

Estándar de reutilización y apertura de datos

Estándares de desarrollo

Palma, septiembre de 2025



**Vicepresidència Primera i Conselleria
d'Economia, Hisenda i Innovació**
Direcció General d'Estratègia Digital
i Desenvolupament Tecnològic



Índice

Historial de versiones.....	3
1. Introducción.....	4
1.1. Objetivos.....	4
2. API REST de reutilización y apertura de datos.....	6
2.1. Publicación y documentación de la API REST.....	6
3. Servicios REST de reutilización y apertura de datos.....	8
3.1. Publicación y documentación de los servicios REST.....	8
3.2. Campos de salida de los servicios REST.....	9
3.3. Paginación.....	11
3.4. Parámetros de entrada de los servicios REST.....	13
3.5. Validación de la API y servicios REST.....	15
3.6. Guía y ejemplos de implementación.....	15



Historial de versiones

Fecha	Versión	Descripción	Autor
29/07/25	1.0	Redacción del estándar	DGEDDT
16/09/25	1.1	Revisión del estándar	DGEDDT
19/09/25	1.1	Publicación del estándar	DGEDDT



1. Introducción

La reutilización y apertura de datos es uno de los pilares fundamentales de la transformación digital de las administraciones públicas. En este sentido, el Gobierno de las Illes Balears (GOIB) promueve el uso de servicios digitales que faciliten la consulta y reutilización de sus datos. Esto se aplica tanto a nivel interno como externo, utilizando tecnologías basadas en arquitecturas abiertas y formatos estructurados. Este documento establece el estándar técnico que deben cumplir las API REST y los servicios REST para garantizar una reutilización de datos efectiva, segura e interoperable.

Este estándar se enmarca en el cumplimiento de la **Directiva 2019/1024/UE** del Parlamento Europeo y del Consejo, del 20 de junio de 2019, relativa a los datos abiertos y la reutilización de la información del sector público. También cumple con la **Ley 37/2007**, del 16 de noviembre, sobre la reutilización de la información del sector público, y con el **Acuerdo del Consejo de Gobierno del 17 de abril de 2023**, que establece medidas para impulsar la apertura y reutilización de los datos del Gobierno de las Illes Balears. El objetivo de esta normativa es fomentar la reutilización y apertura de los datos del Sector Público para fines sociales, económicos e institucionales, con el fin de promover una administración más abierta, eficiente y colaborativa.

1.1. Objetivos

El estándar de reutilización y apertura de datos del Gobierno de las Illes Balears establece directrices técnicas comunes para todos los servicios REST de reutilización y apertura de datos desarrollados o gestionados por la Administración de la Comunidad Autónoma de las Illes Balears (CAIB).

Los objetivos principales de este estándar son:

- **Garantizar la accesibilidad y reutilización de los datos públicos** por parte de la ciudadanía, empresas y otras administraciones. Esto se logra mediante servicios digitales seguros, documentados y accesibles a través de Internet.
- **Evitar silos de datos y facilitar la reutilización interna de la información** por parte de la propia Administración. El propósito es mejorar la eficiencia administrativa y optimizar la elaboración y evaluación de políticas públicas.



- **Estandarizar la publicación y consumo de datos** a través de API REST. Esto se hace con estructuras de datos coherentes, metadatos claros y formatos abiertos, principalmente JSON.
- **Asegurar el cumplimiento normativo** en materia de reutilización y apertura de datos en los ámbitos europeo, estatal y autonómico. También se busca garantizar la protección de datos personales y la seguridad de la información.
- **Impulsar la cultura del dato abierto** como recurso estratégico para el desarrollo de servicios públicos digitales, la innovación y la toma de decisiones basada en datos.

2. API REST de reutilización y apertura de datos

Las aplicaciones desarrolladas deben publicar servicios web que permitan la integración con otras aplicaciones y la reutilización de datos. Se restringe el uso de servicios SOAP en favor de los servicios REST.

La publicación de los servicios REST se realizará a través de una API REST. Se pueden definir dos tipos de API REST: externa e interna.

