



DOCUMENTACIÓN API DEL SERVICIO DE LISTA ROBINSON V1.3.1

CONTROL DOCUMENTAL			
Versión	Fecha	Modificaciones	Revisor
v0.1	05/02/2019	Documentación preproducción.	E. S.
v1.0	08/04/2019	Documentación producción.	E. S.
v1.0.1	25/04/2019	Correcciones menores.	E. S.
v.1.0.2	24/07/2019	Añadido ejemplo alta concurrencia. Correcciones menores.	E. S.
v.1.1.0	24/01/2020	Método POST. Mejora de la documentación sobre errores. Revisión recomendaciones. Correcciones menores.	E. S.
v.1.1.1	18/05/2020	Incrementados los registros por peticiones. Añadidos ejemplos de firma con método POST.	E. S.
v.1.3	23/05/2023	Incrementado el número de registros máximos por llamada a API en método POST. Aclaración concurrencia por endpoint. La numeración de índice ahora coincide con el número de página de documento.	E. S.
v.1.3.1	15/06/2023	Correcciones menores.	E. S.

Contenido

1. Implementación de la API del Servicio de Lista Robinson.....	5
1.1. Formas de implementar la API.....	5
1.2. Cliente API.....	5
1.3. Desarrollo propio.....	5
1.4. Herramientas desarrolladas por un tercero.....	5
2. Acerca de la implementación de la API mediante un desarrollo propio.....	6
2.1. Autenticación y obtención de claves.....	6
2.2. Cómputo e imputación de consultas.....	6
2.3. Preparación de las consultas.....	7
2.4. Realización de pruebas durante el desarrollo.....	7
3. Autenticación de las peticiones.....	8
3.1. De la autenticación.....	8
3.2. Ejemplo de petición POST.....	8
3.2.1. Con los parámetros de autenticación en la URL.....	8
3.2.2. Con los parámetros de autenticación en las cabeceras.....	9
4. Gestión de errores consultando la API.....	10
5. Concurrencia y ejemplos de uso.....	12
5.1. Concurrencia.....	12
5.2. Ejemplo de uso: depuración con baja concurrencia.....	12
5.3. Ejemplo de uso: depuración con alta concurrencia.....	12
6. Datos generales.....	14
7. Endpoints.....	14
7.1. Consultar si un registro se encuentra en la Lista Robinson.....	14
7.1.1. Información.....	14
7.1.2. Cómo generar un hash.....	15
7.1.3. Orden de concatenación de los campos para computar la huella digital de cada registro a consultar.....	15
7.1.4. Preparación de la consulta.....	16
7.1.5. Ejemplos.....	16
7.2. Obtener cuántas consultas se han realizado a través de la API.....	20
7.2.1. Información.....	20
8. <i>Endpoints</i> obsoletos.....	21
8.1. Acerca de estos <i>endpoint</i>	21
8.2. [OBSOLETO] Consultar si un registro se encuentra en la Lista Robinson.....	21

8.2.1. Motivo obsolescencia.....	21
8.2.2. Información	21
8.2.3. Cómo generar un hash	22
8.2.4. Orden de concatenación de campos.....	22
8.2.5. Ejemplos	23

1. Implementación de la API del Servicio de Lista Robinson

1.1. Formas de implementar la API

A la hora de implementar la API del Servicio de Lista Robinson en el sistema de la entidad consultante, puede optarse por:

- implementar el Cliente API desarrollado por Adigital, o
- implementar la API con un desarrollo propio, siguiendo la especificación establecida por este documento,
- implementar una herramienta desarrollada por un tercero.

1.2. Cliente API

El [Cliente API](https://cliente.api.listarobinson.es/)¹ es la herramienta gratuita y de [código abierto](#)² desarrollada por Adigital para facilitar el proceso de consulta de la Lista Robinson a través de la API del Servicio.

Puede utilizarse a través de su interfaz de usuario, ya sea en navegador o descargando e instalando la aplicación en el equipo, o a través de su ejecutable para línea de comandos (CLI), recomendada para su uso en servidores y que permite una completa automatización del proceso.

El Cliente API implementa la especificación y recomendaciones establecidas en este documento y será actualizado de forma periódica de acuerdo con él.

1.3. Desarrollo propio

En este documento se desarrollan los aspectos técnicos necesarios para una implementación completa de la API del Servicio de Lista Robinson, incluyendo recomendaciones de implementación y las distintas formas de realizar las llamadas a la API. Como referencia y apoyo al desarrollo de una herramienta propia, [puede consultarse el código fuente del Cliente API](#). También es posible realizar modificaciones al Cliente API o reutilizar alguno de sus componentes.

A fin de reducir los costes de desarrollo y mantenimiento, de manera general se recomienda el uso del Cliente API, ya sea en sus versiones con interfaz gráfica o en su versión de línea de comandos (CLI).

1.4. Herramientas desarrolladas por un tercero

También es posible implementar herramientas desarrolladas por terceros para la consulta de la Lista Robinson a través de la API. Las características concretas de estas herramientas, así como su funcionamiento y compatibilidad, son establecidas por las entidades que las desarrollan, pudiendo adaptarse mejor a ciertos casos de uso o pudiendo incorporar mejoras respecto a las recomendaciones y características técnicas desarrolladas en este documento.

