObject tokenObject = Json.createReader(tokenResponse.getEntity().getContent()).readObject();
System.out.printf("Result: %s\n", tokenObject.toString());
String accessToken = tokenObject.getString("access_token");
System.out.printf("Access token: %s\n", accessToken);

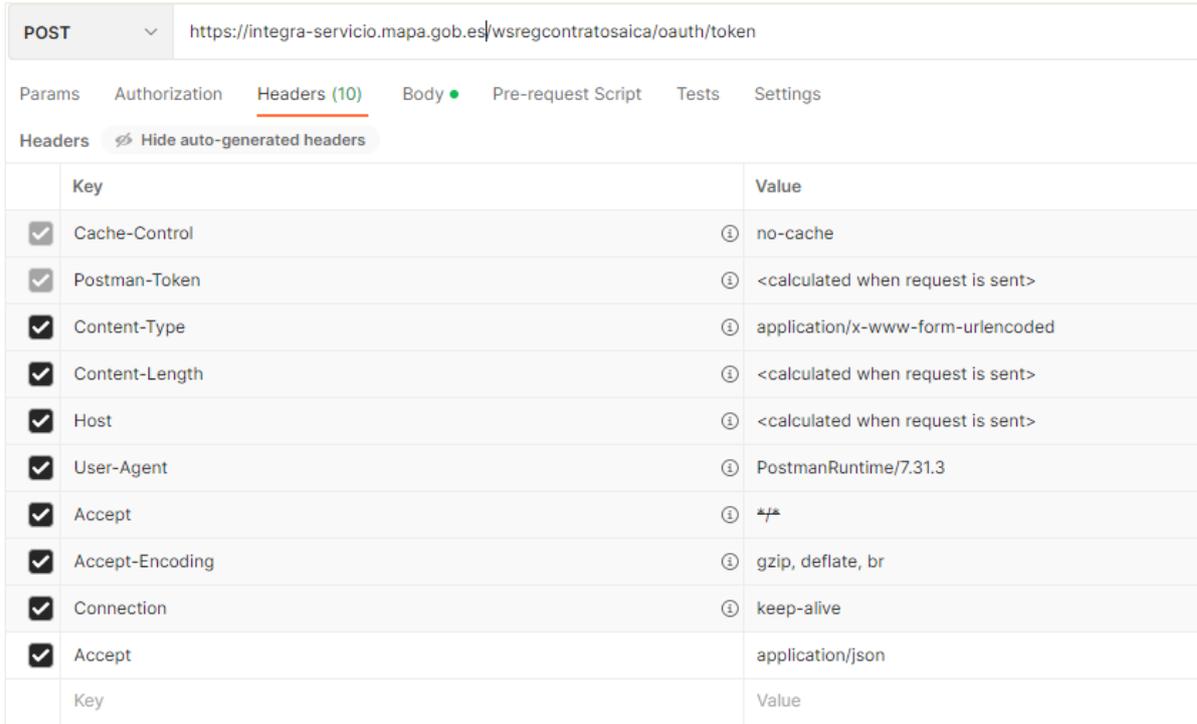
```

5.2.2. Ejemplo de obtención y uso de Token con Postman

La configuración de dicho método en Postman se realiza de la siguiente manera:

- ▶ Seleccionar como tipo de método la opción POST.
- ▶ Indicar en la URL la dirección:

<https://integra-servicio.mapa.gob.es/wsregcontratosaiica/oauth/token>



Configuramos la petición de *token* desde la pestaña *Body* seleccionando contenido tipo *x-www-form-urlencoded* y definiendo los tres parámetros que se muestran en pantalla.

NOTA: *client_secret* y *client_id* serán diferentes, dependiendo del usuario que solicite el token.

POST ▼ | <https://integra-servicio.mapa.gob.es/wsregcontratosaioca/oauth/token>

Params Authorization Headers (10) **Body** ● Pre-request Script Tests Settings

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

	Key	Value
<input checked="" type="checkbox"/>	client_id	QK9CS7
<input checked="" type="checkbox"/>	client_secret	pass
<input checked="" type="checkbox"/>	grant_type	client_credentials

Una vez configurada esta llamada al método de petición de *token*, pulsando en el botón SEND que está a la derecha de la URL, se enviará la petición al servicio.

Si la configuración era correcta, el *endpoint* nos devolverá un *token* como respuesta.

Body Cookies Headers (12) Test Results 200 200 · 2.44 s · 633 B · Save Response