- **API externa.** Aquí se publicarán los servicios REST accesibles desde Internet que permitirán la reutilización externa de los datos. Por defecto, los servicios REST de reutilización se publicarán en esta API externa.
- **API interna.** Aquí se publicarán los servicios REST no accesibles desde Internet que permitirán la reutilización interna de los datos por part del GOIB. Para poder publicar un servicio REST en la API interna, será necesario justificar su necesidad y recibir la autorización previa de la Dirección General de Estategia Digial y Desarrollo Tecnológico (DGEDDT) del GOIB.

2.1. *Publicación y documentación de la API REST*

- 1) Cada aplicación deberá publicar su API REST, tanto externa como interna, en su propio contexto, con el formato `/<código-aplicación>api/` (por ejemplo, `/ibdonamapaapi/`).



- 2) La API REST debe estar documentada en catalán¹ utilizando la herramienta Swagger UI que debe ser de acceso libre, sin necesidad de autenticación o control de acceso.
- 3) El Swagger UI debe estar disponible a través de una URL con la siguiente estructura: `https://<servidor>/<código-aplicación>api/[interna|externa]`. Por ejemplo, «`https://intranet.caib.es/ibdonamapaapi/externa`» (no se admitirán otras estructuras).
- 4) La API REST debe tener un título en mayúsculas que identifique el código de la aplicación, seguido de un guión (-) y el tipo de API (API EXTERNA o API INTERNA). Un ejemplo sería IBDONAMAPA – API EXTERNA.
- 5) También debe mostrar una descripción de la aplicación informática sobre la que se ha implementado la API REST. Un ejemplo de descripción es: Servei de digitalització mapa de recursos i serveis per a l'atenció de dones víctimes de violència masclista de les Illes Balears.



Figura 1: Ejemplo de descripción de la API REST.

- 6) El texto de la pestaña del navegador web tiene que ser igual que el texto del título del Swagger UI. Ejemplo,



Figura 2: Ejemplo del nombre de la pestaña del navegador Web.

- 7) Todos los servicios REST de reutilización publicados en el Swagger UI se tienen que listar agrupados en la misma sección (tag) con el nombre **Serveis de reutilització**. Se muestra un ejemplo de como se tendría que presentar la lista de servicios.

¹ Artículos 21 y 22 del Decreto 49/2018, de 21 de diciembre, sobre el uso de las lenguas oficiales en la Administración de la Comunidad Autónoma de las Illes Balears.



The screenshot shows the 'IBDONAMAPA - API REST EXTERNA' interface. At the top right, there are version and compliance indicators: '7.2.0.GA-redhat-00009' and 'OAS3'. Below the title, the URL '/bdonamapsapi/externa/apidocs' is displayed. A description states: 'Servei de digitalització mapa de recursos i serveis per a l'atenció de dones víctimes de violència masclista de les illes Balears.' A 'Servers' dropdown menu is set to 'https://intranet.caib.es/ibdonamapaarpiexterna - Generated server url!'. The 'Serveis de reutilització' section is highlighted with a red box and contains three service entries:

Method	Endpoint	Description
GET	/reutilitzacio/recursos	Consulta paginada de recursos
GET	/reutilitzacio/edificis-tutelats	Consulta d'edificis tutelats per IB-Dona
GET	/reutilitzacio/ubicacions	Consulta paginada de ubicacions

Figura 3: Ejemplo de lista de servicios REST en la sección Serveis de reutilització.

3. Servicios REST de reutilización y apertura de datos

En cumplimiento de los principios recogidos en la **Ley 37/2007**, de 16 de noviembre, sobre reutilización de la información del sector público y el **Acuerdo del Consejo de Gobierno, de 17 de abril de 2023**, por el que se establecen las medidas para impulsar la apertura y reutilización de los datos del Gobierno de las Illes Balears, es obligatorio que los sistemas de información del GOIB cuenten con servicios REST que permitan la reutilización, interna y/o externa, de sus datos.