¹ <https://cliente.api.listarobinson.es/>

² <https://github.com/adigital-org/slr-client>

2. Acerca de la implementación de la API mediante un desarrollo propio

2.1. Autenticación y obtención de claves

Todas las peticiones a la API deben autenticarse según *AWS Signature Version 4*³. Las credenciales para la autenticación, formadas por una clave de acceso y un secreto, pueden obtenerse a través del panel “[Suscripción](#)” de la plataforma del Servicio de Lista Robinson⁴.

La clave es almacenada por la plataforma con la finalidad de imputar el consumo de API a la entidad consultante, pero el secreto no queda registrado en la plataforma, por lo que la entidad consultante debe almacenar tanto la clave como el secreto en un lugar seguro.

Si la entidad consultante perdiese las credenciales no podrá recuperarlas a través de la plataforma, únicamente podrá generar una nueva pareja de credenciales. La generación de nuevas credenciales desautoriza el juego anterior.

La plataforma conserva un histórico interno de las claves de acceso (no de los secretos) asociado a la cuenta de usuario que las genera y la fecha y la hora de creación.

2.2. Cómputo e imputación de consultas

Todas las peticiones POST con respuesta con código HTTP 200 y las GET con respuesta HTTP 200 o 404 son computadas e imputadas a la entidad consultante. El número de consultas imputadas depende del número de registros consultados. Así, en una petición POST con 10 registros consultados, se imputarán 10 consultas, en una petición POST o GET con un único registro computado, se imputará una consulta.

Las respuestas HTTP 400 (petición incorrecta) son imputadas a la entidad con un coste equivalente al de la consulta de 1 registro.

El resto de respuestas se consideran errores internos y no serán imputadas. En el punto [4 GESTIÓN DE ERRORES CONSULTANDO LA API](#) se detalla cómo se deben gestionar las distintas respuestas obtenidas por la API. Ignorar estas indicaciones puede suponer el bloqueo del acceso temporal o permanente al Servicio y, en su caso, costes adicionales.

El número de consultas realizado por una entidad se contabiliza una vez al día. Por lo tanto, los resultados devueltos por la consulta API descrita en el punto [7.2 OBTENER CUÁNTAS CONSULTAS SE HAN REALIZADO A TRAVÉS DE LA API](#) no son en tiempo real y se corresponden con el último recuento realizado por la plataforma.

³ https://docs.aws.amazon.com/es_es/general/latest/gr/signature-version-4.html

⁴ <https://www.listarobinson.es/enterprise/subscription-method>

2.3. Preparación de las consultas

Las consultas deberán realizarse según lo dispuesto en este documento y en el [Reglamento del Servicio de Lista Robinson](#)⁵.

2.4. Realización de pruebas durante el desarrollo

No existe un entorno específico para la realización de pruebas, no obstante, pueden utilizarse los siguientes componentes del Cliente API para la verificación de los desarrollos propios:

- Para probar la normalización y pseudonimización de los datos, pueden compararse los resultados del desarrollo propio con los generados por el componente [SLRUtils.js](#)⁶ del Cliente API. SLRUtils.js incorpora su propia documentación y va acompañado un [conjunto de pruebas](#)⁷ concretas que también pueden extrapolarse a un desarrollo propio.
- Para probar el sistema de gestión de errores y los elementos básicos de la composición de las peticiones y el manejo de las respuestas, puede ser de utilidad el [SimpleApiSimulator](#)⁸, un simulador básico de la API del Servicio de Lista Robinson que permite simular respuestas y errores de la API. No se trata de un simulador exhaustivo y, por lo tanto, no simula el funcionamiento de la API al completo, como se indica en su [documentación](#), pero puede ser de utilidad para detectar los principales errores.
- Para el proceso de firma de las peticiones y su autenticación ante la API, puede consultarse la documentación indicada más abajo y el componente [Signing.js](#)⁹ del Cliente API. En caso de error de autenticación en la API, el *body* de la respuesta recibida incorpora información de utilidad.

⁵ <https://legal.adigital.org/reglamento-slr/>

⁶ <https://github.com/adigital-org/slr-client/blob/master/src/util/SLRUtils.js>

⁷ <https://github.com/adigital-org/slr-client/blob/master/src/util/SLRUtils.test.js>

⁸ <https://github.com/adigital-org/slr-client/tree/master/utis/SimpleApiSimulator>

⁹ <https://github.com/adigital-org/slr-client/blob/master/src/util/Signing.js>

3. Autenticación de las peticiones

3.1. De la autenticación

Las peticiones realizadas a la API de Lista Robinson deben autenticarse de acuerdo con AWS *Signature V4*, cuya documentación puede encontrarse [aquí](#)¹⁰ o, en PDF, [aquí](#)¹¹. Cabe destacar que los parámetros de autenticación de la petición pueden enviarse en las cabeceras (*headers*) o en la URL.

Si las peticiones se van a lanzar desde un navegador web, los parámetros de autenticación deben enviarse en la URL, de lo contrario podría producirse una excepción [CORS](#)¹² que bloquee la petición.

