

SensIA - Servicio web de anonimización

Manual del integrador

Versión 1.0 · Aplicable a SensIA 1.0.0-rc.1

Índice

- Control de versiones
- 1. Objeto del documento
- 2. Descripción general
  - 2.1. Flujo de integración habitual
- 3. Requisitos, seguridad y límites
  - 3.1. Requisitos del sistema cliente
  - 3.2. Dirección del servicio
  - 3.3. Seguridad del transporte
  - 3.4. Autenticación de aplicación
  - 3.5. Límites de uso
- 4. Contrato del servicio
  - 4.1. Datos generales
  - 4.2. Convenciones de los mensajes
  - 4.3. Coordenadas
- 5. Opciones de procesamiento
  - 5.1. Vista previa
  - 5.2. Detección en documentos (DocumentRedactionOptions)
  - 5.3. Reconocimiento de entidades nombradas (NER)
  - 5.4. Seudonimización
  - 5.5. Rostros (FaceMaskingOptions)
  - 5.6. Detector de rostros (YunetOptions)
  - 5.7. Códigos QR y de barras (QROptions)
- 6. Operaciones
  - 6.1. submitBatch
    - Petición (submitBatchRequest)
    - Respuesta (submitBatchResponse)
    - Ejemplo
  - 6.2. getStatus
    - Petición (getStatusRequest)
    - Respuesta (getStatusResponse)
    - Ejemplo
  - 6.3. redactImage
    - Petición (redactImageRequest)
    - Respuesta (redactImageResponse)
    - Ejemplo
  - 6.4. extractImages
    - Petición (extractImagesRequest)
    - Respuesta (extractImagesResponse)
    - Ejemplo
- 7. Revisión y confirmación de marcas
- 8. Gestión de errores

- 9. Referencia de tipos y enumerados
  - 9.1. Estados del documento (ProcessStateEnum)
  - 9.2. Tipos de detección (DetectionTypeEnum)
  - 9.3. Acciones (ActionStateEnum)
  - 9.4. Resultado de la operación (StatusEnum)
  - 9.5. Detección (RedactionItem)
  - 9.6. Otros tipos
- 10. Buenas prácticas y protección de datos

Control de versiones

Versión	Fecha	Descripción de los cambios
1.0	01/10/2026	Primera edición pública, aplicable a SensIA 1.0.0-rc.1.

1. Objeto del documento

Este documento describe la integración de sistemas externos con el servicio web SOAP de SensIA para la anonimización automatizada de documentos PDF e imágenes. Está dirigido al personal de desarrollo y de arquitectura de software de las Administraciones Públicas, o de otras entidades, que desee incorporar SensIA a sus aplicaciones de gestión documental, sedes electrónicas o procesos internos.

El documento recoge el contrato del servicio, las opciones de procesamiento y la descripción de cada operación, con ejemplos de mensajes.

El uso de la aplicación web se describe en el *Manual de usuario*, y la instalación y configuración del servidor, en el *Manual técnico y de despliegue*.

2. Descripción general

SensIA ofrece un servicio web SOAP 1.1, de estilo *document/literal*, con cuatro operaciones:

Operación	Modo	Finalidad
`submitBatch`	Asíncrono	Envía uno o varios documentos PDF para su anonimización. Devuelve un identificador de seguimiento por documento.
`getStatus`	Consulta	Devuelve el estado de un documento enviado. Cuando el procesamiento ha finalizado, devuelve también el documento resultante y el detalle de las detecciones.
`redactImage`	Síncrono	Detecta y oculta rostros en una imagen JPG o PNG, y devuelve el resultado en la misma respuesta.
`extractImages`	Síncrono	Extrae las imágenes incrustadas en un PDF, sin anonimizarlas.

El procesamiento de PDF es asíncrono porque el OCR y el análisis de imágenes pueden requerir tiempos variables según el tamaño y el tipo de documento.

Todo el procesamiento se realiza en los servidores de la instalación, sin enviar los documentos a servicios externos.

2.1. Flujo de integración habitual

1. El sistema cliente envía los documentos con `submitBatch` y guarda el identificador de seguimiento (`serverTicketId`) de cada documento, asociado a su identificador propio (`clientId`).
2. Consulta periódicamente cada documento con `getStatus` hasta que alcanza un estado final (apartado 9.1).
3. Si el estado es `COMPLETED` , recupera el documento anonimizado (`processedDocument`) y el detalle de las detecciones (`redactionResults`). Debe guardarlos de inmediato, porque el resultado solo puede recuperarse una vez (apartado 3.5).

Si una persona debe revisar las detecciones antes de generar el documento definitivo, se utiliza el flujo de revisión descrito en el apartado 7. Las imágenes sueltas se procesan directamente con `redactImage` .

3. Requisitos, seguridad y límites

3.1. Requisitos del sistema cliente

- **Cliente de servicios web** compatible con SOAP 1.1. Debe admitir las cabeceras SOAP declaradas en el enlace (*binding*) del WSDL o permitir añadirlas manualmente.
- **Codificación Base64** del contenido de los ficheros (PDF, JPG o PNG).
- **Conexión HTTPS** con el servidor de SensIA, con TLS 1.2 o superior y verificación del certificado del servidor.
- **Token de autenticación**, facilitado por el administrador de la instalación.
- **Certificado de cliente** en formato PKCS#12, junto con su contraseña. Solo es necesario si se utiliza la vía con certificado (apartado 3.3).

3.2. Dirección del servicio

En una instalación estándar, el servicio se publica a través del proxy HTTPS de la instalación, con dos vías de acceso:

Vía	Dirección	Autenticación
Puerto 443 (cerrada, con código 403, si la instalación exige certificado de cliente)	`https://sensia.tuorganismo.es/services/SensiaService`	Token
Puerto 8443, con certificado de cliente (solo si la instalación la tiene activada)	`https://sensia.tuorganismo.es:8443/services/SensiaService`	Certificado de cliente autorizado y token

El token se exige siempre, en las dos vías.

Los puertos pueden variar en cada instalación; el administrador facilitará las direcciones aplicables.

El contrato WSDL se obtiene añadiendo `?wsdl` a la dirección del servicio. El WSDL publicado anuncia la dirección HTTPS de la instalación. En cambio, el fichero `service.wsdl` del repositorio contiene una dirección de desarrollo (`http://localhost:8080/SensiaService`) que no debe utilizarse. Por ello, se recomienda que el sistema cliente configure explícitamente la dirección del servicio, sin tomarla del WSDL.

El servicio no se atiende por HTTP: las peticiones sin cifrar se rechazan con el código 403. Los puertos internos del motor no son accesibles desde el exterior. En entornos de desarrollo o de pruebas, la dirección la facilita el equipo técnico responsable de la instalación.

3.3. Seguridad del transporte

Las comunicaciones se realizan siempre por HTTPS, con TLS 1.2 o 1.3. El sistema cliente debe verificar el certificado del servidor y no debe desactivar esa verificación en ningún entorno.

Si la instalación utiliza un certificado emitido por una autoridad de certificación interna, por ejemplo la CA interna de SensIA (`ca_interna.crt`), el certificado de esa autoridad debe añadirse al almacén de confianza del cliente.