El objetivo es evitar sistemas de información estancos que no puedan ser consultados y facilitar el uso y la reutilización de los datos que manejan. Estos servicios REST deben permitir consultar los datos en un formato abierto, estructurado y de manera desagregada.

A continuación, se exponen los requisitos que deben cumplir los servicios REST de reutilización de datos. El objetivo de estos requisitos es estandarizar y homogeneizar la implementación de estos servicios REST.

3.1. **Publicación y documentación de los servicios REST**

- 1) Por defecto, los servicios REST de reutilización de datos deben publicarse en la API REST externa (accesible desde Internet) y sin securizar.



- 2) Si se desea publicar el servicio REST de reutilización en la API interna (no accesible desde Internet), será necesario justificar la necesidad y obtener la autorización de la DGEDDT.
- 3) En casos excepcionales, los servicios publicados en la API interna podrán estar securizados mediante mecanismos de autenticación y control de acceso. En tal caso, se deberá justificar la necesidad y obtener la autorización de la DGEDDT.
- 4) Los servicios REST deben definirse como métodos GET (NO como métodos POST).
- 5) Los servicios REST deben devolver los datos en formato JSON.
- 6) La URL del servicio REST de reutilización de datos debe incluir el nombre del recurso que se consulta. Este nombre debe estar escrito en catalán y en plural.
- 7) Esta URL debe seguir la siguiente estructura: `https://<servidor>/<codi-aplicació>api/[interna|externa]/reutilitzacio/<nombre-recurso>`:
«`https://www.caib.es/carpetaapi/externa/reutilitzacio/usuaris`»
- 8) Si el nombre del recurso está compuesto por dos o más palabras, deberán separarse mediante un guion (-). Por ejemplo:
«`https://www.caib.es/carpetaapi/externa/reutilitzacio/centres-concertats`»
- 9) Cada servicio REST debe ir acompañado de una descripción completa y en catalán² de su funcionalidad.

3.2. Campos de salida de los servicios REST

3.2.1. Campos de metadatos

- 1) Las respuestas de los servicios REST deben mostrar los siguientes 5 campos de metadatos, en formato *camelCase* y siguiendo el mismo orden:
 1. **title:** nombre del conjunto de datos (por ejemplo, Allotjaments turístics de les Illes Balears).
 2. **description:** descripción clara y concisa del conjunto de datos. Se explicará qué contiene y para qué sirve.

² Artículos 21 y 22 del Decreto 49/2018, de 21 de diciembre, sobre el uso de las lenguas oficiales en la Administración de la Comunidad Autónoma de las Illes Balears.



3. **spatial:** cobertura geográfica de los datos. El contenido de este campo debe ser uno de los siguientes valores: Illes Balears, Mallorca, Menorca, Eivissa o Formentera.
 4. **creator:** el [código DIR3](#) de la unidad orgánica propietaria de los datos.
 5. **dateDownload:** fecha de la descarga (en formato ISO8601: 2022-07-26T12:58:55+02:00).
- 2) Todos los campos anteriores se tienen que agrupar en el campo **metadata**, tal y como se muestra en la figura siguiente:

```
{
  "metadata": {
    "title": "Allotjaments turístics Mallorca",
    "description": "Dades dels allotjaments turístics registrats a l'illa de Mallorca.",
    "spatial": "Mallorca",
    "creator": "A04003894",
    "dateDownload": "2025-07-02T16:00:00+02:00"
  },
}
```

Figura 4: Metadatos del dataset.

3.2.2. Campos de los registros de datos

Los registros de datos que devuelva la consulta estarán compuestos por un conjunto de campos que deberán cumplir las siguientes especificaciones:

- 1) Los nombres de los campos deben escribirse en catalán³ y en formato *camelCase*.
- 2) Todos los registros de datos deben tener los mismos campos, estén informados o no. La estructura de campos de todos los registros debe ser la misma.
- 3) Los nombres de los campos no podrán incluir ninguno de los caracteres especiales de las URL: ! * ' () ; : @ & = + \$, / ? % # [] .
- 4) Los campos de tipo fecha deberán definirse con el formato ISO 8601. Ejemplo: 2023-07-26T12:58:55+02:00.
- 5) Los campos booleanos deben tomar los valores true o false.
- 6) Los campos de tipo float/double no deben usar separador de miles, pero sí el punto («.») como separador decimal: 1234.56.

