BORRADOR - GUÍA TÉCNICA DE INTEROPERABILIDAD EN EL ESTADO DE CHILE

La presente guía complementa y especifica la Norma Técnica de Interoperabilidad, en relación a los estándares y procedimientos asociados a la citada norma. Se espera pueda ser actualizada con la periodicidad requerida para mantener vigentes los protocolos y estándares asociados al intercambio de información, así como también a los procedimientos definidos en la Norma, que permiten solicitar la inclusión de nuevos estándares.

TABLA DE CONTENIDOS

1. Definiciones técnicas

- **Codificación en Base64**: Sistema de codificación que usa 64 caracteres como base y que tiene la capacidad de representar cualquier dato en un formato que permite facilitar el transporte de los mismos a través de las redes, reduciendo la posibilidad de ser

modificados en tránsito por error. Estos datos pueden ser archivos de texto plano, ejecutables, imágenes, binarios, etc.

- **Dublin Core**: Dublin Core es un modelo de metadatos elaborado y auspiciado por la Dublin Core Metadata Initiative (DCMI). Es un sistema de definiciones semánticas que tiene como objeto describir el contenido, la propiedad intelectual y la creación e identidad de los datos.
- **Especificación OpenAPI (sigla en inglés: OAS):** Define un estándar para la descripción de una interfaz de programación con lenguaje agnóstico para una API REST, el permite a humanos y computadoras, entender y descubrir las capacidades de un servicio de interoperabilidad.
- **HTTP**: Protocolo de Transferencia de Hipertexto. Es un protocolo sin estado de la capa de aplicación diseñado para la transmisión de documentos hipermedia, como HTML. Basado en una conexión del tipo TCP/IP, puede usarse con cualquier otro protocolo del nivel de la capa de transporte orientado a la conexión.
- **Identificador de Objetos (sigla en inglés: OID):** Permite identificar objetos o recursos, y se define de acuerdo a una asignación jerárquica, establecida por la Organización Internacional de Normalización (ISO). Está basado en la Norma ISO/IEC 8824-1.
- **Interfaz de programación de aplicaciones (sigla en inglés: API):** Conjunto de procesos, funciones y métodos que brinda una determinada biblioteca de software a modo de capa de abstracción para que sea empleada por otro software.
- **Métodos de petición HTTP**: Son los que indican la acción que se desea realizar para un recurso determinado; GET, HEAD, POST, PUT, DELETE, CONNECT, OPTIONS, TRACE y PATCH.
- **OpenAPI**: Iniciativa encargada de la estandarización de cómo se describe un API REST.
- **Transferencia de estado representacional (sigla en inglés: REST)**: Arquitectura de software basada en el uso de peticiones HTTP para el diseño de servicios web.

2. Estándares

Los servicios de interoperabilidad serán implementados a través de APIs, por tanto, en el resto de esta guía técnica, cada vez que se mencione una API, se entenderá como un servicio de interoperabilidad.

a. Tipo de arquitectura:

Los servicios de interoperabilidad se implementan según el estilo de arquitectura REST:

- Todas las operaciones de las APIs deben utilizar el protocolo HTTP[1], ser procesadas en tiempo real y se debe entregar una respuesta según la codificación establecida en el protocolo HTTP.
- Los servicios de interoperabilidad deben implementar las operaciones utilizando los siguientes métodos HTTP:
    - GET: Solicitar un recurso.
    - POST: Crear un nuevo recurso subordinado dentro de una URL existente.
    - DELETE: Eliminar un recurso.
    - PUT: Crear un nuevo recurso a una nueva URL o modificar un recurso existente en una URL.
    - HEAD: Idéntica a GET excepto que no se retorna un cuerpo del mensaje en la respuesta.
    - CONNECT: establece un túnel hacia el servidor identificado por el recurso.
    - OPTIONS: utilizado para describir las opciones de comunicación para el recurso de destino.
    - TRACE: realiza una prueba de bucle de retorno de mensaje a lo largo de la ruta al recurso de destino.
    - PATCH: es utilizado para aplicar modificaciones parciales a un recurso.