```

1  {}
2  {
3    "access_token": "eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXLTJ0eS9ndQx0dXN0eS1mV4cCI6MTc0NDU4Njk1MjY0LmFkZCz7SjYXoJkVWWh5ZGdldk4P7Hvth023ciGeo",
4    "token_type": "Bearer",
5    "expires_in": "1000",
6    "scope": "des",
7    "jti": ""
  }
  
```

6. Descripción de *endpoints* detallada

En los siguientes apartados se describe en detalle cada uno de los métodos públicos de la API de servicios de RCA.

6.1.1. Nuevo contrato

Para inscribir un nuevo contrato, el método del servicio es:

URL: <https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/nuevocontrato>

Método: POST

NOTA: Este método está preparado para recibir un *multipart/form-data* en el que una parte del contenido (los documentos) venga en un array de bytes y otra parte (los datos del comprador y productor) venga en un JSON.

Parámetros de cabecera:

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
<i>authorization</i>	Texto (216)	<i>Bearer + token</i> de seguridad obtenido para el usuario. Ver apartado 5.2.	S

A continuación, se describen los **parámetros de entrada:**

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
<i>company_id</i>	Texto (20)	Parámetro del JSON Metadata. %NIF, DNI o NIE del comprador% que corresponde al comprador asignado al contrato que se pretende dar de alta.	S
<i>producer_id</i>	Texto (20)	Parámetro del JSON Metadata. %NIF, DNI o NIE del productor% que corresponde al productor con el que el comprador tiene el contrato que se pretende dar de alta.	S
<i>files</i>	<i>File/bytes(5)</i>	Documento/s que se anexan al contrato. Los documentos no podrán ser más de 5	S

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
		documentos en formato PDF plano ¹ . Los documentos no podrán pesar más de 5MB la unidad.	

A continuación, se describen los **parámetros de salida**:

El método devuelve una estructura JSON con los siguientes elementos:

RESPUESTA EXITO	VALOR
<i>status</i>	Resultado de la petición. "OK"
<i>contract_id</i>	Identificación del contrato por el que se podrá consultar en la aplicación web RCA.
<i>receipt</i>	Justificante de la inscripción del contrato en Base64. Este justificante se puede obtener en formato Pdf desde la aplicación RCA.

RESPUESTA ERROR	VALOR
<i>status</i>	Resultado de la petición. "ERROR"
<i>error_code</i>	Código de error devuelto por el API.
<i>error_description</i>	Descripción del mismo cuando falla la petición de inscripción de un nuevo contrato alimentario.

Validaciones:

El método valida los siguientes puntos:

- ▶ Los datos relativos del comprador y productor existan en el sistema RCA.
- ▶ El número de documentos que anexos al contrato no puede superar el límite establecido.
- ▶ El formato de todos los documentos sea el adecuado. Inicialmente pdf.
- ▶ El tamaño de cada documento no puede exceder de un tamaño establecido.

¹ El PDF plano es un PDF de salida sin elementos interactivos, como campos de formulario.

6.1.1.1. Ejemplo de llamada a *nuevocontrato* (test Java)

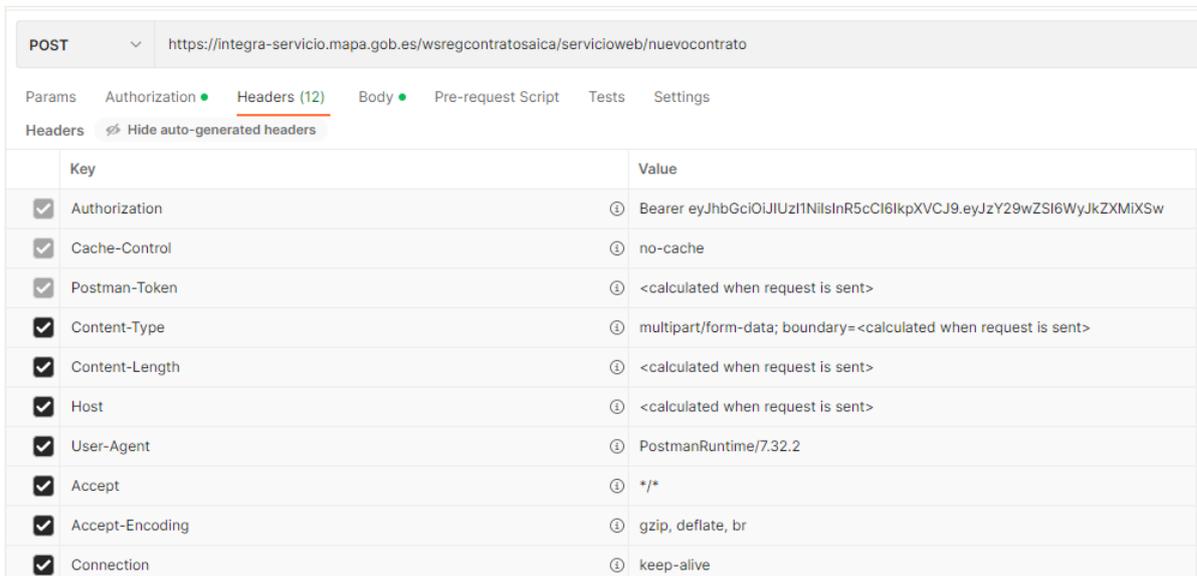
```
System.out.println("Llamando servicio securizado oauth...");
//Petición de Token
// AccessToken = Apartado 5.2 del documento
String json = "{\n" +
    "  \"company_id\": \"%NIF del comprador%\",\n" +
    "  \"producer_id\": \"%NIF del productor%\",\n" +
    "}";
Path pdfPath = Paths.get("C:\\doc1.pdf");
byte[] pdf = Files.readAllBytes(pdfPath);
Path pdfPath2 = Paths.get("C:\\doc2.pdf");
byte[] pdf2 = Files.readAllBytes(pdfPath2);
HttpEntity dataPrueba =
    MultipartEntityBuilder.create().setMode(HttpMultipartMode.BROWSER_COMPATIBLE).addPart("metadata",
    new StringBody(json, ContentType.MULTIPART_FORM_DATA.withCharset("UTF-8")))
    .addPart("files", new ByteArrayBody(pdf, "doc1.pdf"))
    .addPart("files", new ByteArrayBody(pdf2, "doc2.pdf"))
    .build();
HttpRequest requestPrueba =
    RequestBuilder.post("https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/nuevocontrato")
    .addHeader("Authorization", "Bearer " + accessToken)
    .setEntity(dataPrueba).build();
CloseableHttpResponse serviceResponse = client.execute(requestPrueba);
JsonObject serviceObject = Json.createReader(serviceResponse.getEntity().getContent()).readObject();
System.out.println(serviceObject.toString());
```

6.1.1.2. Ejemplo de llamada a *nuevocontrato* con Postman

La configuración de dicho método en POSTMAN se realiza de la siguiente manera:

- ▶ Seleccionar como tipo de método la opción POST.
- ▶ Indicar en la URL la dirección:

<https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/nuevocontrato>



Key	Value
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzY29wZSI6WyJKZXNlbnR5cCJ9.eyJzY29wZSI6WyJKZXNlbnR5cCJ9
<input checked="" type="checkbox"/> Cache-Control	no-cache
<input checked="" type="checkbox"/> Postman-Token	<calculated when request is sent>
<input checked="" type="checkbox"/> Content-Type	multipart/form-data; boundary=<calculated when request is sent>
<input checked="" type="checkbox"/> Content-Length	<calculated when request is sent>
<input checked="" type="checkbox"/> Host	<calculated when request is sent>
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.32.2
<input checked="" type="checkbox"/> Accept	*/*
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br
<input checked="" type="checkbox"/> Connection	keep-alive

Configuramos la llamada a *nuevocontrato* desde la pestaña *Body* seleccionando contenido tipo *form-data* y definiendo los dos parámetros que se muestran en pantalla.

NOTA: Antes de poder ejecutar la petición es necesario disponer de un token válido que se ha de incluir en la cabecera.

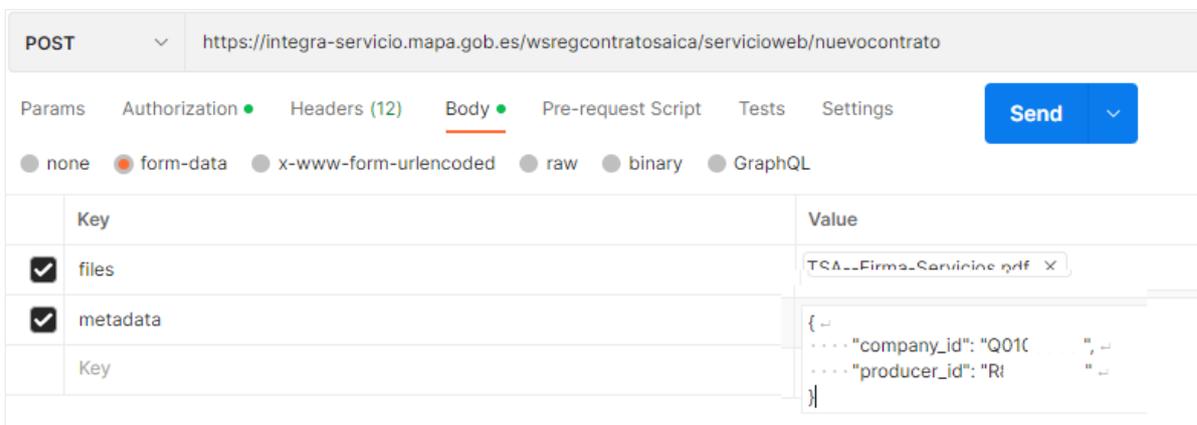
Se ha de definir un JSON con la siguiente estructura para metadata:

```

{
  "company_id": "%NIF, DNI o NIE del comprador%",
  "producer_id": "%NIF, DNI o NIE del productor%"
}

```

Se ha de incluir/seleccionar los ficheros asociados al contrato en la etiqueta files.



Key	Value
<input checked="" type="checkbox"/> files	TSA--Firma-Servicio.pdf
<input checked="" type="checkbox"/> metadata	<pre> { "company_id": "Q010", "producer_id": "R1" } </pre>

Una vez configurada esta llamada al método de *nuevocontrato*, pulsando en el botón SEND que está a la derecha de la URL, se enviará la petición al servicio.

6.1.2. Actualización de un contrato

Para actualizar un contrato alimentario ya existente en el sistema, el método de la API es:

URL: <https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/actualizarcontrato>

Método: POST

NOTA: Este método está preparado para recibir un multipart/form-data en el que una parte del contenido (los documentos) venga en un array de bytes y otra parte (los datos del comprador y productor) venga en un JSON.

Parámetros de cabecera:

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
<i>authorization</i>	Texto (216)	<i>Bearer + token</i> de seguridad obtenido para el usuario. Ver apartado 5.2 .	S

A continuación, se describen los **parámetros de entrada**:

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
<i>contract_id</i>	Texto (20)	Parámetro del JSON Metadata. %número de contrato% que corresponde con el contrato en RCA que se pretende actualizar.	S
<i>company_id</i>	Texto (20)	Parámetro del JSON Metadata. %NIF del comprador% que corresponde al comprador del contrato que se pretende actualizar.	S
<i>producer_id</i>	Texto (20)	Parámetro del JSON Metadata. %NIF del productor% que corresponde al productor con el que el comprador tiene el contrato que se pretende actualizar.	S
<i>files</i>	<i>File/bytes(5)</i>	Documento/s que anexan al contrato con objeto de actualizar/incluir nuevos documentos al contrato existente. Los documentos no podrán ser más de 5	S

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
		documentos en formato PDF plano ² . Los documentos no podrán pesar más de 5MB cada uno.	

A continuación, se describen los **parámetros de salida**:

RESPUESTA EXITO	VALOR
<i>status</i>	Resultado de la petición. "OK"
<i>contract_id</i>	Identificación del contrato por el que se podrá consultar en la aplicación web RCA.
<i>receipt</i>	Justificante de la actualización del contrato en Base64. Este justificante se puede obtener en formato PDF desde la aplicación RCA.

RESPUESTA ERROR	VALOR
<i>status</i>	Resultado de la petición. "ERROR"
<i>error_code</i>	Código de error devuelto por el API.
<i>error_description</i>	Descripción del mismo cuando falla la petición de actualización de un contrato alimentario.

Validaciones:

El método valida los siguientes puntos:

- ▶ El número de contrato exista en el sistema RCA.
- ▶ Los datos relativos del comprador y productor existan en el sistema RCA.
- ▶ El número de documentos que anexos al contrato no puede superar el límite establecido.
- ▶ El formato de todos los documentos sea el adecuado. Inicialmente PDF.

² El PDF plano es un PDF de salida sin elementos interactivos, como campos de formulario.

- ▶ El tamaño de cada documento no puede exceder de un tamaño establecido.

6.1.2.1. Ejemplo de llamada a *actualizarcontrato* (test Java)