³ Artículos 21 y 22 del Decreto 49/2018, de 21 de diciembre, sobre el uso de las lenguas oficiales en la Administración de la Comunidad Autónoma de las Illes Balears.



- 7) Siempre que sea posible, los datos deberán incluir los siguientes campos:
 - a) Identificador (por ejemplo, NIF de persona jurídica).
 - b) Coordenadas geográficas (latitud/longitud, utm(x)/utm(y)). El formato de estos campos debe ser numérico.
 - c) Campo de género (hombre/mujer/no binario).
 - d) Campo de sexo (masculino/femenino).
 - e) Fecha de alta del registro de datos.
 - f) Fecha de la última actualización de los datos.
- 8) Si los datos hacen referencia a localidad, municipio⁴, isla, provincia o comunidad autónoma, deberán devolverse con dos campos:
 - 1) El nombre oficial.
 - 2) El código según el INE (Instituto Nacional de Estadística).
- 9) Para cualquier campo de código que tenga asociado un campo de descripción, deberán devolverse ambos campos (código y descripción). El nombre de estos campos debe seguir el siguiente esquema:
 - 1) **codi**[NombreCampo]. Ejemplo, codiTipusDocument.
 - 2) **descripcio**[NombreCampo]. Ejemplo, descripcioTipusDocument.
- 10) Por defecto, los datos de carácter personal deberán ofuscarse, salvo que la finalidad para la que fueron obtenidos permita su reutilización interna. (Ejemplo: NIF de persona física). En cualquier caso, la no ofuscación de un campo con datos personales requerirá autorización previa de la DGEDDT.
- 11) La codificación de los caracteres de los campos de salida debe ser UTF-8. Se debe tener en cuenta lo indicado en el apartado 3.6 de este documento.
- 12) Se recomienda que los registros de datos devueltos tengan una estructura plana de campos, sin jerarquías. En caso de que las jerarquías faciliten la comprensión de los datos devueltos, podrán utilizarse.
- 13) Todos los campos de los registros de datos se tienen que agrupar en el campo **data**, tal y como se muestra en la siguiente figura:

⁴ Los nombres oficiales de los municipios de las Illes Balears se pueden consultar en el Catálogo de Datos Abiertos GOIB: <https://intranet.caib.es/opendatacataleg/dataset/municipis>



```
{
  "metadata": {
    "title": "Allotjaments turístics Mallorca",
    "description": "Dades dels allotjaments turístics registrats a l'illa de Mallorca.",
    "spatial": "Mallorca",
    "creator": "A04003094",
    "dateDownload": "2025-07-02T16:00:00+02:00"
  },
  "data": [
    {
      "nom": "Hotel Bellavista",
      "dataAlta": "2023-05-12",
      "adreca": "Carrer de la Mar, 15, Palma",
      "illa": "Mallorca"
    },
    {
      "nom": "Apartaments Sol i Platja",
      "dataAlta": "2022-11-03",
      "adreca": "Passeig Marítim, 45, Alcúdia",
      "illa": "Mallorca"
    }
  ]
}
```

Figura 5: Estructura de datos plana, sin jerarquías.

3.3. Paginación

- 1) En el caso de que el servicio REST devuelva una gran cantidad de datos y esto pueda comprometer el buen funcionamiento de la aplicación, se deberá paginar la salida de datos. En este caso, la respuesta deberá incluir estos 7 campos que informan sobre la paginación:
 1. **totalCount:** número total de registros de datos existentes.
 2. **itemsReturned:** número de registros de datos devueltos en una página.
 3. **pageSize:** número de registros de datos que se solicitan por página (el tamaño por defecto de una página será como mínimo de 100 registros de datos).
 4. **totalPages:** número total de páginas.
 5. **page:** número de la página actual (la primera página debe empezar en 1).
 6. **nextUrl:** URL completa para acceder a la página siguiente. Cuando se llegue a la última página, el campo nextUrl deberá ser null. La URL tiene que ser navegable, por ejemplo:
<https://intranet.caib.es/cathosapi/externa/allotjaments-turistics?page=2&pageSize=50>.