La **vía con certificado de cliente** (mTLS) es opcional y la activa el administrador de la instalación. Sus características son las siguientes:

- Utiliza un puerto propio, por defecto el 8443. Cuando está activada, el servicio deja de atenderse por el puerto 443, que responde con el código 403.
- El cliente presenta durante la conexión un certificado de cliente cuya huella SHA-256 debe estar registrada en el servidor.
- El certificado lo valida directamente el motor de SensIA; el proxy no descifra la conexión.
- Cada petición debe incluir, además, el token (apartado 3.4).

En cuanto al certificado de cliente:

- **Entrega.** El administrador entrega el certificado en formato PKCS#12 (`.p12`) y su contraseña por canales distintos. La mayoría de plataformas utilizan directamente este formato. Si se necesita otro, puede convertirse; por ejemplo, a JKS con la utilidad `keytool` de Java.
- **Certificados del organismo.** El organismo también puede autorizar certificados emitidos por su propia autoridad de certificación, si el administrador la ha configurado en la instalación.
- **Validez y renovación.** Los certificados emitidos por la instalación tienen una validez de 365 días. Conviene coordinar su renovación con el administrador con antelación suficiente.

3.4. Autenticación de aplicación

Todas las operaciones requieren la cabecera SOAP `AuthHeader` con el token de acceso de la instalación (`SOAP_API_TOKEN`):

```
<soapenv:Header>
   <sen:AuthHeader xmlns:sen="http://sensia.github.io/">
      <token>[TOKEN]</token>
      <cliente>IDENTIFICADOR_DEL_USUARIO_O_APLICACION</cliente>
   </sen:AuthHeader>
</soapenv:Header>
```

El elemento `AuthHeader` pertenece al espacio de nombres del servicio. El elemento `token` se escribe sin prefijo (apartado 4.2).

Si el token no se incluye o no es válido, el servicio devuelve un SOAP Fault con el código `AUTH_ERROR`.

En la versión actual, el token es común a todos los sistemas integradores de la instalación. Su renovación afecta a todos ellos, por lo que el administrador la comunicará con antelación, salvo que deba sustituirse de forma inmediata por haber quedado expuesto.

El token es una credencial: debe almacenarse de forma segura y no debe incluirse en el código fuente ni registrarse en los logs del sistema cliente.

cliente es obligatorio (1 a 128 caracteres). Cada cliente puede tener como máximo 2 documentos en proceso, y un lote (submitBatch) admite como máximo 2 documentos. Si se supera, la respuesta lleva cause = SATURADO y se puede reintentar más tarde.

3.5. Límites de uso

Los siguientes valores corresponden a la configuración estándar de la instalación. Pueden variar en cada instalación, por lo que conviene confirmarlos con el administrador:

Límite	Valor por defecto	Consideraciones
Tamaño máximo de la petición	70 MB	Incluye todos los documentos del lote, codificados en Base64. La codificación aumenta el tamaño en torno a un 33 %, por lo que el contenido original admitido por petición es de unos 50 MB.
Documentos en cola	12	Límite común a todos los integradores y a la aplicación web. Si un lote no cabe en las plazas libres, se rechaza completo con `status/result` `ERROR` («Servicio saturado») y debe reintentarse más tarde o dividirse.
Peticiones simultáneas	8	Las peticiones que superan el límite esperan en cola hasta 30 segundos. Si no pueden atenderse, el servidor responde con HTTP 503 y la petición debe reintentarse más tarde, con una espera creciente.
Conservación de resultados	Una sola recuperación	Tras una consulta `getStatus` con estado `COMPLETED`, el resultado se elimina del servidor y las consultas posteriores devuelven `NOT_FOUND`. Los resultados que no se recogen se eliminan al finalizar el periodo de retención del servidor.

4. Contrato del servicio

4.1. Datos generales

Elemento	Valor
Protocolo	SOAP 1.1, estilo *document/literal*
Espacio de nombres	`http://sensia.github.io/`
Servicio	`SensiaService`
Puerto	`SensiaServicePort`
Binding	`SensiaServiceSOAPBinding`
Cabecera HTTP `Content-Type`	`text/xml; charset=utf-8`

Cada operación tiene su propio valor de `SOAPAction`, que se envía en la cabecera HTTP del mismo nombre:

Operación	SOAPAction
`submitBatch`	`http://sensia.github.io/SensiaService/submitBatch`
`getStatus`	`http://sensia.github.io/SensiaService/getStatus`
`redactImage`	`http://sensia.github.io/SensiaService/redactImage`
`extractImages`	`http://sensia.github.io/SensiaService/extractImages`

4.2. Convenciones de los mensajes

- **Espacios de nombres.** Solo pertenecen al espacio de nombres del servicio:
  - el elemento raíz de cada operación (por ejemplo, `submitBatchRequest`);
  - la cabecera `AuthHeader`;
  - el detalle de error `ServiceFault`.

  Los elementos hijos se escriben sin prefijo, como en los ejemplos de este manual. Los clientes generados a partir del WSDL aplican esta regla automáticamente.
- **Orden de los elementos.** Los elementos deben enviarse en el orden definido en el esquema, que es el de las tablas de este manual.

- **Contenido de ficheros.** El contenido se envía en Base64, preferiblemente sin saltos de línea, en el elemento `fileContent`. Le acompañan el nombre (`fileName`) y el tipo MIME (`contentType`), que puede ser `application/pdf`, `image/jpeg` o `image/png`.
- **Elementos opcionales.** Si se omite una opción, se aplica el valor configurado en el servidor. Los valores por defecto que declara el WSDL (atributo `default`) son orientativos y pueden no coincidir con esa configuración. En las tablas de este manual, la columna «Por defecto» indica el valor de la configuración estándar del servidor.
- **Bloques con elementos obligatorios.** Si se incluye un bloque `yunetOptions`, `qrOptions` o `Coordinate`, deben informarse todos sus elementos.
- **Caracteres especiales.** Los mensajes se codifican en UTF-8. Los caracteres `&` y `<` de los valores de texto, por ejemplo en las listas de palabras, deben escaparse como `&` y `<`.
- **Respuestas.** Los prefijos de espacio de nombres de las respuestas pueden variar, por lo que el sistema cliente no debe depender de ellos. Los textos del elemento `message` que aparecen en los ejemplos son orientativos.

4.3. Coordenadas

Las zonas se expresan con el tipo `Coordinate`, formado por cuatro valores decimales:

- `x` e `y`: posición de la zona;
- `w`: ancho;
- `h`: alto.

En los documentos PDF, los valores se expresan en puntos (1/72 de pulgada), la unidad de medida de las páginas PDF, con origen en la esquina inferior izquierda de la página: `x` crece hacia la derecha e `y` hacia arriba, y el punto (`x`, `y`) corresponde a la esquina inferior izquierda de la zona. Como referencia, una página A4 mide 595 × 842 puntos; la zona del logotipo de la configuración estándar (`0, 700, 200, 142`) ocupa una franja de 200 × 142 puntos en la esquina superior izquierda de una página A4.