```
System.out.println("----- INICIO TEST CLIENTE ----- ");
//Petición de Token
// AccessToken = Apartado 5.2 del documento
String json = "{\n" + "  \"contract_id\": \"%Numero de contrator%\",\n" + "  \"company_id\": \"%NIF del comprador%\",\n" + "  \"producer_id\": \"%NIF del productor%\",\n" + "  }";
Path pdfPath = Paths.get("C:\\prueba2.pdf");
byte[] pdf = Files.readAllBytes(pdfPath);
Path pdfPath2 = Paths.get("C:\\prueba3.pdf");
byte[] pdf2 = Files.readAllBytes(pdfPath2);
HttpEntity dataPrueba =
MultipartEntityBuilder.create().setMode(HttpMultipartMode.BROWSER_COMPATIBLE).addPart("metadata",
new StringBody(json, ContentType.MULTIPART_FORM_DATA.withCharset("UTF-8")))
.addPart("files", new ByteArrayBody(pdf, "prueba2.pdf"))
.addPart("files", new ByteArrayBody(pdf2, "prueba3.pdf"))
.build();
HttpRequest requestPrueba = RequestBuilder.post(" https://integra-
servicio.mapa.gob.es/wsregcontratosaic/servicioweb/actualizarcontrato")
.addHeader("Authorization", "Bearer " + accessToken)
.setEntity(dataPrueba).build();
CloseableHttpResponse serviceResponse = client.execute(requestPrueba);
JsonObject serviceObject= Json.createReader(serviceResponse.getEntity().getContent()).readObject();
System.out.println(serviceObject.toString());
```

6.1.2.2. Ejemplo de llamada a *actualizarcontrato* con Postman

La configuración de dicho método en Postman se realiza de la siguiente manera:

- ▶ Seleccionar como tipo de método la opción POST.
- ▶ Indicar en la URL la dirección:

<https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/actualizarcontrato>

6.1.3. Agregar productores primarios

Para agregar productores primarios asociados al comprador, el método de la API es:

URL: <https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/agregarproductores>

Método: POST

NOTA:

- Este método está preparado para recibir un JSON con uno o varios productores primarios.
- Se espera recibir la información con juego de caracteres UTF-8.

Parámetros de cabecera:

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
<i>authorization</i>	Texto (216)	<i>Bearer + token</i> de seguridad obtenido para el usuario. Ver apartado 5.2 .	S

A continuación, se describen los **parámetros de entrada:**

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
<i>company_id</i>	Texto (20)	Parámetro del JSON Metadata. %NIF del comprador% que corresponde al comprador al que se quieren asociar los productores.	S
<i>producers</i>	json	Array de productores primarios que se quieren agregar	S
<i>producer_id</i>	Texto (20)	Parámetro del JSON producers. %NIF del productor% que se desea agregar.	S
<i>registered_name</i>	Texto (100)	Parámetro del JSON producers. %Razón social del productor% que se desea agregar.	S
<i>address</i>	Texto (100)	Parámetro del JSON producers. %Dirección del productor% que se desea agregar.	S
<i>zip_code</i>	Texto (15)	Parámetro del JSON producers. %Código postal del productor% que se desea	S

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
		agregar. En caso de productor español no podrá tener más de 5 caracteres.	
<i>city_name</i>	Texto (100)	Parámetro del JSON producers. %Municipio del productor% que se desea agregar.	S
<i>region_id</i>	Number (2)	Parámetro del JSON producers. %Código de autonomía del productor% que se desea agregar. Será obligatorio en caso de productor español. En caso de productores extranjeros el valor válido es "".	N
<i>province_id</i>	Number (2)	Parámetro del JSON producers. %Código de provincia del productor% que se desea agregar. Será obligatorio en caso de productor español. En caso de productores extranjeros el valor válido es "".	N
<i>country_id</i>	Texto (2)	Parámetro del JSON producers. %Código de país del productor% que se desea agregar.	S

A continuación, se describen los **parámetros de salida**:

RESPUESTA	VALOR
<i>response</i>	JSON con todos los productores enviados y cada uno de sus resultados
<i>producer_id</i>	Parámetro del JSON response. %NIF del productor% que se ha intentado agregar
<i>status</i>	Resultado de la petición. "OK" en caso de que la inserción se haya podido realizar o "Error" en caso de que no se haya podido realizar la inserción
<i>error_code</i>	Código de error devuelto por la API. En caso de que el productor no haya podido ser agregado

<i>error_description</i>	Descripción del mismo cuando falla la inserción del productor.
--------------------------	--

Validaciones:

El método valida los siguientes puntos:

- ▶ El comprador exista en el sistema RCA.
- ▶ El productor primario no esté ya asociado al comprador.
- ▶ Los parámetros de entrada obligatorios tengan valor.
- ▶ Los parámetros de entrada cumplan con los tamaños máximos especificados.
- ▶ El código de País (*) exista.
- ▶ Los códigos de autonomía y provincia (*) existan y estén correctamente relacionadas en caso de tratarse de productor español y que sea "" en caso de productor extranjero.
- ▶ El NIF del productor sea correcto en caso de productor español.

(*) [Código ISO para el país \(en ISO 3166\)](#) y [Códigos INE de autonomía y provincia](#)

6.1.3.1. Ejemplo de llamada a *agregarproductores* (test Java)

```

System.out.println("Inicio Test");
//Petición de Token
// AccessToken = Apartado 5.2 del documento
String json = "{\n"
    + "\"company_id\": \"A34182766\", \n"
    + "\"producers\": [\n"
    + "{\n"
    + "\"producer_id\": \"19495526J\", \n"
    + "\"registered_name\": \"PRUEBA 220\", \n"
    + "\"address\": \"DIRECCIÓN DE PRUEBA 220\", \n"
    + "\"zip_code\": \"38001\", \n"
    + "\"city_name\": \"LOCALIDAD DE PRUEBA 220\", \n"
    + "\"region_id\": \"38\", \n"
    + "\"province_id\": \"5\", \n"
    + "\"country_id\": \"ES\" \n"
    + "}, \n"
    + "{\n"
    + "\"producer_id\": \"23895677B\", \n"
    + "\"registered_name\": \"PRUEBA 219\", \n"
  
```

```

+ "\"address\": \"DIRECCIÓN DE PRUEBA 219\", \"n\"
+ "\"zip_code\": \"44001\", \"n\"
+ "\"city_name\": \"LOCALIDAD DE PRUEBA 219\", \"n\"
+ "\"region_id\": \"44\", \"n\"
+ "\"province_id\": \"2\", \"n\"
+ "\"country_id\": \"ES\", \"n\"
+ \"} \"n\"
+ \"} \"n\"
+ \"} \"n\";

HttpEntity dataPrueba = new ByteArrayEntity(json.getBytes(\"UTF-8\"));

HttpPost requestPrueba = new HttpPost(\"https://integra-servicio.mapa.gob.es/wsregcontratosai/servicioweb/agregarproductores\");
requestPrueba.addHeader(\"Authorization\", \"Bearer \" + accessToken);
requestPrueba.setEntity(dataPrueba);