7. **previousUrl**: URL completa para acceder a la página anterior. Cuando se llegue a la primera página, el campo previousUrl deberá ser null.
- 2) Los campos **page** y **pageSize** deberán ser parámetros de entrada opcionales en la consulta. El valor del parámetro pageSize deberá mantenerse inalterado durante la navegación de la paginación. A continuación, se muestra un ejemplo de URL con estos parámetros: «<https://intranet.caib.es/cathosapi/externa/allotjaments-turisticis?page=2&pageSize=50>».
- 3) Todos los campos de paginación anteriores se tienen que agrupar dentro del campo **metadata**, tal y como se muestra en la figura siguiente:

```
{
  "metadata": {
    "title": "Allotjaments turístics Mallorca",
    "description": "Dades dels allotjaments turístics registrats a l'illa de Mallorca.",
    "spatial": "Mallorca",
    "creator": "A04003894",
    "dateDownload": "2025-07-02T16:00:00+02:00",
    "totalCount": 1250,
    "itemsReturned": 50,
    "pageSize": 50,
    "totalPages": 25,
    "page": 1,
    "nextUrl": "https://intranet.caib.es/cathosapi/externa/allotjaments-turisticis?page=2&pageSi",
    "previousUrl": null
  },
  "data": [
    {
      "nom": "Hotel Bellavista",
      "dataAlta": "2023-05-12",
      "adreca": "Carrer de la Mar, 15, Palma",
      "illa": "Mallorca"
    },
    {
      "nom": "Apartaments Sol i Platja",
      "dataAlta": "2022-11-03",
      "adreca": "Passeig Marítim, 45, Alcúdia",
      "illa": "Mallorca"
    }
  ]
}
```

Figura 6: Metadatos de paginación.



3.4. *Parámetros de entrada de los servicios REST*

Los servicios REST deben tener definidos parámetros de entrada para filtrar los datos que se consultan. Estos parámetros deben cumplir las siguientes características:

- 1) Los parámetros de entrada no deben tener valores por defecto; deben estar vacíos.
- 2) Todos los parámetros de entrada deben ser opcionales, no pueden ser obligatorios. En caso de querer definir parámetros obligatorios, será necesario justificar su obligatoriedad y obtener la autorización de la DGEDDT.
- 3) Los parámetros de entrada se deben pasar por URL, por ejemplo: ? clave1=valor1&clave2=valor2.
- 4) Los campos de los registros de datos de la respuesta del servicio REST (apartado 3.2.2) deben formar parte del conjunto de parámetros de entrada (por ejemplo, id, nombre, tipo, etc.). Los nombres de los campos y los parámetros deben coincidir.
 - a) Los campos de tipo fecha deben tener asociados dos parámetros de entrada para poder filtrar mediante un rango de fechas (fecha de inicio y fecha de fin), por ejemplo: dataPublicacioInici y dataPublicacioFi, en formato ISO8601: 2022-07-26.
 - b) Los campos de tipo numérico deben tener asociados dos parámetros de entrada para poder filtrar mediante un rango de valores (mínimo y máximo), por ejemplo: importTotalMin e importTotalMax.
 - c) Los parámetros de tipo float/double no deben usar separador de miles, pero deben usar el punto («.») como separador de decimales, por ejemplo: 1234.56.
 - d) No se pueden definir parámetros de entrada a partir de campos que contengan datos personales.
- 5) Si los datos se devuelven en más de un idioma, se debe añadir un parámetro de entrada que indique el idioma en el que se devuelven los datos. El valor de este parámetro debe seguir el formato ISO 639-1, por



ejemplo: idioma=ca. Por defecto, si no se indica ningún valor para este parámetro, los datos se deben devolver en catalán

- 6) Si la respuesta está paginada, los campos pageSize y page se deben definir como parámetros de entrada. En el servidor se deben controlar sus valores y, si son nulos o no válidos, se deben aplicar valores por defecto.
- 7) En el Swagger UI, los parámetros de entrada deben describirse en catalán⁵ de forma completa y precisa. Si tienen valores por defecto o un rango concreto de valores, se deben indicar. Por ejemplo:

Indica l'idioma de la resposta: 'ca' (català) per defecte, o 'es' (espanyol).