Puede consultarse un ejemplo completo en el componente [Signing.js](#)¹³ del [Cliente API](#)¹⁴, la herramienta de consulta desarrollada por Adigital.

3.2. Ejemplo de petición POST

A continuación, se muestra un ejemplo de los principales pasos para firmar una petición POST, tanto con las credenciales enviadas en la URL como enviadas en las cabeceras de la petición HTTP.

Las credenciales utilizadas en ambos ejemplos, son:

```
key: AKIAXXXXXXXXXXXXXXXXXX
secret: xxxxxxxxxxxx
```

Y el *body* de ambas peticiones:

```
d47b1054350b5b3c8ccf923b13da089a396bd18f298489bcab90e21baf23a98d,25a60cf567fb3fd948bc65f70e275a8d6077e6b38e01e54aa56ac87626438315
```

3.2.1. Con los parámetros de autenticación en la URL

3.2.1.1. Solicitud canónica (*CanonicalRequest*):

```
POST
/v1/api/user
X-Amz-Algorithm=AWS4-HMAC-SHA256&X-Amz-
Credential=AKIAXXXXXXXXXXXXXXXXXX%2F20200518%2Ffeu-west-1%2Fexecute-
api%2Faws4_request&X-Amz-Date=20200518T122810Z&X-Amz-SignedHeaders=host
host:api.listarobinson.es

host
016068e1847395f1cd2060ce97001f43620f87c2699e48b554a81389d27ddf34
```

Es preciso tener en cuenta que la segunda línea contiene la URL sin protocolo, ni *host* ni parámetros de autenticación. Los parámetros de autenticación van en la tercera línea, correspondiente a la *query string* y que después serán añadidos a la URL final a la que llamar.

¹⁰ https://docs.aws.amazon.com/es_es/general/latest/gr/signature-version-4.html

¹¹ https://docs.aws.amazon.com/es_es/general/latest/gr/aws-general.pdf#signature-version-4

¹² <https://developer.mozilla.org/en-US/docs/Web/HTTP/CORS>

¹³ <https://github.com/adigital-org/slr-client/blob/master/src/util/Signing.js>

¹⁴ <https://cliente.api.listarobinson.es/>

3.2.1.2. Cadena para firmar (*StringToSign*):

```
AWS4-HMAC-SHA256
20200518T122810Z
20200518/eu-west-1/execute-api/aws4_request
ec16349fb8153f20b8777d11f3f7a341fc566969cea6a0af5e8bda17308477ea
```

El cuarto parámetro de la cadena para firmar es el hash de la solicitud canónica.

3.2.1.3. Envío de los parámetros de autenticación:

En este caso, la petición POST fue firmada esperando que los parámetros de autenticación se envíen la URL, por lo tanto, la URL final a la que habría que llamar sería:

```
https://api.listarobinson.es/v1/api/user?X-Amz-Algorithm=AWS4-HMAC-SHA256&X-
Amz-Credential=AKIAXXXXXXXXXXXXXXXXXX%2F20200518%2Feu-west-1%2Fexecute-
api%2Faws4_request&X-Amz-Date=20200518T122810Z&X-Amz-SignedHeaders=host&X-
Amz-
Signature=96425ad1c0ef1824bf7148cfd85cfd4921a55bc0b3fdb5f912150b4bc4c95fa6
```

3.2.2. Con los parámetros de autenticación en las cabeceras

3.2.2.1. Solicitud canónica (*CanonicalRequest*):

```
POST
/v1/api/user

host:api.listarobinson.es

host
016068e1847395f1cd2060ce97001f43620f87c2699e48b554a81389d27ddf34
```

En este caso, la *query string* de la tercera línea está vacía, ya que los parámetros de autenticación van en las cabeceras y no en la URL.

3.2.2.2. Cadena para firmar (*StringToSign*):

```
AWS4-HMAC-SHA256
20200518T122316Z
20200518/eu-west-1/execute-api/aws4_request
654ea9cd4506dc92c7ff2166d251a3e1747062852e1057d1f3b0f7e5b82746fe
```

El cuarto parámetro de la cadena para firmar, es el hash de la solicitud canónica.

3.2.2.3. Envío de los parámetros de autenticación:

En este caso, la petición POST fue firmada esperando que los parámetros de autenticación se envíen la URL, por lo tanto, la URL final a la que habría que llamar sería:

Se requieren dos cabeceras, *AuthorizationHeader*, que contiene el resultado de firmar *StringToSign* con el secreto de la empresa consultar y cuyo valor sería, en este caso:

```
AWS4-HMAC-SHA256 Credential=AKIAXXXXXXXXXXXXXXXXXX/20200518/eu-west-1/execute-
api/aws4_request, SignedHeaders=host,
Signature=d832d636b4f05040283f0e14626b1aea532cf954c6c401ebf8a699b0256425ff
```

Y *X-Amz-Date*, que contiene la fecha utilizada durante el proceso de firma, cuyo valor en este caso sería:

```
20200518T122316Z
```

4. Gestión de errores consultando la API

Cuando como respuesta a una petición a la API se reciba una respuesta distinta a las especificadas en el punto [7 ENDPOINTS](#) (para peticiones POST, HTTP 200, para peticiones GET, HTTP 200 o HTTP 404 con json con campo “signature”), el software de la empresa consultante deberá tener en cuenta las siguientes consideraciones:

- HTTP status code 400:
 - Tipo respuesta:
 - json
 - Respuesta:
 - {"message": "string"}
 - Significado: petición errónea.
 - Acción a tomar: detener el proceso de consulta y revisar las peticiones realizadas, pues la petición es incorrecta, por ejemplo, porque uno de los identificadores hash enviados está vacío, se han solicitado demasiados registros en la petición POST, etc.
- HTTP status code 404 en peticiones POST o en peticiones GET sin campo “signature” en el json de respuesta:
 - Tipo respuesta:
 - json
 - Respuesta:
 - {"message": "string"}
 - Significado: no se encuentra el recurso solicitado.
 - Acción a tomar: revisar las URL de los endpoint a los que se está llamando.
- HTTP status code 429:
 - Tipo respuesta:
 - json
 - Respuesta:
 - {"message": "string"}
 - Significado: no se ha podido procesar la petición por sobrecarga.
 - Acción a tomar: esta petición no se deberá repetir hasta transcurridos, al menos, 15 segundos. Al recibir este error, el software de la entidad deberá reducir la concurrencia (número de llamadas simultáneas) progresivamente de forma temporal, con un mínimo de 15 minutos.
- HTTP status code 403:
 - Tipo respuesta:
 - json
 - Respuesta:
 - {"message": "string"}
 - Significado: error en la autenticación de la petición.
 - Acción a tomar: se deberá interrumpir el proceso de consulta y examinar el proceso de autenticación de las peticiones. En el cuerpo de la respuesta se encuentran los detalles de la denegación de acceso.
- Otros:
 - Tipo respuesta:
 - json
 - Respuesta:

- {"message": "*string*"}
- Significado: variable según el código de respuesta.
- Acción a tomar: cuando el error sea un código distinto a los anteriores, como códigos 500, 502 o 504, es posible mantener el proceso de consulta con la misma concurrencia salvo que la proporción de errores sea elevada (>1/50.000). Para la gestión de estas respuestas, se recomienda incorporar un sistema de reintentos máximos o detener el proceso y realizar un análisis manual a fin de encontrar la naturaleza del problema.

La incorrecta gestión de las respuestas obtenidas por la API podría implicar una limitación o restricción en el acceso a la misma, así como costes adicionales por pérdida de bonificaciones o descuentos.

5. Concurrencia y ejemplos de uso

5.1. Concurrencia

La API del Servicio de Lista Robinson se sirve desde una infraestructura de alta disponibilidad y autoescalable que se adapta a la demanda requerida en cada momento. No obstante, el proceso de escalado no siempre es instantáneo y pueden producirse errores por sobrecarga, por lo que es esencial que cualquier desarrollo incorpore el sistema de gestión de errores detallado en el apartado [4. GESTIÓN DE ERRORES CONSULTANDO LA API](#).

5.2. Ejemplo de uso: depuración con baja concurrencia

Si se desea consultar periódicamente una base de datos extensa, se recomienda evitar el lanzamiento de un gran número de consultas concurrentes durante un breve espacio de tiempo. En su lugar, se recomienda mantener un proceso permanente que realice consultas de forma continuada y con una baja concurrencia.

A modo de ejemplo:

Una entidad tiene una base de datos de 10.000.000 de registros, realiza campañas semanales y necesita depurar la base de datos con la Lista Robinson.

La entidad podría establecer un servicio en segundo plano en uno de sus sistemas que realizase unas 17 consultas por segundo, depurando la base de datos por completo cada 7 días, con lo que el impacto en su infraestructura sería muy reducido.

Si esta empresa deseara realizar menos consultas, podría optar por una carga a 8 peticiones por segundo, depurando la base de datos al completo cada 15 días, lo que seguiría asegurando el cumplimiento de lo establecido en la normativa, que da una validez a la consulta de un (1) mes, a la vez que reduciría aún más el impacto en sus sistemas.

5.3. Ejemplo de uso: depuración con alta concurrencia

Si bien el método de consulta descrito en el apartado anterior es el recomendado por su estabilidad, control y bajo impacto en los sistemas de la entidad consultante, es posible que en ocasiones una entidad necesite consultar un volumen de registros elevado en un breve periodo de tiempo. En estos casos, es posible ajustar el número de peticiones simultáneas para lograr reducir el tiempo de espera.

A modo de ejemplo:

Una entidad necesita consultar una base de datos 5.000.000 de registros para una empresa cliente que desea realizar una gran campaña comercial que arrancará en 7 días.

Si la entidad desea entregar a su cliente la base de datos lista antes de 120 horas (5 días), puede iniciar un proceso de consulta con una concurrencia de unas 12 consultas por segundo.

Si, en cambio, la entidad desea entregar a su cliente la base de datos lista antes de 48 horas, podrá iniciar un proceso de consulta con una concurrencia de unas 29 consultas por segundo.

Si necesitase realizar la entrega antes de 6 horas, podría iniciar un proceso de consulta con una concurrencia de unas 232 consultas por segundo

Para una entrega antes de 1 hora, podría iniciar un proceso de consulta con una concurrencia de aproximadamente 1400 consultas por segundo.