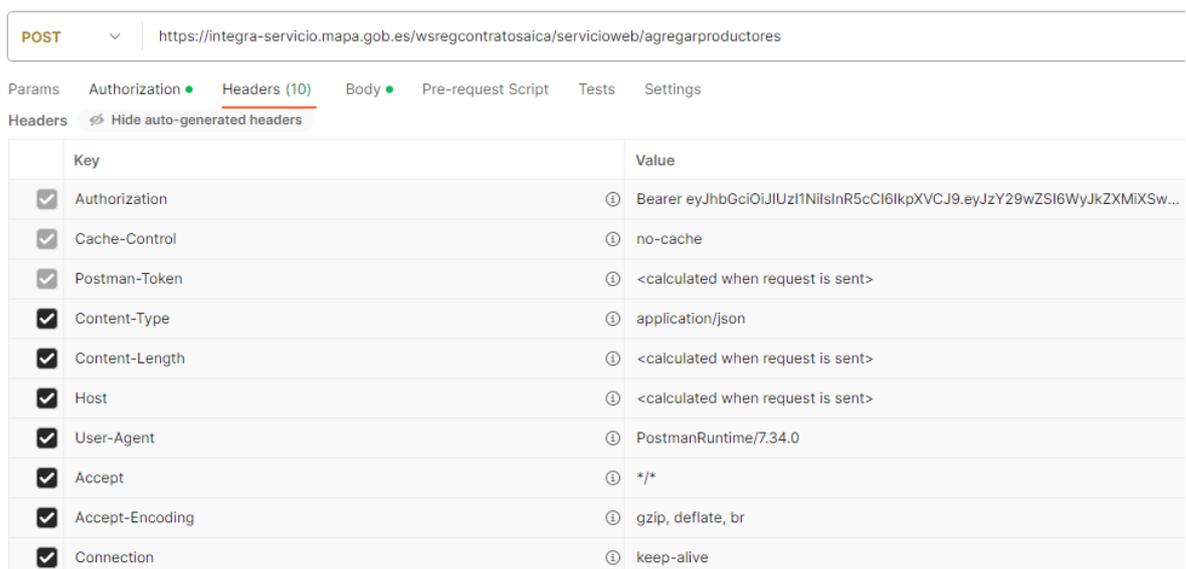
CloseableHttpResponse serviceResponse = client.execute(requestPrueba);
JsonObject serviceObject = Json.createReader(serviceResponse.getEntity().getContent()).readObject();
System.out.println(serviceObject.toString());

```

6.1.3.2. Ejemplo de llamada a agregarproductores con Postman

La configuración de dicho método en Postman se realiza de la siguiente manera:

- ▶ Seleccionar como tipo de método la opción POST.
- ▶ Indicar en la URL la dirección:
<https://integra-servicio.mapa.gob.es/wsregcontratosai/servicioweb/agregarproductores>



Key	Value
<input checked="" type="checkbox"/> Authorization	Bearer eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzY29wZSI6ImVkaXZlbnR5IiwiaWF0IjoiMTY1MjM0NTY3In0
<input checked="" type="checkbox"/> Cache-Control	no-cache
<input checked="" type="checkbox"/> Postman-Token	<calculated when request is sent>
<input checked="" type="checkbox"/> Content-Type	application/json
<input checked="" type="checkbox"/> Content-Length	<calculated when request is sent>
<input checked="" type="checkbox"/> Host	<calculated when request is sent>
<input checked="" type="checkbox"/> User-Agent	PostmanRuntime/7.34.0
<input checked="" type="checkbox"/> Accept	/*/*
<input checked="" type="checkbox"/> Accept-Encoding	gzip, deflate, br
<input checked="" type="checkbox"/> Connection	keep-alive

Configuramos la solicitud a *agregarproductores* desde la pestaña *Body*, seleccionando contenido tipo *json* y definiendo los parámetros que se muestran en pantalla.

Nota: Antes de poder ejecutar la petición es necesario disponer de un token válido que se ha de incluir en la cabecera.

Se ha de definir un JSON con la siguiente estructura:

```
{
  "company_id": "%NIF, DNI o NIE del comprador%",
  "producers": [
    {
      "producer_id": "%NIF, DNI o NIE del productor%",
      "registered_name": "%razón social del productor%",
      "country_id": "%código de país del productor%",
      "region_id": "%código de comunidad autónoma del productor%",
      "province_id": "%código de provincia del productor%",
      "city_name": "%localidad del productor%",
      "address": "%dirección del productor%",
      "zip_code": "%código postal del productor%"
    },
    {
      "producer_id": "%NIF, DNI o NIE del productor3%",
      "registered_name": "%razón social del productor3%",
      "country_id": "%código de país del productor3%",
      "region_id": "%código de comunidad autónoma del productor3%",
      "province_id": "%código de provincia del productor3%",
      "city_name": "%localidad del productor3%",
      "address": "%dirección del productor3%",
      "zip_code": "%código postal del productor3%"
    },
    ...
    {
      "producer_id": "%NIF, DNI o NIE del productorn%",
      "registered_name": "%razón social del productorn%",
      "country_id": "%código de país del productorn%",
      "region_id": "%código de comunidad autónoma del productorn%",
      "province_id": "%código de provincia del productorn%",
      "city_name": "%localidad del productorn%",
      "address": "%dirección del productorn%",
      "zip_code": "%código postal del productorn%"
    }
  ]
}
```

POST | <https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/agregarproductores> | **Send**

Params | Authorization | Headers (10) | **Body** | Pre-request Script | Tests | Settings | Cookies

none | form-data | x-www-form-urlencoded | raw | binary | GraphQL | **JSON** | Beautify

```

1  {
2    "company_id": "PRUEBAEXCEL3",
3    "producers": [
4      {
5        "producer_id": "19495526J",
6        "registered_name": "PRUEBA 220",
7        "address": "DIRECCIÓN DE PRUEBA 220"
8        "zip_code": "38001",
9        "city_name": "LOCALIDAD DE PRUEBA 220",
10       "province_id": 38,
11       "region_id": 5,
12       "country_id": "ES"
13     },
14     {
15       "producer_id": "88317715S",
16       "registered_name": "NOMBRE 88317715S",
17       "address": "DIRECCION 88317715S",
18       "zip_code": "18001",
19       "city_name": "LOCALIDAD 88317715S",
20       "province_id": 18.
  
```

Una vez configurada esta llamada al método de *agregarproductores*, pulsando en el botón SEND que está a la derecha de la URL, se enviará la petición al método.

Si la configuración es correcta, el servicio nos devolverá un JSON como respuesta con un array con el resultado de agregar cada uno de los productores enviados con un "OK" en caso de inserción correcta o con un "Error" y la descripción del error en caso contrario.

```

1  {
2    "response": [
3      {
4        "producer_id": "35762951K",
5        "status": "OK"
6      },
7      {
8        "producer_id": "17773199H",
9        "status": "Error",
10       "error_code": ";43;",
11       "error_description": "Cod provincia se ha superado el nº de caracteres permitido. "
12     },
13     {
14       "producer_id": "43388228Q",
15       "status": "Error",
16       "error_code": ";24;26;",
17       "error_description": "Es obligatorio indicar una Direccion. Es obligatorio indicar un Código Postal. "
18     }
19   ]
20 }
  
```

6.1.3.3.Respuesta

Si los datos de la llamada cumplen con todas las validaciones de entrada, se realiza la inserción de los productores enviados.

En este caso, el método devuelve una respuesta JSON con la información sobre el estado de inserción de cada uno de los productores enviados.