Para definir zonas fijas o de exclusión, se recomienda tomar como referencia las coordenadas que devuelve el servicio en `redactionResults` al procesar con vista previa un documento de la misma plantilla. En la operación `redactImage`, las coordenadas se refieren a la propia imagen.

Las zonas definidas en las opciones de procesamiento no incluyen número de página, por lo que se aplican a todas las páginas del documento. En los resultados, el campo `page` indica la página de cada detección.

5. Opciones de procesamiento

El elemento `options` (tipo `ProcessingOptions`) define las reglas que se aplican a todos los documentos de un lote. Todos sus elementos son opcionales:

Elemento	Tipo	Descripción
`documentOptions`	`DocumentRedactionOptions`	Detección sobre el contenido del PDF (apartados 5.2 a 5.4).
`faceOptions`	`FaceMaskingOptions`	Detección y ocultación de rostros (5.5).
`yunetOptions`	`YunetOptions`	Parámetros del detector de rostros (5.6).
`qrOptions`	`QROptions`	Parámetros de detección de códigos QR y de barras (5.7).
`redactionPreview`	boolean	Si es `true`, genera una vista previa en lugar del documento definitivo (5.1). Por defecto, `false`.
`reconstructionMode`	boolean	Si es `true`, genera el documento definitivo aplicando las marcas confirmadas en `confirmedRedactions`, sin repetir el análisis (apartado 7).

Los valores por defecto indicados en este apartado corresponden a la configuración estándar del servidor y pueden variar en cada instalación.

5.1. Vista previa

Con `redactionPreview` a `true`, el motor no aplica la anonimización definitiva. Devuelve el documento con las detecciones marcadas mediante recuadros de color, junto con la lista de detecciones:

- en rojo, los elementos que se ocultarían;
- en verde, las excepciones y zonas protegidas;
- con un aspa roja, las imágenes que se eliminarían.

La vista previa está pensada para la revisión descrita en el apartado 7. También está disponible en la operación `redactImage`.

El documento de vista previa no está anonimizado. Conserva todo el contenido original y no debe publicarse ni almacenarse como documento anonimizado.

5.2. Detección en documentos (DocumentRedactionOptions)

La detección de datos con estructura conocida se realiza siempre, con los patrones configurados en el servidor. En la configuración estándar, se detectan:

- documentos de identidad (NIF, NIE y CIF);
- tarjetas bancarias (Visa, Mastercard y American Express) y cuentas IBAN;
- referencias catastrales y direcciones postales;
- códigos seguros de verificación (CSV) e identificadores únicos (UUID);
- correos electrónicos y teléfonos;
- bastidores y matrículas de vehículos;
- códigos de barras de pago de 42 dígitos.

Otros patrones, como el número de afiliación a la Seguridad Social, los códigos postales o los importes, se incluyen desactivados en la configuración estándar. Solo se detectan en las instalaciones que los hayan activado.

Las opciones de este bloque activan detecciones adicionales y definen excepciones y zonas:

Elemento	Tipo	Por defecto	Descripción
`redactQRCodes`	boolean	`false`	Detecta y oculta códigos QR y códigos de barras (5.7).
`redactNames`	boolean	`false`	Detecta nombres y apellidos mediante diccionario.
`redactNERNames`	boolean	`false`	Detecta personas mediante reconocimiento de entidades (5.3).
`redactNEROrg`	boolean	`false`	Detecta organizaciones (5.3).
`redactNERLoc`	boolean	`false`	Detecta lugares (5.3).
`redactNERMisc`	boolean	`false`	Detecta otras entidades (5.3).
`redactNERScore`	float	`0.65`	Umbral de confianza del reconocimiento de entidades, entre 0 y 1 (5.3).
`removeImages`	boolean	`false`	Elimina las imágenes incrustadas, salvo las situadas en zonas de exclusión y las que ocupan casi toda la página, como fondos o páginas escaneadas.
`preserveLogo`	boolean	`true`	Protege la zona del logotipo definida en `logoExclusionZone`.

`logoExclusionZone`	`Coordinate`	`0, 700, 200, 142`	Zona del logotipo institucional. Su contenido no se oculta ni se elimina.
`imageExclusionZones`	`CoordinatesList`	—	Zonas en las que no se eliminan imágenes.
`fixedRedactionZones`	`CoordinatesList`	—	Zonas que se cubren siempre con cajas blancas en todas las páginas, con independencia de su contenido (por ejemplo, áreas de firma).
`allowedWords`	`WordList`	—	Lista blanca de la petición (véase a continuación).
`blockedWords`	`WordList`	—	Lista negra de la petición (véase a continuación).
`redactionThickness`	int	`2`	Grosor del tachado. Amplía el margen de cada caja para cubrir completamente los caracteres.
`usePseudonyms`	boolean	`false`	Sustituye los datos detectados por seudónimos (5.4).

Lista blanca (`allowedWords`). Contiene términos que no se ocultan aunque coincidan con un criterio de detección. Sus reglas son las siguientes:

- Cada término se envía en un elemento `words`.
- Las entradas literales se comparan sin distinguir mayúsculas ni acentos.
- Pueden usarse expresiones regulares, con sintaxis de Java, anteponiendo el prefijo `regex:`.
- Las detecciones excluidas se devuelven con la acción `WHITELISTED`.

La lista de la petición se aplica además de la lista blanca global del servidor. La lista global de la configuración estándar excluye, entre otros:

- los cargos públicos;
- las fechas en formato largo (por ejemplo, «1 de enero de 2026»);
- los NIF de entidades públicas (los que empiezan por P, Q o S);
- las denominaciones de órganos (consejo, junta, pleno, etc.).

Cada organismo adapta esta lista a sus necesidades, por lo que conviene consultar su contenido al administrador.

Lista negra (`blockedWords`). Contiene términos que se ocultan en todo el documento. Las entradas se tratan como texto literal, no como expresiones regulares. Se buscan como palabras completas y sin distinguir mayúsculas y minúsculas.

```
<allowedWords>
   <words>Ayuntamiento de Ejemplo</words>
   <words>regex:(?i)jefe de servicio</words>
</allowedWords>
<blockedWords>
   <words>confidencial</words>
</blockedWords>
```

5.3. Reconocimiento de entidades nombradas (NER)

El reconocimiento de entidades complementa la detección por diccionario con el modelo de lenguaje público `BSC-LT/roberta_model_for_anonimization`, del Barcelona Supercomputing Center (BSC-CNS). Este modelo identifica en el texto personas, organizaciones, lugares y otras entidades, y está orientado al castellano. En la instalación estándar, el servicio de reconocimiento de entidades se despliega siempre; su uso se activa en cada petición.

Cada categoría se activa por separado:

Opción	Categoría	Valor de `subType` en los resultados
`redactNERNames`	Personas	`PER`
`redactNEROrg`	Organizaciones	`ORG`
`redactNERLoc`	Lugares	`LOC`
`redactNERMisc`	Otras entidades	`OTH`

Las detecciones se devuelven con el tipo `NAMED_ENTITY` y la categoría en el campo `subType`.

El umbral `redactNERScore`, entre 0 y 1, fija la confianza mínima para aceptar una entidad; por defecto es `0.65`. Un valor más bajo aumenta las detecciones y también los falsos positivos. Un valor más alto los reduce, a cambio de más omisiones.