Figura 7: Ejemplo de descripción completa y precisa.

3.5. Validación de la API y servicios REST

- 1) Tanto la API como los servicios REST deben cumplir todas las especificaciones definidas en este documento para ser aprobados por el Departamento de Desarrollo Tecnológico del Gobierno de las Illes Balears.
- 2) Si la salida del servicio REST está paginada, se debe poder validar la paginación y la navegación entre páginas. El conjunto de pruebas debe ser suficiente para paginar al menos 3 páginas. Se recuerda que el número mínimo de registros de datos por página es de 100.
- 3) Se deben poder validar todos los metadatos de la salida del servicio REST. Todos los metadatos deben estar informados con datos reales.
- 4) Se deben poder validar todos los campos de los registros de datos. Todos los campos de los registros de datos deben estar informados de forma adecuada.

3.6. Guía y ejemplos de implementación

- 1) Se recomienda que cada aplicación disponga de un cliente Java que pueda consumir los servicios REST.

⁵ Artículos 21 y 22 del Decreto 49/2018, de 21 de diciembre, sobre el uso de las lenguas oficiales en la Administración de la Comunidad Autónoma de las Illes Balears.



- 2) Se debe utilizar la codificación UTF-8 para los caracteres de salida de los servicios REST. Para implementar esto, se deberá tener en cuenta lo siguiente:
 - a) Los archivos .java deben utilizar el conjunto de caracteres UTF-8.
 - b) Se debe añadir la siguiente etiqueta en todos los archivos XML:
<?xml version="1.0" encoding="UTF-8"?>
 - c) Se debe añadir la siguiente propiedad en el archivo pom.xml del proyecto Maven padre:
<project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
 - d) En el caso de las API REST, la codificación debe añadirse en la clase que genera la salida del servicio REST, siguiendo el siguiente ejemplo:
String value = new String(myString.getBytes("UTF-8"), "UTF-8");
- 3) Además, es recomendable revisar que:
 - a) La JVM que utiliza el JBoss tenga configurado el siguiente parámetro:
-Dfile.encoding=UTF-8
 - b) También se recomienda que las variables de entorno del servidor que controlan la configuración regional e idioma tengan configurado el charset como UTF-8.
- 4) Si los servicios REST de reutilización y apertura de datos definidos en la API interna están securizados mediante mecanismos de autenticación y control de acceso, se deberán tener en cuenta ciertos aspectos:
 - a) Es necesario crear un usuario de aplicación para poder ejecutarlos. El nombre y los roles asignados a este usuario dependerán del entorno en el que se cree. A continuación, se muestra una tabla con los nombres y roles que deberían tener según el entorno.



Entorno	Nombre usuario	Roles
Preproducción	\$opendata_<código-aplicación>_pre	<PREFIJO-APLICACIÓN>_API_EXTERNA y/o <PREFIJO-APLICACIÓN>_API_INTERNA
Producción	\$opendata_<código-aplicación>	<PREFIJO-APLICACIÓN>_API_EXTERNA y/o <PREFIJO-APLICACIÓN>_API_INTERNA

- b) Si un usuario sin permisos intenta ejecutar un servicio REST autenticado, se tendrá que mostrar un aviso de este tipo:

```
Error: Unauthorized
Response body
{
  "errorCode": 401,
  "errorMessage": "No autenticat. Has d'iniciar sessió.",
}
```

Figura 8: Ejemplo de error en el acceso y ejecución de un servicio REST.