Para una entrega antes de 30 minutos, el proceso podría iniciarse con una concurrencia de aproximadamente 2800 consultas por segundo.

Por lo tanto, el número de peticiones concurrentes se puede incrementar o reducir según las necesidades en cada momento de la entidad. No obstante, en todo caso es esencial implementar correctamente la gestión de errores de acuerdo con lo establecido en el punto [4. GESTIÓN DE ERRORES CONSULTANDO LA API.](#)

6. Datos generales

- Protocolo: https
- Host: api.listarobinson.es
- Versión: v1
- Ejemplo: <https://api.listarobinson.es/v1/api/user>

7. Endpoints

7.1. Consultar si un registro se encuentra en la Lista Robinson

7.1.1. Información

- Dirección: /api/user
- Método: POST
- *Timeout* máximo: 30s
- Peticiones máximas por segundo: 100
- Cuerpo de la petición (payload):
 - Huellas digitales separadas por comas, hasta un máximo de 100 por petición.
- Autenticación: AWS Signature Version 4¹⁵
 - Región: eu-west-1
 - Servicio: execute-api
- Respuesta:
 - HTTP status code:
 - 200
 - Tipo respuesta:
 - json
 - Ejemplo de respuesta:
 - {*"hash"*:{"found":*"bool"*,"signature":*"str"*,"sectors":*"str_array"*}, ... }
 - Significado: los registros han sido consultados. Cada registro se encuentra en una entrada del JSON que tiene por clave la huella digital y su valor es un nuevo JSON con tres claves:
 - *found*: *true* o *false* según si el registro consta o no consta, respectivamente, en la Lista Robinson.
 - *signature*: prueba de consulta.
 - *sectors*: array con los identificadores de los sectores según su código de sector¹⁶. Puede ser un array vacío si el ciudadano no tiene especificado ningún sector publicitario en su perfil.
- Elementos a considerar:
 - El orden de las huellas digitales devuelto puede ser distinto al solicitado.
 - En una misma petición no puede haber dos huellas digitales iguales.
 - Si una de las huellas solicitadas tiene longitud cero (0), se devolverá un error.

¹⁵ https://docs.aws.amazon.com/es_es/general/latest/gr/signature-version-4.html

¹⁶ <https://www.listarobinson.es/sectores-publicitarios-y-sus-codigos-api>

- Si en una petición no hay huellas o hay más de las permitidas, se devolverá un error.

7.1.2. Cómo generar un hash

1. Normalizar los campos del registro a consultar de acuerdo con el Anexo I del Reglamento del Servicio de Lista Robinson.
2. Concatenar los campos correspondientes al registro que se desea consultar en el orden indicado en el siguiente apartado, precedidos por el código que corresponda:
 - a. Para consultas realizadas por DNI/NIF/NIE: "00"¹⁷
 - b. Para consultas realizadas al canal postal: "01"
 - c. Para consultas realizadas al canal correo electrónico: "02"
 - d. Para consultas realizadas al canal SMS, MMS y similares: "03"
 - e. Para consultas realizadas al canal telefónico: "04"
3. Aplicar el algoritmo SHA256 a la cadena del paso anterior y representar en forma de cadena hexadecimal en minúsculas.
4. La huella digital resultante será la que se deba enviar para la consulta de este registro.

7.1.3. Orden de concatenación de los campos para computar la huella digital de cada registro a consultar

- Consulta por número de identificación:
 1. Número identificación
- Consulta por dirección postal:
 1. Nombre
 2. Apellido1
 3. Apellido2
 4. Nombre vía
 5. Número portal
 6. Código postal
 7. Código provincia
- Consulta por correo electrónico:
 1. Correo electrónico
- Consulta por canal telefónico en modalidad completa:
 1. Nombre
 2. Apellido1
 3. Apellido2
 4. Número de teléfono
- Consulta por canal telefónico en modalidad simplificada:
 1. Número de teléfono
- Consulta por canal SMS/MMS y similares en modalidad completa:
 1. Nombre
 2. Apellido1
 3. Apellido2
 4. Número de teléfono
- Consulta por canal SMS/MMS y similares en modalidad simplificada:

¹⁷ La consulta por número de identificación (DNI/NIF/NIE) sirve para todos los canales, de acuerdo con el Reglamento del Servicio de Lista Robinson.

1. Número de teléfono

7.1.4. Preparación de la consulta

Se pueden consultar varios registros mediante una única llamada a la API como se indica en [7.1 CONSULTAR SI UN REGISTRO SE ENCUENTRA EN LA LISTA ROBINSON](#). Este valor podría verse modificado en un futuro, por lo que se recomienda a los implementadores disponer de un sistema que permita ajustar el número de registros por llamada de forma sencilla (por ejemplo, mediante una variable de entorno).

La imputación de las consultas se realizará de acuerdo a lo establecido en el apartado [2.2 CÓMPUTO E IMPUTACIÓN DE CONSULTAS](#).

El cuerpo de la petición debe contener únicamente las huellas digitales de los registros a consultar separados por comas.

Dentro de una misma llamada a la API no puede haber dos huellas digitales iguales, por lo que es necesario suprimir duplicados antes de realizar la llamada. La realización de una llamada con huellas duplicadas producirá un error 400.