```
{
  "response": [
    {
      "producer_id": "19087010A"
      "status": "OK",
    },
    {
      "producer_id": "19087010A"
      "status": "OK",
    }
  ]
}
```

6.1.4. Modificar productores primarios

Para modificar productores primarios asociados al comprador, el método de la API es:

URL: <https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/modificarproductores>

Método: POST

NOTA:

- Este método está preparado para recibir un JSON con uno o varios productores primarios.
- Se espera recibir la información con juego de caracteres UTF-8.

Parámetros de cabecera:

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
<i>authorization</i>	Texto (216)	<i>Bearer + token</i> de seguridad obtenido para el usuario. Ver apartado 5.2 .	S

A continuación, se describen los **parámetros de entrada**:

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
<i>company_id</i>	Texto (20)	Parámetro del JSON Metadata. %NIF del comprador% que corresponde al comprador del que se quieren modificar los productores.	S
<i>producers</i>	json	Array de productores primarios que se quieren modificar.	S
<i>producer_id</i>	Texto (20)	Parámetro del JSON producers. %NIF del productor% que se desea modificar.	S
<i>registered_name</i>	Texto (100)	Parámetro del JSON producers. %Razón social del productor% que se desea modificar.	S
<i>address</i>	Texto (100)	Parámetro del JSON producers. %Dirección del productor% que se desea modificar.	S
<i>zip_code</i>	Texto (15)	Parámetro del JSON producers. %Código postal del productor% que se desea	S

NOMBRE	TIPO	VALOR	¿OBLIGATORIO?
		modificar. En caso de productor español no podrá tener más de 5 caracteres.	
<i>city_name</i>	Texto (100)	Parámetro del JSON producers. %Municipio del productor% que se desea modificar.	S
<i>region_id</i>	Number (2)	Parámetro del JSON producers. %Código de autonomía% del productor que se desea modificar. Será obligatorio en caso de productor español. En caso de productores extranjeros el valor válido es "".	N
<i>province_id</i>	Number (2)	Parámetro del JSON producers. %Código de provincia del productor% que se desea modificar. Será obligatorio en caso de productor español. En caso de productores extranjeros el valor válido es "".	N
<i>country_id</i>	Texto (2)	Parámetro del JSON producers. %Código de país del productor% que se desea modificar.	S

A continuación, se describen los **parámetros de salida**:

RESPUESTA	VALOR
<i>response</i>	JSON con todos los productores enviados y cada uno de sus resultados
<i>producer_id</i>	Parámetro del JSON response. %NIF del productor% que se ha intentado modificar
<i>status</i>	Resultado de la petición. "OK" en caso de que la modificación se haya podido realizar o "Error" en caso de que no se haya podido realizar la modificación
<i>error_code</i>	Código de error devuelto por la API. En caso de que el productor no haya podido ser modificado

<i>error_description</i>	Descripción del mismo cuando falla la modificación del productor.
--------------------------	---

Validaciones:

El método valida los siguientes puntos:

- ▶ El comprador exista en el sistema RCA.
- ▶ El productor primario esté asociado al comprador.
- ▶ Los parámetros de entrada obligatorios tengan valor.
- ▶ Los parámetros de entrada cumplan con los tamaños máximos especificados.
- ▶ El código de País (*) exista.
- ▶ Los códigos de autonomía y provincia (*) existan y estén correctamente relacionadas en caso de tratarse de productor español y que sean "" en caso de productor extranjero.
- ▶ El NIF del productor sea correcto en caso de productor español.

(*) [Código ISO para el país \(en ISO 3166\)](#) y [Códigos INE de autonomía y provincia](#)

6.1.4.1. Ejemplo de llamada a modificar productores (test Java)

```

System.out.println("Inicio Test");

//Petición de Token
// AccessToken = Apartado 5.2 del documento
String json = "{\n"
    + "\"company_id\": \"A34182766\", \n"
    + "\"producers\": [\n"
    + "{\n"
    + "\"producer_id\": \"19495526J\", \n"
    + "\"registered_name\": \"PRUEBA 220\", \n"
    + "\"address\": \"DIRECCIÓN DE PRUEBA 220\", \n"
    + "\"zip_code\": \"38001\", \n"
    + "\"city_name\": \"LOCALIDAD DE PRUEBA 220\", \n"
    + "\"region_id\": \"38\", \n"
    + "\"province_id\": \"5\", \n"
    + "\"country_id\": \"ES\", \n"
    + "}, \n"
    + "{\n"
    + "\"producer_id\": \"23895677B\", \n"
  
```

```

+ "\"registered_name\": \"PRUEBA 219\", \n"
+ "\"address\": \"DIRECCIÓN DE PRUEBA 219\", \n"
+ "\"zip_code\": \"44001\", \n"
+ "\"city_name\": \"LOCALIDAD DE PRUEBA 219\", \n"
+ "\"region_id\": \"44\", \n"
+ "\"province_id\": \"2\", \n"
+ "\"country_id\": \"ES\" \n"
+ " \n"
+ " ] \n"
+ " }";

HttpEntity dataPrueba = new ByteArrayEntity(json.getBytes("UTF-8"));

HttpPost requestPrueba = new HttpPost("https://integra-servicio.mapa.gob.es/wsregcontratosaiica/servicioweb/modificarproductores");

requestPrueba.addHeader("Authorization", "Bearer " + accessToken);

requestPrueba.setEntity(dataPrueba);

CloseableHttpResponse serviceResponse = client.execute(requestPrueba);

JsonObject serviceObject = Json.createReader(serviceResponse.getEntity().getContent()).readObject();

System.out.println(serviceObject.toString());

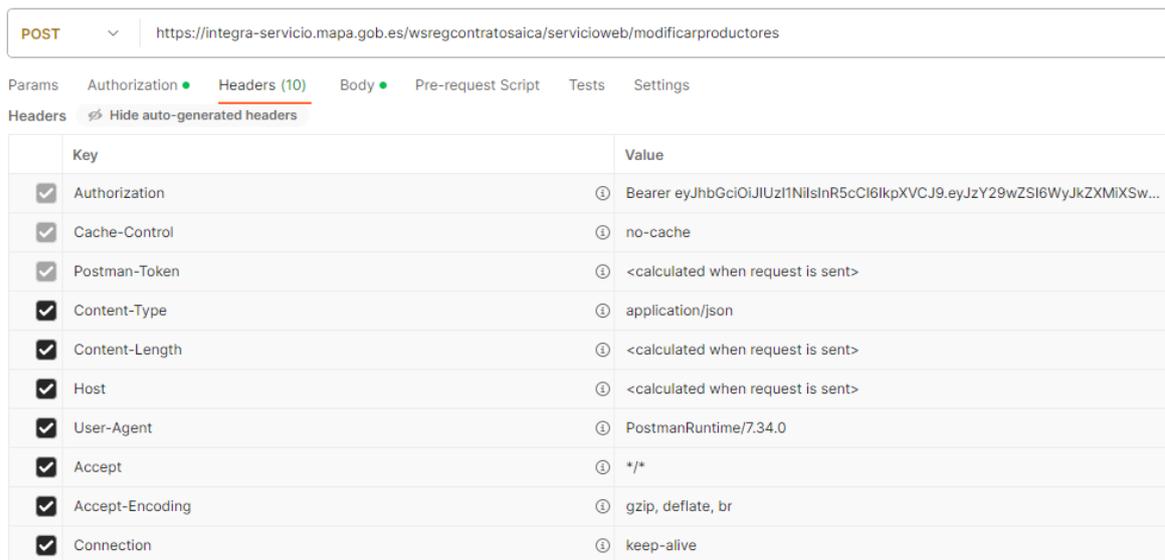
```

6.1.4.2. Ejemplo de llamada a modificarproductores con Postman

La configuración de dicho método en Postman se realiza de la siguiente manera:

- ▶ Seleccionar como tipo de método la opción POST.
- ▶ Indicar en la URL la dirección:

<https://integra-servicio.mapa.gob.es/wsregcontratosaiica/servicioweb/modificarproductores>



Configuramos la solicitud a *modificarproductores* desde la pestaña *Body*, seleccionando contenido tipo *json* y definiendo los parámetros que se muestran en pantalla.

Nota: Antes de poder ejecutar la petición es necesario disponer de un token válido que se ha de incluir en la cabecera.

Se ha de definir un JSON con la siguiente estructura:

```
{
  "company_id": "%NIF, DNI o NIE del comprador%",
  "producers": [
    {
      "producer_id": "%NIF, DNI o NIE del productor%",
      "registered_name": "%razón social del productor%",
      "country_id": "%código de país del productor%",
      "region_id": "%código de comunidad autónoma del productor%",
      "province_id": "%código de provincia del productor%",
      "city_name": "%localidad del productor%",
      "address": "%dirección del productor%",
      "zip_code": "%código postal del productor%"
    },
    {
      "producer_id": "%NIF, DNI o NIE del productor3%",
      "registered_name": "%razón social del productor3%",
      "country_id": "%código de país del productor3%",
      "region_id": "%código de comunidad autónoma del productor3%",
      "province_id": "%código de provincia del productor3%",
      "city_name": "%localidad del productor3%",
      "address": "%dirección del productor3%",
      "zip_code": "%código postal del productor3%"
    },
    ...
    {
      "producer_id": "%NIF, DNI o NIE del productorn%",
      "registered_name": "%razón social del productorn%",
      "country_id": "%código de país del productorn%",
      "region_id": "%código de comunidad autónoma del productorn%",
      "province_id": "%código de provincia del productorn%",
      "city_name": "%localidad del productorn%",
      "address": "%dirección del productorn%",
      "zip_code": "%código postal del productorn%"
    }
  ]
}
```

}

POST ▼ https://integra-servicio.mapa.gob.es/wsregcontratosaic/servicioweb/modificarproductores Send ▼

Params Authorization ● Headers (10) **Body ●** Pre-request Script Tests Settings Cookies

● none ● form-data ● x-www-form-urlencoded ● **raw** ● binary ● GraphQL **JSON** ▼ Beautify

```

1  {
2  "company_id":"PRUEBAEXCEL3",
3  "producers":[
4  .....
5  ..... "producer_id":"19495526J",
6  ..... "registered_name":"MOD",
7  ..... "address":"DIRECCIÓN DE PRUEBA 220"
8  ..... "zip_code":"38001",
9  ..... "city_name":"LOCALIDAD DE PRUEBA 220",
10 ..... "province_id":38,
11 ..... "region_id":5,
12 ..... "country_id":"ES"
13 .....
14 .....
15 ..... "producer_id":"88317715S",
16 ..... "registered_name":"NOME 88317715S",
17 ..... "address":"DIRECCION 88317715S",
18 ..... "zip_code":"18001",
19 ..... "city_name":"LOCALIDAD 88317715S",
20 ..... "province id":18.

```

Una vez configurada esta llamada al método de *modificarproductores*, pulsando en el botón SEND que está a la derecha de la URL, se enviará la petición al método.

Si la configuración es correcta, el servicio nos devolverá un JSON como respuesta con un array con el resultado de la modificación de cada uno de los productores enviados con un "OK" en caso de modificación correcta o con un "Error" y la descripción del error en caso contrario.

```

1  [
2  "response": [
3  {
4  "producer_id": "35762951K",
5  "status": "OK"
6  },
7  {
8  "producer_id": "17773199H",
9  "status": "Error",
10 "error_code": ";43;",
11 "error_description": "Cod provincia se ha superado el nº de caracteres permitido. "
12 },
13 {
14 "producer_id": "43388228Q",
15 "status": "Error",
16 "error_code": ";24;26;",
17 "error_description": "Es obligatorio indicar una Direccion. Es obligatorio indicar un Código Postal. "
18 }
19 ]
20 ]

```

6.1.4.3.Respuesta

Si los datos de la llamada cumplen con todas las validaciones de entrada, se realiza la modificación de los productores enviados.

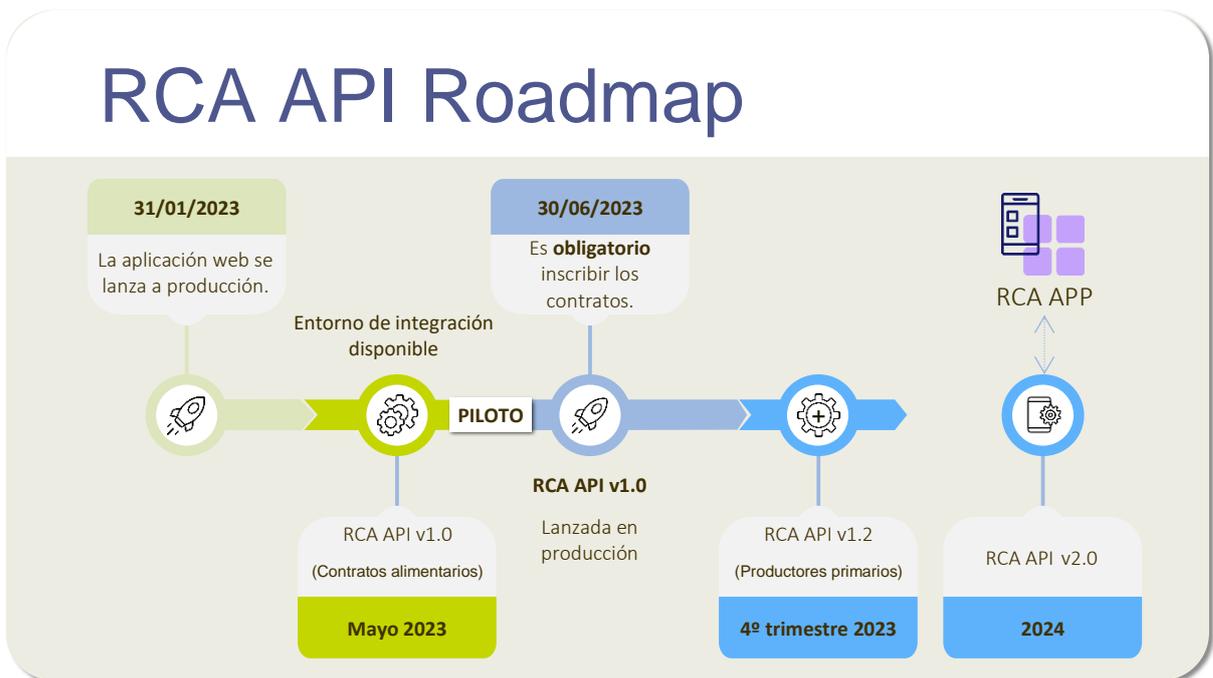
En este caso, el método devuelve una respuesta JSON con la información sobre el estado de modificación de cada uno de los productores enviados.