Los nombres de organizaciones y lugares suelen corresponder a información pública, como el nombre del propio organismo. Si se activan estas categorías, conviene añadir esos términos a la lista blanca.

Si se solicita el reconocimiento de entidades y el servicio no está disponible, el documento termina con el estado `ERROR` (apartado 9.1).

5.4. Seudonimización

Con `usePseudonyms` a `true`, los datos detectados se sustituyen por un seudónimo en lugar de ocultarse con una caja opaca. Su funcionamiento es el siguiente:

- El dato se cubre con una caja blanca y el seudónimo se escribe encima.

- Un mismo dato recibe el mismo seudónimo en todo el documento.
- El seudónimo asignado se devuelve en el campo `pseudonym` de cada detección.
- Solo se aplica a texto; las imágenes, rostros y códigos se tratan de la forma habitual.

Conforme al Reglamento General de Protección de Datos (RGPD), la seudonimización no equivale a anonimización: la información seudonimizada puede seguir considerándose dato personal.

5.5. Rostros (FaceMaskingOptions)

Estas opciones se aplican a las imágenes incrustadas en los PDF y a las imágenes enviadas con `redactImage`. En la configuración estándar, la ocultación de rostros está activada; para desactivarla, debe enviarse `maskFaces` con valor `false`.

Elemento	Tipo	Por defecto	Descripción
`maskFaces`	boolean	`true`	Activa la detección y ocultación de rostros.
`minFaceAspectRatio` / `maxFaceAspectRatio`	float	`0`	Proporción mínima y máxima entre ancho y alto de la detección. El valor `0` desactiva cada límite.
`minFaceArea` / `maxFaceArea`	float	`0`	Área mínima y máxima, en píxeles de la imagen enviada, de los rostros que se procesan. El valor `0` desactiva cada límite.
`pixelationIntensity`	int	`70`	Intensidad de la ocultación, de 0 a 100: mosaico de 16 bloques (0) a 4 bloques (99) en el lado mayor del rostro y, con 100, relleno de color sólido. En los rostros pequeños los bloques tienen un tamaño mínimo, por lo que el mosaico tiene menos; los muy pequeños se rellenan de color. Se oculta también un margen del 20 % alrededor del rostro.

Los filtros permiten descartar detecciones de tamaño o forma anómalos. Los rostros descartados por un filtro no se ocultan ni se incluyen en `redactionResults`; por ello, un área mínima elevada puede dejar sin ocultar los rostros de personas que aparecen en segundo plano. Para documentos que se van a publicar se recomienda `pixelationIntensity` mayor de `70`.

5.6. Detector de rostros (YunetOptions)

Este bloque contiene los parámetros del modelo de detección facial YuNet. Si se incluye, deben informarse sus tres elementos. Se recomienda omitirlo y mantener la configuración del servidor, salvo que se observen problemas de detección.

Elemento	Tipo	Por defecto	Descripción

`scoreThreshold`	float	`0.30`	Confianza mínima, entre 0 y 1, para considerar una detección como rostro. Un valor más bajo detecta más rostros (por ejemplo, de perfil o con poca luz) y también más falsos positivos.
`nmsThreshold`	float	`0.45`	Umbral de supresión de no máximos: elimina las detecciones duplicadas que se solapan sobre un mismo rostro.
`topK`	int	`5000`	Número máximo de candidatos que se evalúan por imagen antes de eliminar duplicados.

Como orientación, la configuración del servidor incluye tres perfiles:

Perfil	`scoreThreshold`	`nmsThreshold`	`topK`	Uso recomendado
Bajo falso positivo	`0.90`	`0.25`	`5000`	Documentos con elementos gráficos que generan detecciones erróneas.
Estándar	`0.60`	`0.30`	`5000`	Uso general.
Alta sensibilidad (por defecto)	`0.30`	`0.45`	`5000`	Fotografías con rostros pequeños, de perfil o con poca luz. Aumenta los falsos positivos.

Los valores por defecto que declara el WSDL (`0.30`, `0.45` y `5000`) corresponden al perfil de alta sensibilidad.

5.7. Códigos QR y de barras (QROptions)

Este bloque contiene los parámetros de la detección de códigos, que se activa con `redactQRCodes`. Se detectan códigos QR, PDF417, Data Matrix, Code 128, Code 39, EAN-13 e ITF. Si se incluye el bloque, deben informarse sus cuatro elementos.

Elemento	Tipo	Por defecto	Descripción
`maxQrPage`	int	`2`	Número máximo de códigos que se buscan en cada página intermedia, es decir, en todas salvo la primera y la última.
`maxQrFirstPage`	int	`4`	Número máximo de códigos que se buscan en la primera página.
`maxQrLastPage`	int	`3`	Número máximo de códigos que se buscan en la última página.
`cloneQRMargin`	boolean	`true`	Si se detecta un código en el margen de la segunda página, en las páginas siguientes solo se comprueba esa misma posición. Acelera el análisis de documentos con un código repetido en el margen de cada página.

La búsqueda en cada página finaliza al alcanzar el máximo indicado. Si las páginas de un tipo de documento pueden contener más códigos, conviene aumentar el valor correspondiente.

6. Operaciones

6.1. submitBatch

Envía uno o varios documentos PDF para su procesamiento asíncrono. Las opciones se aplican a todos los documentos del lote. Si se necesitan opciones distintas, los documentos deben enviarse en lotes separados.

Petición (submitBatchRequest)

Elemento	Tipo	Obligatorio	Descripción
`options`	`ProcessingOptions`	Sí	Opciones de procesamiento (apartado 5). Puede enviarse vacío (`<options/>`) para aplicar la configuración del servidor.
`documents`	`BatchDocumentItem` (1..n)	Sí	Documentos del lote.

Cada elemento `documents` contiene:

Elemento	Tipo	Obligatorio	Descripción
`clientId`	string	Sí	Identificador asignado por el sistema cliente para relacionar el resultado con su propio registro. Debe ser único dentro del lote y no debe contener datos personales (apartado 10).
`document`	`DocumentItem`	Sí	Documento: `fileName` , `contentType` y `fileContent` .
`confirmedRedactions`	`RedactionItem` (0..n)	No	Marcas revisadas que deben aplicarse. Se utilizan junto con `reconstructionMode` a `true` (apartado 7).

El tamaño total de la petición, con todos los documentos codificados en Base64, no puede superar el límite indicado en el apartado 3.5.

Respuesta (submitBatchResponse)

Elemento	Tipo	Descripción
`status`	`ServiceStatus`	Resultado de la recepción del lote.

`tickets`	`BatchTicketItem` (0..n)	Un elemento por documento, con su `clientId` y el identificador de seguimiento `serverTicketId`.

Antes de procesar los tickets, debe comprobarse el valor de `status/result`. Cada ticket se asocia a su documento mediante `clientId`; no debe presuponerse que los tickets se devuelven en el mismo orden que los documentos.