Si el número de huellas digitales incluidas es cero (0) o mayor del máximo admitido, o si el tamaño de alguna de las huellas digitales es cero (0), también se generará un error 400.

Ejemplo correcto con tres huellas digitales:

```
ca978112ca1bbdcfac231b39a23dc4da786eff8147c4e72b9807785afee48bb, 3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb73eeaed59c009d, 2e7d2c03a9507ae265ecf5b5356885a53393a2029d241394997265a1a25aefc6
```

Ejemplo con una huella digital duplicada que provocará un error HTTP 400:

```
ca978112ca1bbdcfac231b39a23dc4da786eff8147c4e72b9807785afee48bb, ca978112ca1bbdcfac231b39a23dc4da786eff8147c4e72b9807785afee48bb
```

Ejemplo con una huella digital de tamaño igual a cero (0), que producirá un error HTTP 400:

```
ca978112ca1bbdcfac231b39a23dc4da786eff8147c4e72b9807785afee48bb,, 3e23e8160039594a33894f6564e1b1348bbd7a0088d42c4acb73eeaed59c009d
```

(nótese las dos comas consecutivas sin un valor entre ellas)

7.1.5. Ejemplos

7.1.5.1. Consulta de un único registro al canal telefónico

En la base de datos a procesar:

- Nombre: José
- Apellido 1: Martínez
- Apellido 2: Martínez
- Teléfono: 555.44 43 33

Campos normalizados según el Anexo I del Reglamento del Servicio de Lista Robinson:

- Nombre: jose
- Apellido 1: martinez
- Apellido 2: martinez

La petición y su resultado serían los siguientes:

- Realizar petición POST junto con los parámetros de autenticación indicados en [2.1 AUTENTICACIÓN Y OBTENCIÓN DE CLAVES](#) a la siguiente URL y con el siguiente contenido:
 - URL: `https://api.listarobinson.es/v1/api/user`
 - *Payload:*
`0e49dec1f67efda3811000e8de73bb9522eeef1178ec9c549416c1cbfbc1864a,998202202f5b7a2027d8acb1775419f7c9cb684f1889ff559a622b0efe9d0ba3,aca5e366bf9d236e6889b08e931b0723bc26985ad4b98879cbbe379106403be`
- Respuesta recibida:
 - HTTP status code: 200
 - *Payload:*
`{"0e49dec1f67efda3811000e8de73bb9522eeef1178ec9c549416c1cbfbc1864a":{"found":true,"signature":"MTU30TAyMjk4NzA2ODowZTQ5ZGVjMwY2N2VmZGEzODExMDAwZThkZTczYmI5NTIyZWVlZjExNzhIYzljNTQ5NDE2YzFjYmZiYzE4NjRhOlNFQzMsU0VDMjoyNDgyOWNiMDhlYjFhNDI3ZGEyNGJhZTE2NWYyNmM0ZTY1NDBhMDI1NTdlOWVmMjUxNGFiYzQxMDgxMmI2ODlj","sectors":["SEC2","SEC3"]},"998202202f5b7a2027d8acb1775419f7c9cb684f1889ff559a622b0efe9d0ba3":{"found":false,"signature":"MTU30TAyMjk4NzA2ODo5OTgyMDIyMDJmNWl3YTIwMjJkOGFjYjE3NzU0MTlmN2M5Y2I2ODRmMTg4OWZmNTU5YTlYmMlWZWZlOWQwYmEzOjphZWZjZjMxMWNlYjE2YWwYmRlMzA5YzZkMGExNTYyYzFkMGM1NGY3M2MxMDgyYmMyY2YyZTFjZGJhODZhNDA3","sector s":[]},"aca5e366bf9d236e6889b08e931b0723bc26985ad4b98879cbbe379106403b":{"found":false,"signature":"MTU30TAyMjk4NzA2ODphY2FhNWUzNjZiZjlkMjZTY4ODliMDhlOTMxYjA3MjNiYzI2OTg1YWQ0Yjk4ODc5Y2JiZTM3OTEwNjQwM2I6Ojc0NjRlYzY2M2NzhmYTUxNzY4MjI5N2JjZDRiMTNiYTcwZWRiNTMyZTI2YmE2MWU5MDg0NGVlYjU4YjY0MTcwM2U=","sectors":[]}}`

```
{
  "0e49dec1f67efda3811000e8de73bb9522eeef1178ec9c549416c1cbfbc1864a": {
    "found": true,
    "signature":
    "MTU30TAyMjk4NzA2ODowZTQ5ZGVjMwY2N2VmZGEzODExMDAwZThkZTczYmI5NTIyZWVlZjExNzhIYzljNTQ5NDE2YzFjYmZiYzE4NjRhOlNFQzMsU0VDMjoyNDgyOWNiMDhlYjFhNDI3ZGEyNGJhZTE2NWYyNmM0ZTY1NDBhMDI1NTdlOWVmMjUxNGFiYzQxMDgxMmI2ODlj",
    "sectors": ["2","3"]
  },
  "998202202f5b7a2027d8acb1775419f7c9cb684f1889ff559a622b0efe9d0ba3": {
    "found": true,
    "signature":
    "MTU30TAyMjUwMjc4NjU0OTgyMDIyMDJmNWl3YTIwMjJkOGFjYjE3NzU0MTlmN2M5Y2I2ODRmMTg4OWZmNTU5YTlYmMlWZWZlOWQwYmEzOjphZWZjZjMxMWNlYjE2YWwYmRlMzA5YzZkMGExNTYyYzFkMGM1NGY3M2MxMDgyYmMyY2YyZTFjZGJhODZhNDA3",
    "sectors": ["8","7","6","5","4","3","2","1"]
  },
  "aca5e366bf9d236e6889b08e931b0723bc26985ad4b98879cbbe379106403b": {
    "found": false,
    "signature":
    "MTU30TAyMjk4NzA2ODphY2FhNWUzNjZiZjlkMjZTY4ODliMDhlOTMxYjA3MjNiYzI2OTg1YWQ0Yjk4ODc5Y2JiZTM3OTEwNjQwM2I6Ojc0NjRlYzY2M2NzhmYTUxNzY4MjI5N2JjZDRiMTNiYTcwZWRiNTMyZTI2YmE2MWU5MDg0NGVlYjU4YjY0MTcwM2U=",
    "sectors": []
  }
}
```