```
{
  "response": [
    {
      "producer_id": "19087010A"
      "status": "OK",
    },
    {
      "producer_id": "19087010A"
      "status": "OK",
    }
  ]
}
```

7. Roadmap

Como ya se ha mencionado anteriormente, RCA API sale a producción el **1 de julio de 2023** publicando los *endpoints* de inscribir un nuevo contrato y actualizar un contrato.

En el cuarto trimestre de 2023 se agregarán los *endpoints* que gestionarán los productores primarios y las agrupaciones de productores.



8. Anexos

En este apartado se indica información adicional no recogida en el resto de los apartados de este documento.

8.1. Ejemplo de justificante de inscripción de nuevo contrato o actualización

A continuación, se muestra un ejemplo del justificante de inscripción generado por la API, y facilitado al cliente en la respuesta a cada petición de nuevo contrato o actualización de un contrato. Para poder obtener el documento en formato PDF es necesario realizar la conversión de la respuesta obtenida de la API RCA en formato Base64:

Registro de Contratos Alimentarios

JUSTIFICANTE DE INSCRIPCIÓN DE CONTRATO ALIMENTARIO

INSCRIPCIÓN (A TRAVÉS DE LA API DE SERVICIOS)

Fecha y hora de inscripción: 23/09/2023 15:20:02
Número de contrato: 123-5A3-980
Operación: Nuevo contrato

USUARIO DE LA API DE SERVICIOS

Usuario: A3B1C2
Acuerdo de uso firmado el: 13/08/2023
Rol en el comprador: Consumidor de la API

COMPRADOR (Sujeto obligado)

Núm. de doc. identificativo: U05791728
NIF, DNI o NIE
Razón social: Acme Fruits Company, Inc.

PRODUCTOR PRIMARIO

Núm. de doc. identificativo: G02643112
NIF, DNI o NIE
Razón social: Fruits and Vegetables Company, Inc.

DOCUMENTOS INSCRITOS

NOMBRE	TAMAÑO (MB)	HUELLA DIGITAL
contrato.pdf	3,567	e0b3b684b15f8b54d4194eecd3e186b1778f6b76f6c5abe041a1a7ed8132407

8.2. Errores

Se incluye en este documento la lista detallada de errores controlados. En cada mensaje de error devuelto por el servicio el código se acompaña de un texto descriptivo del mismo.

CODIGO	VALOR
1	La estructura del JSON de la petición no es la esperada. Por favor revise la información.
2	Alguno de los valores indicados en el JSON no se corresponde con el formato esperado.
3	Faltan datos en la cabecera de la petición.
4	Credenciales de autenticación en la cabecera de la petición incorrectas.
5	La petición no atiende a los estándares multipart/form-data.
6	Ha ocurrido un error al inscribir el contrato. Se ha cancelado la operación.
7	No se recibieron archivos.
8	Ha ocurrido un error durante el proceso de grabación del contrato. La operación se ha cancelado.
9	Ha ocurrido un error durante la generación del justificante. La operación se ha cancelado.
10	No se encontró el nif del productor primario registrado relacionado con el nif de comprador.
11	No se encontró un nif de comprador registrado.
12	No se recibió un nif de comprador.
13	No se recibió un nif de productor.
14	Se debe recibir la etiqueta 'files' que contiene los archivos y la etiqueta 'metadata' que contenga el JSON. No se permiten etiquetas adicionales.
15	La etiqueta 'metadata' debe contener un JSON.
16	La etiqueta 'files' espera recibir los archivos .pdf.
17	El usuario utilizado para acceder al servicio no es el asignado al comprador. Consulte de nuevo las credenciales de acceso en la aplicación web de RCA.

18	El número de contrato aportado no existe.
19	Este método no permite recibir un número de contrato, por favor revise la información.
20	Este método debe recibir el número de contrato que se quiere actualizar, por favor revise la información.
21	Se han detectado documentos con formato y/o tamaños incorrectos en la petición.
22	Se ha superado el número máximo permitido de documentos en la petición.
23	No se encontró el nif del productor primario registrado relacionado con el nif de comprador, o bien el número de contrato aportado no existe.
24	Es obligatorio indicar una Dirección.
25	El comprador no está registrado correctamente en RCA, tiene que finalizar el proceso de alta asociando algún representante legal.
26	Es obligatorio indicar un Código Postal.
27	Formato del código postal no válido.
28	Es obligatorio indicar una Autonomía.
29	Es obligatorio indicar una Provincia.
30	Es obligatorio indicar un código de País.
31	Es obligatorio indicar una Localidad.
32	Es obligatorio indicar la Razón Social Productor.
33	Código postal no válido.
34	Cifnif no válido.
35	Productor primario ya existe.
36	Código de país no existe.
37	Código de autonomía no existe.
38	Código de provincia no corresponde con código de autonomía.

39	Cifnif productor se ha superado el nº de caracteres permitido.
40	Razon social se ha superado el nº de caracteres permitido.
41	ID pais se ha superado el nº de caracteres permitido.
42	Cod autonomia se ha superado el nº de caracteres permitido.
43	Cod provincia se ha superado el nº de caracteres permitido.
44	Localidad se ha superado el nº de caracteres permitido.
45	Dirección se ha superado el nº de caracteres permitido.
46	Cod postal se ha superado el nº de caracteres permitido.
47	Se ha registrado correctamente.
48	Se ha producido un error. Contacte con el administrador.
49	La Comunidad Autónoma ha de estar vacía para países extranjeros.
50	La Provincia ha de estar vacía para países extranjeros.
51	Se ha excedido el número máximo permitido de productores primarios a agregar en una misma petición.
52	Productor primario no está registrado.
53	Se han detectado caracteres no permitidos en la petición.

8.2.1. Caracteres no permitidos

No se permite ningún carácter que pueda afectar a la estructura correcta que de un JSON.

Por motivos de seguridad, los siguientes caracteres y cadenas de caracteres se verán representados como espacios en blanco en cualquiera de los métodos disponibles del servicio web:

>, <, \\

Los siguientes caracteres y cadenas no están permitidos en los nombres de documentos:

\, /, *, :, ;, ?, ", <, >, |, ;