Ejemplo

Petición con dos documentos:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:sen="http://sensia.github.io/">
   <soapenv:Header>
      <sen:AuthHeader>
         <token>[TOKEN]</token>
         <cliente>IDENTIFICADOR_DEL_USUARIO_O_APLICACION</cliente>
      </sen:AuthHeader>
   </soapenv:Header>
   <soapenv:Body>
      <sen:submitBatchRequest>
         <options>
            <documentOptions>
               <redactQRCodes>true</redactQRCodes>
               <redactNames>true</redactNames>
               <redactNERNames>true</redactNERNames>
               <allowedWords>
                  <words>Ayuntamiento de Ejemplo</words>
               </allowedWords>
            </documentOptions>
            <faceOptions>
               <maskFaces>true</maskFaces>
            </faceOptions>
            <redactionPreview>false</redactionPreview>
         </options>
         <documents>
            <clientId>EXP-2026-001</clientId>
            <document>
               <fileName>resolucion.pdf</fileName>
               <contentType>application/pdf</contentType>
               <fileContent>JVBERi0xLjQK...</fileContent>
            </document>
         </documents>
         <documents>
            <clientId>EXP-2026-002</clientId>
            <document>
               <fileName>notificacion.pdf</fileName>
               <contentType>application/pdf</contentType>
               <fileContent>JVBERi0xLjcK...</fileContent>
            </document>
         </documents>
      </sen:submitBatchRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Respuesta:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:sen="http://sensia.github.io/">
   <soapenv:Body>
      <sen:submitBatchResponse>
         <status>
            <result>SUCCESS</result>
            <message>Lote encolado correctamente</message>
         </status>
         <tickets>
            <clientId>EXP-2026-001</clientId>
            <serverTicketId>8f7b2c9a-1122-4a5c-b8e9-d7e6f5a4b3c2</serverTicketId>
         </tickets>
         <tickets>
            <clientId>EXP-2026-002</clientId>
            <serverTicketId>2c4e6a8b-3344-4d5e-9f10-a1b2c3d4e5f6</serverTicketId>
         </tickets>
      </sen:submitBatchResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

6.2. getStatus

Consulta el estado de un documento enviado con `submitBatch`. Se recomienda una espera inicial de 5 segundos entre consultas, que se aumenta progresivamente hasta un máximo de 30 segundos, hasta obtener un estado final (apartado 9.1).

Petición (getStatusRequest)

Elemento	Tipo	Obligatorio	Descripción
`ticketId`	string	Sí	Identificador de seguimiento (`serverTicketId`) devuelto por `submitBatch` .

Respuesta (getStatusResponse)

Elemento	Tipo	Descripción
`ticketId`	string	Identificador consultado.
`processedDocument`	`DocumentItem`	Documento resultante. Solo se incluye cuando el estado es `COMPLETED` .
`redactionResults`	`RedactionItem` (0..n)	Detalle de las detecciones (apartado 9.5).
`taskStatus`	`TaskStatus`	Estado del documento (`processState`) y, en su caso, un mensaje descriptivo (`message`).

`status`	`ServiceStatus`	Resultado de la consulta. Es independiente del estado del documento: una consulta correcta puede informar de un documento con estado `ERROR`.

Recuperación única. Cuando el estado es `COMPLETED`, el resultado se elimina del servidor tras la respuesta (apartado 3.5). Por ello:

- El sistema cliente debe guardar el documento y las detecciones antes de dar por finalizada la consulta.
- Si la respuesta no llega a recibirse por un error de red o de tiempo de espera, el documento debe enviarse de nuevo.

Si se solicitó vista previa, `processedDocument` contiene el documento de vista previa (apartado 5.1).

Documento rasterizado. El documento definitivo se rasteriza: cada página se convierte en imagen. Por ello, no conserva:

- la capa de texto;
- las capas ocultas;
- los metadatos originales;
- el contenido situado bajo las zonas ocultadas.

Si el documento debe ser accesible o permitir búsquedas de texto, será necesario un tratamiento posterior.

Ejemplo

Petición:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:sen="http://sensia.github.io/">
   <soapenv:Header>
      <sen:AuthHeader>
         <token>[TOKEN]</token>
         <cliente>IDENTIFICADOR_DEL_USUARIO_O_APLICACION</cliente>
      </sen:AuthHeader>
   </soapenv:Header>
   <soapenv:Body>
      <sen:getStatusRequest>
         <ticketId>8f7b2c9a-1122-4a5c-b8e9-d7e6f5a4b3c2</ticketId>
      </sen:getStatusRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Respuesta mientras el documento se procesa:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:sen="http://sensia.github.io/">
   <soapenv:Body>
      <sen:getStatusResponse>
         <ticketId>8f7b2c9a-1122-4a5c-b8e9-d7e6f5a4b3c2</ticketId>
         <taskStatus>
            <processState>PROCESSING</processState>
            <message>Documento en proceso</message>
         </taskStatus>
         <status>
            <result>SUCCESS</result>
            <message>OK</message>
         </status>
      </sen:getStatusResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

Respuesta con el documento finalizado:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:sen="http://sensia.github.io/">
   <soapenv:Body>
      <sen:getStatusResponse>
         <ticketId>8f7b2c9a-1122-4a5c-b8e9-d7e6f5a4b3c2</ticketId>
         <processedDocument>
            <fileName>resolucion_censurado.pdf</fileName>
            <contentType>application/pdf</contentType>
            <fileContent>JVBERi0xLjQK...</fileContent>
         </processedDocument>
         <redactionResults>
            <detectionType>FACE</detectionType>
            <action>REDACTED</action>
            <description>Rostro ocultado</description>
            <page>1</page>
            <coordinates>
               <x>150.5</x><y>300.0</y><w>50.0</w><h>50.0</h>
            </coordinates>
         </redactionResults>
         <redactionResults>
            <detectionType>SENSITIVE_DATA</detectionType>
            <subType>NIF</subType>
            <action>REDACTED</action>
            <description>NIF</description>
            <text>12345678Z</text>
            <page>2</page>
            <coordinates>
               <x>100.0</x><y>500.0</y><w>80.0</w><h>15.0</h>
            </coordinates>
         </redactionResults>
         <redactionResults>
            <detectionType>NAMED_ENTITY</detectionType>
            <subType>ORG</subType>
            <action>WHITELISTED</action>
            <text>Ayuntamiento de Ejemplo</text>
            <page>1</page>
            <coordinates>
               <x>60.0</x><y>760.0</y><w>160.0</w><h>12.0</h>
            </coordinates>
         </redactionResults>
         <taskStatus>
            <processState>COMPLETED</processState>
            <message>Procesamiento finalizado</message>
         </taskStatus>
         <status>
            <result>SUCCESS</result>
            <message>OK</message>
         </status>
      </sen:getStatusResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

6.3. redactImage

Detecta y oculta rostros en una imagen JPG o PNG, y devuelve el resultado en la misma respuesta.

Petición (redactImageRequest)

Elemento	Tipo	Obligatorio	Descripción

`image`	`DocumentItem`	Sí	Imagen que se va a procesar.
`faceOptions`	`FaceMaskingOptions`	No	Opciones de ocultación de rostros (5.5). Si se omite, se aplica la configuración del servidor, que en la estándar oculta los rostros.
`yunetOptions`	`YunetOptions`	No	Parámetros del detector de rostros (5.6). Si se incluye, deben informarse sus tres elementos.
`redactionPreview`	boolean	No	Si es `true`, devuelve la imagen con las detecciones marcadas, sin aplicar el pixelado. Por defecto, `false`.