En este caso, el JSON de respuesta contiene tres entradas, una por cada una de las huellas solicitadas.

7.1.5.3. Cómputo de la huella digital de una dirección de correo electrónico

En la base de datos a procesar:

- Correo electrónico: Ejemplo_Testing123@listarobinson.net

Campos normalizados según el Anexo I del Reglamento del Servicio de Lista Robinson:

- Correo electrónico: ejemplo_testing123@listarobinson.net

Campo concatenado con el código de consulta:

- 02ejemplo_testing123@listarobinson.net

Cómputo de la huella digital mediante SHA256:

- 45d6758764e9c0925283b6dc3aab97b344314ec2492473fae3d6a73d523539f9

7.2. Obtener cuántas consultas se han realizado a través de la API

7.2.1. Información

- Dirección: /api/requests
- Método: GET
- *Timeout* máximo: 30s
- Conexiones máximas simultáneas: 1
- Autenticación: AWS Signature Version 4¹⁸
 - Región: eu-west-1
 - Servicio: execute-api
- Respuestas:
 - Positiva:
 - HTTP status code:
 - 200
 - Tipo respuesta:
 - json
 - Respuesta:
 - {"requests": "*number*", "timestamp": "*number*"}
 - Significado: el número indicado es la cantidad de consultas que la entidad consultante tenía realizadas e imputadas a la suscripción vigente y activa cuando se ejecutó el último recuento¹⁹. Las suscripciones al Servicio tienen una duración ordinaria anual, de 1 de enero a 31 de diciembre.
 - Negativa:
 - HTTP status code:
 - 404
 - Tipo respuesta:
 - json
 - Respuesta:
 - {"message": "No se ha encontrado coincidencia", "timestamp": "*number*"}
 - Significado: aún no se han computado consultas para la entidad consultante, ya que no tenía claves API cuando se ejecutó el último recuento.

¹⁸ https://docs.aws.amazon.com/es_es/general/latest/gr/signature-version-4.html

¹⁹ El recuento se realiza una vez al día.

8. Endpoints obsoletos

8.1. Acerca de estos endpoint

Los siguientes *endpoints* se encuentran disponibles en la API del Servicio de Lista Robinson en modo de mantenimiento y únicamente para garantizar la compatibilidad de la misma con desarrollos de acuerdo con anteriores versiones de esta documentación. Se trata, por lo tanto, de *endpoints* heredados y, aunque siguen siendo plenamente funcionales, no se recomienda que los nuevos desarrollos los implementen.

8.2. [OBSOLETO] Consultar si un registro se encuentra en la Lista Robinson

8.2.1. Motivo obsolescencia

Este *endpoint* ha sido sustituido por la versión POST del mismo, descrita en [7.1 CONSULTAR SI UN REGISTRO SE ENCUENTRA EN LA LISTA ROBINSON](#) y con capacidad para consultar varios registros por llamada, obteniendo un ahorro de recursos y una mayor eficiencia.

8.2.2. Información

- Dirección: /api/user/{hash}
- Método: GET
- Peticiones máximas por segundo: 250
- Autenticación: AWS Signature Version 4²⁰
 - Región: eu-west-1
 - Servicio: execute-api
- Respuestas:
 - Positiva:
 - HTTP status code:
 - 200
 - Tipo respuesta:
 - json
 - Ejemplo de respuesta:
 - {"signature": "*string*", "sectors": "*string_array*"}
 - Significado: el registro consultado se encuentra en la Lista Robinson asociado a los sectores publicitarios indicados. Los sectores se representan según su [código de sector](#)²¹. El campo "sectors" tendrá por valor un array vacío ([]) o valor nulo ("null") si el registro consultado no tiene ningún sector asociado.
 - Negativa:
 - HTTP status code:
 - 404

²⁰ https://docs.aws.amazon.com/es_es/general/latest/gr/signature-version-4.html

²¹ <https://www.listarobinson.es/sectores-publicitarios-y-sus-codigos-api>

- Tipo respuesta:
 - json
- Respuesta:
 - {"signature": "*string*"}
- Significado: el registro consultado no se encuentra en la Lista Robinson.