Respuesta (redactImageResponse)

Elemento	Tipo	Descripción
`image`	`DocumentItem`	Imagen resultante, siempre en JPEG (`image/jpeg`) y reducida a un máximo de 1024 píxeles. No se incluye si se produce un error.
`redactionResults`	`RedactionItem` (0..n)	Detalle de los rostros detectados.
`status`	`ServiceStatus`	Resultado de la operación.

Al ser una operación síncrona, conviene configurar en el cliente un tiempo de espera suficiente, en particular para imágenes de gran tamaño (apartado 8).

Ejemplo

Petición:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:sen="http://sensia.github.io/">
   <soapenv:Header>
      <sen:AuthHeader>
         <token>[TOKEN]</token>
         <cliente>IDENTIFICADOR_DEL_USUARIO_O_APLICACION</cliente>
      </sen:AuthHeader>
   </soapenv:Header>
   <soapenv:Body>
      <sen:redactImageRequest>
         <image>
            <fileName>fotografia.jpg</fileName>
            <contentType>image/jpeg</contentType>
            <fileContent>/9j/4AAQSkZJRg...</fileContent>
         </image>
         <faceOptions>
            <maskFaces>true</maskFaces>
            <pixelationIntensity>65</pixelationIntensity>
         </faceOptions>
         <!-- Opcional. Si se incluye, deben informarse sus tres elementos (perfil
estándar) -->
         <yunetOptions>
            <scoreThreshold>0.60</scoreThreshold>
            <nmsThreshold>0.30</nmsThreshold>
            <topK>5000</topK>
         </yunetOptions>
      </sen:redactImageRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Respuesta:

```xml
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:sen="http://sensia.github.io/">
   <soapenv:Body>
      <sen:redactImageResponse>
         <image>
            <fileName>fotografia_censurada.jpg</fileName>
            <contentType>image/jpeg</contentType>
            <fileContent>/9j/4AAQSkZJRg...</fileContent>
         </image>
         <redactionResults>
            <detectionType>FACE</detectionType>
            <action>REDACTED</action>
            <coordinates>
               <x>212.0</x><y>88.0</y><w>96.0</w><h>118.0</h>
            </coordinates>
         </redactionResults>
         <redactionResults>
            <detectionType>FACE</detectionType>
            <action>REDACTED</action>
            <coordinates>
               <x>430.0</x><y>102.0</y><w>84.0</w><h>104.0</h>
            </coordinates>
         </redactionResults>
         <status>
            <result>SUCCESS</result>
            <message>Imagen procesada. 2 rostros detectados.</message>
         </status>
      </sen:redactImageResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

6.4. extractImages

Extrae las imágenes incrustadas en un documento PDF. Esta operación no aplica ninguna anonimización: las imágenes se devuelven tal como figuran en el documento original. Pueden contener datos personales y deben tratarse como tales.

Petición (extractImagesRequest)

Elemento	Tipo	Obligatorio	Descripción
`document`	`DocumentItem`	Sí	Documento PDF de origen.
`minWidth` / `minHeight`	int	No	Dimensiones mínimas, en píxeles. Las imágenes menores se descartan. Por defecto, `0`.
`maxWidth` / `maxHeight`	int	No	Dimensiones máximas, en píxeles. Si se omiten, no se aplica límite superior.

Los filtros de tamaño permiten descartar elementos gráficos pequeños, como iconos, líneas separadoras o viñetas.

Respuesta (extractImagesResponse)

Elemento	Tipo	Descripción
`images`	`DocumentList`	Imágenes extraídas, cada una en un elemento `document` con su nombre, tipo MIME y contenido en Base64.
`status`	`ServiceStatus`	Resultado de la operación.

Ejemplo

Petición:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:sen="http://sensia.github.io/">
   <soapenv:Header>
      <sen:AuthHeader>
         <token>[TOKEN]</token>
         <cliente>IDENTIFICADOR_DEL_USUARIO_O_APLICACION</cliente>
      </sen:AuthHeader>
   </soapenv:Header>
   <soapenv:Body>
      <sen:extractImagesRequest>
         <document>
            <fileName>documento.pdf</fileName>
            <contentType>application/pdf</contentType>
            <fileContent>JVBERi0xLjQK...</fileContent>
         </document>
         <minWidth>100</minWidth>
         <minHeight>100</minHeight>
      </sen:extractImagesRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

Respuesta:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:sen="http://sensia.github.io/">
   <soapenv:Body>
      <sen:extractImagesResponse>
         <images>
            <document>
               <fileName>extracted_image_1.jpg</fileName>
               <contentType>image/jpeg</contentType>
               <fileContent>/9j/4AAQSkZJRg...</fileContent>
            </document>
            <document>
               <fileName>extracted_image_2.png</fileName>
               <contentType>image/png</contentType>
               <fileContent>iVBORw0KGgo...</fileContent>
            </document>
         </images>
         <status>
            <result>SUCCESS</result>
            <message>Se han extraído 2 imágenes.</message>
         </status>
      </sen:extractImagesResponse>
   </soapenv:Body>
</soapenv:Envelope>
```

7. Revisión y confirmación de marcas

Este flujo permite que una persona revise las detecciones antes de generar el documento definitivo, sin repetir el análisis:

1. **Análisis con vista previa.** El cliente envía el documento con `submitBatch` y `redactionPreview` a `true`. Al finalizar, `getStatus` devuelve el documento de vista previa y la lista de detecciones. Cada detección incluye en `sourceDocumentHash` la huella del documento original analizado. El resultado solo puede recuperarse una vez (apartado 3.5). Por ello, el sistema cliente debe guardar la lista de detecciones, junto con el documento original, hasta completar la confirmación.

2. **Revisión.** El sistema cliente presenta las detecciones a la persona revisora, por ejemplo junto al documento de vista previa, y esta decide cuáles se aplican. Para descartar una detección, se establece su elemento `apply` a `false`.

3. **Confirmación.** El cliente envía de nuevo el mismo documento original con `submitBatch`, con estos elementos:

   - en `options`, `redactionPreview` a `false` y `reconstructionMode` a `true`;
   - en `confirmedRedactions`, la lista de detecciones revisadas.

   Se recomienda mantener el resto de opciones de procesamiento utilizadas en la vista previa. El motor aplica directamente las marcas confirmadas, sin volver a analizar el documento.

4. **Recuperación.** Con `getStatus` se obtiene el documento definitivo. En la aplicación directa de marcas, las imágenes y los rostros confirmados se cubren con cajas blancas.

5. **Verificación de integridad.** Si el documento enviado no coincide con el analizado en el primer paso, la tarea finaliza con el estado `HASH_MISMATCH_ERROR`.

Lo más sencillo es devolver las detecciones recibidas en la vista previa sin modificarlas, cambiando únicamente el valor de `apply` en las descartadas.

Ejemplo de petición de confirmación:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:sen="http://sensia.github.io/">
   <soapenv:Header>
      <sen:AuthHeader>
         <token>[TOKEN]</token>
         <cliente>IDENTIFICADOR_DEL_USUARIO_O_APLICACION</cliente>
      </sen:AuthHeader>
   </soapenv:Header>
   <soapenv:Body>
      <sen:submitBatchRequest>
         <options>
            <documentOptions>
               <redactQRCodes>true</redactQRCodes>
               <redactNames>true</redactNames>
            </documentOptions>
            <redactionPreview>false</redactionPreview>
            <reconstructionMode>true</reconstructionMode>
         </options>
         <documents>
            <clientId>EXP-2026-001</clientId>
            <document>
               <fileName>resolucion.pdf</fileName>
               <contentType>application/pdf</contentType>
               <fileContent>JVBERi0xLjQK...</fileContent>
            </document>
            <confirmedRedactions>
               <detectionType>SENSITIVE_DATA</detectionType>
               <subType>NIF</subType>
               <action>REDACTED</action>
               <text>12345678Z</text>
               <page>2</page>
               <coordinates>
                  <x>100.0</x><y>500.0</y><w>80.0</w><h>15.0</h>
               </coordinates>
               <sourceDocumentHash>[HUELLA_DEVUELTA_EN_LA_VISTA_PREVIA]
</sourceDocumentHash>
               <apply>true</apply>
            </confirmedRedactions>
            <confirmedRedactions>
               <detectionType>PERSON_NAME</detectionType>
               <action>REDACTED</action>
               <text>Ejemplo</text>
               <page>1</page>
               <coordinates>
                  <x>80.0</x><y>210.0</y><w>60.0</w><h>12.0</h>
               </coordinates>
               <sourceDocumentHash>[HUELLA_DEVUELTA_EN_LA_VISTA_PREVIA]
</sourceDocumentHash>
               <apply>false</apply>
            </confirmedRedactions>
         </documents>
      </sen:submitBatchRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

8. Gestión de errores

Los errores se comunican en cuatro niveles:

Nivel	Dónde se informa	Casos habituales
HTTP	Código de estado de la respuesta	`403` si la petición se envía por HTTP sin cifrar o, en instalaciones con la vía con certificado activada, por el puerto 443. `413` si la petición supera el tamaño máximo; estas peticiones se rechazan antes de procesarse. `503` si el servidor está saturado (apartado 3.5).
SOAP Fault	`ServiceFault` : `errorCode` y `errorMessage`	Autenticación no válida (`AUTH_ERROR`) y otros errores de la petición. Conforme a SOAP 1.1, se devuelve con HTTP 500.
Resultado de la operación	`status` : `result` , `message` y, opcionalmente, `cause`	Presente en todas las respuestas (apartado 9.4).
Estado del documento	`taskStatus/processState` en `getStatus`	Estado del procesamiento asíncrono (apartado 9.1).

Ejemplo de SOAP Fault. Los textos y el valor de `faultcode` son orientativos:

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
   <soap:Body>
      <soap:Fault>
         <faultcode>soap:Client</faultcode>
         <faultstring>Autenticación no válida</faultstring>
         <detail>
            <sen:ServiceFault xmlns:sen="http://sensia.github.io/">
               <errorCode>AUTH_ERROR</errorCode>
               <errorMessage>Token de autenticación no válido</errorMessage>
            </sen:ServiceFault>
         </detail>
      </soap:Fault>
   </soap:Body>
</soap:Envelope>
```

Recomendaciones:

- **Errores transitorios.** Reintente, con una espera creciente entre intentos, solo ante HTTP 503, fallos de red o tiempos de espera agotados.
- **Errores que requieren corrección.** No reintente ante `AUTH_ERROR` , `INVALID_DOCUMENT_ERROR` o `HASH_MISMATCH_ERROR` sin corregir antes la causa.
- **Reconocimiento de entidades no disponible.** Ante `ERROR` por esta causa, no reintente de inmediato (apartado 9.1).
- **Resultado no disponible.** Ante `NOT_FOUND` , o si se pierde la respuesta de un `getStatus` con estado `COMPLETED` , envíe de nuevo el documento.

- **Tiempos de espera.** Configure en el cliente HTTP tiempos de espera suficientes para las operaciones síncronas y para el envío de lotes grandes; por ejemplo, 120 segundos.
- **Registro para soporte.** Registre en el sistema cliente, para la resolución de incidencias, el `clientId`, el `serverTicketId`, el estado y el mensaje de cada documento. No registre el contenido de los documentos ni el texto de las detecciones.

9. Referencia de tipos y enumerados

9.1. Estados del documento (ProcessStateEnum)

Valor	Final	Descripción
`QUEUED`	No	El documento está en cola.
`PROCESSING`	No	El documento se está procesando.
`COMPLETED`	Sí	Procesamiento finalizado. El documento resultante y las detecciones están disponibles para una única recuperación.
`NOT_FOUND`	Sí	El identificador no existe o sus resultados ya no se conservan.
`AUTH_ERROR`	Sí	La autenticación no es válida.
`INVALID_DOCUMENT_ERROR`	Sí	El contenido no es un PDF válido, está dañado o está cifrado o protegido con contraseña.
`HASH_MISMATCH_ERROR`	Sí	Las marcas confirmadas no corresponden al documento enviado (apartado 7).
`ERROR`	Sí	Error durante el procesamiento. El elemento `message` contiene el detalle.

Cuando se solicita el reconocimiento de entidades (opciones `redactNER*`) y el servicio no está disponible, se devuelve `ERROR`. En ese caso, reintentar la petición no resuelve el problema. Puede enviarse el documento sin reconocimiento de entidades o solicitarse al equipo técnico de la instalación que restablezca el servicio.

9.2. Tipos de detección (DetectionTypeEnum)

Valor	Descripción
`SENSITIVE_DATA`	Dato detectado por los patrones configurados (NIF, IBAN, teléfono, etc.) o por la lista negra.
`PERSON_NAME`	Nombre o apellido detectado mediante diccionario.
`NAMED_ENTITY`	Entidad detectada por reconocimiento de entidades. La categoría se indica en `subType` (apartado 5.3).
`FACE`	Rostro.

`QR_CODE`	Código QR o código de barras.
`IMAGE`	Imagen incrustada eliminada o, en la vista previa, que se eliminaría.
`FIXED_ZONE`	Zona fija de ocultación.

En las detecciones de tipo `SENSITIVE_DATA`, `subType` contiene la etiqueta del patrón que ha detectado el dato. En la configuración estándar, las etiquetas son las siguientes:

Etiqueta	Dato
`NIF`, `NIE`, `CIF`	Documentos de identidad y de personas jurídicas.
`CREDIT_CARD`	Tarjetas bancarias.
`IBAN`	Cuentas bancarias.
`RC`	Referencias catastrales.
`DIRECCION`	Direcciones postales y ubicaciones rurales.
`CODIGO_SEGURO`, `CSV`	Códigos seguros de verificación y de registro.
`UUID`	Identificadores únicos.
`EMAIL`	Correos electrónicos.
`TELF`	Teléfonos.
`BASTIDOR`, `MATRICULA`	Bastidores y matrículas de vehículos.
`CODE_BAR_PAGO`	Códigos de barras de pago de 42 dígitos.

Cada instalación puede añadir o desactivar patrones, por lo que el sistema cliente debe admitir etiquetas no incluidas en esta tabla.

9.3. Acciones (ActionStateEnum)

Valor	Descripción
`REDACTED`	El elemento se ha ocultado o, en la vista previa, se ocultaría.
`WHITELISTED`	El elemento se detectó, pero no se ha ocultado por coincidir con la lista blanca.

`EXCLUDED_ZONE`	El elemento no se ha procesado por encontrarse en una zona de exclusión, como la del logotipo.
`REMOVED`	El elemento, normalmente una imagen, se ha eliminado del documento.

9.4. Resultado de la operación (StatusEnum)

Valor	Descripción
`SUCCESS`	La operación se ha realizado correctamente.
`WARNING`	La operación se ha completado con incidencias. Los elementos `message` y `cause` contienen el detalle.
`ERROR`	La operación no se ha podido realizar.

9.5. Detección (RedactionItem)

Elemento	Tipo	Obligatorio	Descripción
`detectionType`	`DetectionTypeEnum`	Sí	Tipo de detección (9.2).
`subType`	string	No	Subtipo o categoría de la detección (apartados 5.3 y 9.2).
`action`	`ActionStateEnum`	Sí	Acción aplicada (9.3).
`description`	string	No	Descripción de la detección.
`text`	string	No	Texto detectado. **Contiene el dato original sin anonimizar.**
`page`	int	No	Página en la que se encuentra la detección.
`coordinates`	`Coordinate`	Sí	Posición y tamaño (apartado 4.3).
`sourceDocumentHash`	string	No	Huella del documento original analizado (apartado 7).
`apply`	boolean	No	Indica si la marca debe aplicarse al confirmar (apartado 7). Por defecto, `true`.
`pseudonym`	string	No	Seudónimo asignado, si se utiliza la seudonimización (apartado 5.4).

9.6. Otros tipos

DocumentItem: fichero enviado o devuelto.

Elemento	Tipo	Obligatorio	Descripción
`fileName`	string	Sí	Nombre del fichero. No debe contener datos personales.
`contentType`	string	Sí	Tipo MIME: `application/pdf`, `image/jpeg` o `image/png`.
`fileContent`	base64Binary	Sí	Contenido del fichero codificado en Base64.

BatchTicketItem: identificador de seguimiento de un documento del lote.

Elemento	Tipo	Obligatorio	Descripción
`clientId`	string	Sí	Identificador enviado por el sistema cliente.
`serverTicketId`	string	Sí	Identificador de seguimiento para `getStatus`.

TaskStatus: estado de un documento.

Elemento	Tipo	Obligatorio	Descripción
`processState`	`ProcessStateEnum`	No	Estado del procesamiento (9.1).
`message`	string	No	Descripción del estado.

ServiceStatus: resultado de una operación.

Elemento	Tipo	Obligatorio	Descripción
`result`	`StatusEnum`	Sí	Resultado (9.4).
`message`	string	Sí	Mensaje descriptivo.
`cause`	string	No	Causa del error o de la advertencia.

ServiceFault: detalle de un SOAP Fault.

Elemento	Tipo	Obligatorio	Descripción

`errorCode`	string	Sí	Código del error.
`errorMessage`	string	Sí	Descripción del error.

Coordinate: zona rectangular (apartado 4.3). Contiene los elementos `x`, `y`, `w` y `h`, de tipo float y todos obligatorios.

Listas:

Tipo	Elemento repetible	Uso
`CoordinatesList`	`coordinate` (`Coordinate`, 0..n)	Zonas de exclusión de imágenes y zonas fijas.
`WordList`	`words` (string, 0..n)	Listas blanca y negra.
`DocumentList`	`document` (`DocumentItem`, 0..n)	Imágenes devueltas por `extractImages`.

10. Buenas prácticas y protección de datos

- **Revisión humana.** La detección automática puede presentar omisiones, en particular en estos casos:

  - documentos escaneados de baja resolución.
  - datos repartidos en varias líneas dentro de tablas.
  - códigos QR situados junto a bordes o líneas de tabla.
  - texto contenido en imágenes.
  - rostros girados, por ejemplo, fotografías de documentos de identidad escaneadas de lado o muy pequeños en imágenes de gran tamaño.
  - documentos que combinan páginas con texto y páginas escaneadas, porque la necesidad de OCR se determina a partir de la primera página.

  Se recomienda que una persona revise el resultado antes de publicar un documento, preferiblemente mediante el flujo del apartado 7. Las limitaciones conocidas se describen en el `README.md` del proyecto.

- **Pruebas previas.** Antes de la puesta en producción, valide la integración en un entorno de preproducción con documentos representativos del organismo y con datos ficticios.

- **Datos personales en las respuestas.** El campo `text` de las detecciones contiene los datos detectados sin anonimizar, y `fileContent` contiene el documento completo. Ambos deben tratarse como información personal:

  - con acceso restringido;
  - sin registrarse en logs;
  - eliminándose cuando dejen de ser necesarios.

- **Nombres de fichero e identificadores.** Los valores de `fileName` y `clientId` pueden quedar registrados en el servidor. No deben contener datos personales, como el nombre o el NIF de la persona interesada.

- **Vista previa.** El documento de vista previa conserva todo el contenido original y nunca debe publicarse.

- **Recuperación única.** El resultado de cada documento solo puede recuperarse una vez (apartado 3.5). Debe guardarse en cuanto se recibe, antes de cualquier otro tratamiento.

- **Consultas y concurrencia.**

  - Consulte `getStatus` con una espera inicial de 5 segundos, aumentándola progresivamente hasta 30 segundos.
  - Ante respuestas HTTP 503, reintente con una espera creciente.
  - El límite de peticiones simultáneas del servidor es común a todos los sistemas integradores y a la aplicación web. Por ello, conviene acordar con el administrador el número máximo de envíos simultáneos de cada sistema.

- **Seguridad del transporte.** Utilice siempre HTTPS y verifique el certificado del servidor. No desactive esa verificación en ningún entorno.

- **Credenciales.** El token y, en su caso, el certificado de cliente deben almacenarse de forma segura y sustituirse si se sospecha que han quedado expuestos. El token es común a todos los integradores de la instalación, por lo que cualquier sospecha de exposición debe comunicarse de inmediato al administrador.

- **Seudonimización.** Los documentos seudonimizados pueden seguir conteniendo datos personales (apartado 5.4).

- **Cambios de versión.** Antes de cada actualización de la instalación, consulte en `CHANGELOG.md` los cambios que afecten al contrato del servicio. El sistema cliente debe admitir valores no previstos en los campos informativos, como nuevas etiquetas en `subType` o nuevos textos en `message`.

- **Marco jurídico.** El tratamiento debe figurar en el registro de actividades de tratamiento del organismo (artículo 30 del RGPD). Si la instalación de SensIA la gestiona una entidad distinta de la responsable de los documentos, la relación entre ambas debe formalizarse como encargo del tratamiento (artículo 28 del RGPD). Las medidas de seguridad deben ser coherentes con el Esquema Nacional de Seguridad aplicable al sistema integrador.