8.2.3. Cómo generar un hash

5. Normalizar los campos del registro a consultar de acuerdo con el Anexo I del Reglamento del Servicio de Lista Robinson.
6. Concatenar los campos correspondientes al registro que se desea consultar en el orden indicado en el siguiente apartado, precedidos por el código que corresponda:
 - a. Para consultas realizadas por DNI/NIF/NIE: "00"²²
 - b. Para consultas realizadas al canal postal: "01"
 - c. Para consultas realizadas al canal correo electrónico: "02"
 - d. Para consultas realizadas al canal SMS, MMS y similares: "03"
 - e. Para consultas realizadas al canal telefónico: "04"
7. Aplicar el algoritmo SHA256 a la cadena del paso anterior y representar en forma de cadena hexadecimal en minúsculas.
8. La huella digital resultante será la que se deba enviar mediante la petición GET.

8.2.4. Orden de concatenación de campos

- Consulta por número de identificación:
 1. Número identificación
- Consulta por dirección postal:
 1. Nombre
 2. Apellido1
 3. Apellido2
 4. Nombre vía
 5. Número portal
 6. Código postal
 7. Código provincia
- Consulta por correo electrónico:
 1. Correo electrónico
- Consulta por canal telefónico en modalidad completa:
 1. Nombre
 2. Apellido1
 3. Apellido2
 4. Número de teléfono
- Consulta por canal telefónico en modalidad simplificada:
 1. Número de teléfono
- Consulta por canal SMS/MMS y similares en modalidad completa:
 1. Nombre
 2. Apellido1
 3. Apellido2

²² La consulta por número de identificación (DNI/NIF/NIE) sirve para todos los canales, de acuerdo con el Reglamento del Servicio de Lista Robinson.

4. Número de teléfono

- Consulta por canal SMS/MMS y similares en modalidad simplificada:
 1. Número de teléfono

8.2.5. Ejemplos

8.2.5.1. Consulta al canal telefónico con coincidencia

En la base de datos a procesar:

- Nombre: José
- Apellido 1: Martínez
- Apellido 2: Martínez
- Teléfono: 555.44 43 33

Campos normalizados según el Anexo I del Reglamento del Servicio de Lista Robinson:

- Nombre: jose
- Apellido 1: martinez
- Apellido 2: martinez
- Teléfono: 0034555444333

Campos concatenados:

- *josemartinezmartinez0034555444333*

Campos concatenados con el código de consulta:

- *04josemartinezmartinez0034555444333*

Cómputo de la huella digital mediante SHA256:

- 998202202f5b7a2027d8acb1775419f7c9cb684f1889ff559a622b0efe9d0ba3

Resultado:

- Realizar petición GET junto con las cabeceras de autenticación a la siguiente url:
 - <https://api.listarobinson.es/v1/api/user/998202202f5b7a2027d8acb1775419f7c9cb684f1889ff559a622b0efe9d0ba3>
- Respuesta recibida:
 - HTTP status code: 200
 - {"signature":"MTU0OTQ1NjA0Njg3NToxLDIsMyw0LDYsNyw4Ojk5ODIwMjIwMmY1YjdhMjAyN2Q4YWNiMTc3NTQxOWY3YzljYjY4NGYxODg5ZmY1NTIhNjlyYjBlZmU5ZDBiYTM6ZjVIMzAzMTdjZDU4ZGM5OTMzYTc4ZmMzM2FiOWQ5ZDdmYjJlODMwNzVmMjU4ZmE4OGM1OGNhODA1ZjhmYTU1Ng==","sectors":["1","2","3","4","6","7","8"]}
- Interpretación: el registro se encuentra en la Lista Robinson asociado a los sectores indicados.

8.2.5.2. Consulta al canal correo electrónico sin coincidencia

En la base de datos a procesar:

- Correo electrónico: Ejemplo_&Testing123@listarobinson.net

Campos normalizados según el Anexo I del Reglamento del Servicio de Lista Robinson:

- Correo electrónico: ejemplo_testing123@listarobinson.net

Campo concatenado con el código de consulta:

- *02ejemplo_testing123@listarobinson.net*

Cómputo de la huella digital mediante SHA256:

- 45d6758764e9c0925283b6dc3aab97b344314ec2492473fae3d6a73d523539f9

Resultado:

- Realizar petición GET junto con las cabeceras de autenticación a la siguiente url:
 - <https://api.listarobinson.es/v1/api/user/45d6758764e9c0925283b6dc3aab97b344314ec2492473fae3d6a73d523539f9>
- Respuesta recibida:
 - HTTP status code: 404
 - {"signature":"MTU0OTQ1NjA0Njg4OTo6NDVkJjc1ODc2NGU5YzA5MjUyODNiNmRjM2FhYjk3YjM0NDMxNGVjMjQ5MjQ3M2ZhZTNkNmE3M2Q1MjM1MzlmOTo0NzcxN2lyYmRhZjllOGQyYzI0YzRiYWxMGU5NTFjNjNjNTIiY2VkNmExYzQ2NDdlnjU0MzA0ZmVhZmNmZDU2"}
- Interpretación: el registro no se encuentra en la Lista Robinson.