  Es importante identificar que de los métodos precedentes, los cuatro primeros son los más utilizados, en particular para atender las operaciones CRUD (*create, read, update* y *delete*).

b. Formato para intercambio de datos:

Los mensajes que se intercambien con una API deben utilizar JSON[2] como formato.
En caso de que también se requiera utilizar XML en el intercambio, este formato podrá ser utilizado, pero adicional al formato JSON.
En caso de que una API soporte más de un formato, el modo para especificar el formato solicitado será a través del encabezado (ver sección g. Encabezados de los mensajes), utilizando la cabecera *Accept*. Si no se especifica ningún formato, se responderá con JSON. A continuación se muestra un ejemplo:

```
"Accept": "application/xml" responde en formato XML
"Accept": "application/json" responde en formato JSON. No es necesario especificarlo, ya que es el formato por defecto.
```

En caso de que no se especifique un formato en particular, siempre se utilizará JSON como formato por omisión para la respuesta.

[1] RFC 4918
[2] RFC 7159

Los documentos digitales que sean transferidos entre las Instituciones y que se encuentren en un formato de representación autocontenido (Ejemplo: PDF, JPG, DOCX), deben ser codificados en formato base64 y enviarlos con los metadatos asociados según el estándar Dublin Core (ISO 15836). En el Catálogo de esquemas y metadatos se encuentra el esquema denominado “Recurso Dublin Core” el cual contiene los campos requeridos para el envío de los metadatos junto al base64.

c. Seguridad en la conexión:

Los servicios de interoperabilidad deberán implementar como mecanismo de seguridad el protocolo de autorización OAuth 2[3]. Este protocolo permite flujos simples de autorización de acceso para poder utilizar los servicios de interoperabilidad.

I. El mecanismo de seguridad está compuesto por:

Datos para el proveedor del servicio de interoperabilidad:
Provider ID: URL del servicio de interoperabilidad.
Provider secret: Cadena de texto opaco.

Datos para el consumidor del servicio de interoperabilidad:
Client ID: URL del servicio de interoperabilidad.
Client secret: Cadena de texto opaco.
Json Web token (JWT): Es un token de acceso es una cadena opaca que permite la identificación del consumidor de un servicio de interoperabilidad y permite que éste pueda utilizarlo. El token está compuesto por su valor y tiempo de vigencia.

II. Flujo para consultar un servicio de interoperabilidad:

Se debe utilizar el flujo “Client Credentials Grant” de OAuth 2 el cual implica las siguientes acciones:

1. En el momento en que se habilita un servicio de interoperabilidad en el Catálogo de Servicios, se generará un **Provider ID** y un **Provider secret**.
2. Al firmar un convenio de interoperabilidad en la Plataforma, el consumidor del servicio podrá visualizar el **Client ID** y el **Client Secret**.

[3] RFC 6749

3. El consumidor del servicio de interoperabilidad debe solicitar un **JWT** a la URL definida dentro de la descripción del servicio utilizando el **Client ID** y **Client Secret**.
   La respuesta será un objeto JSON:

```
{
        "access_token":"<JWT, validado con el provider secret>",
        "token_type":"bearer",
        "expires_in":<tiempo de duración, en segundos>,
}
```

4. Con el **JWT**, el consumidor realiza la petición a los servicios protegidos usando un header HTTP:

```
        "Authorization": "Bearer <JWT, validado con el provider secret>"
```

5. El proveedor del servicio verifica el **JWT**: que el secreto compartido sea correcto, aun no expirado, que sea para el servicio en cuestión (campo aud del **JWT** debe coincidir con el **Provider Id**), opcionalmente puede verificar la institución consumidora (campo sub del **JWT**).

El tiempo de vigencia de los tokens debe ser máximo de 5 días, tiempo en el cual el consumidor del servicio debe solicitar un nuevo token para continuar realizando consultas al servicio. Se sugiere realizar la renovación del token una vez al día para mitigar el riesgo de un problema en la asignación del nuevo token.
Para más información respecto de cómo se utiliza la autenticación utilizando el protocolo OAuth 2, se debe revisar el punto referido a “Client Credentials Grant” del RFC 6749 (https://tools.ietf.org/html/rfc6749#section-4.4)

d. Seguridad del endpoint en el proveedor:

Se debe utilizar el protocolo criptográfico TLS[4] versión 1.2 para brindar seguridad en la capa de red entre las conexiones de las instituciones. Se debe deshabilitar los protocolos inseguros, tales como SSLv2 y SSLv3.

La implementación en el ambiente de producción del proveedor del servicio de interoperabilidad, se debe realizar por medio de certificados de sitio seguro (SSL/TLS) emitidos por una Autoridad Certificadora de confianza pública, con el algoritmo de firma SHA-2 (SHA-256) y largo de llaves equivalente en seguridad a RSA 2048-bit o superior. No se debe habilitar la comunicación vía HTTP sin cifrar.

[4] RFC 6176 y RFC 7568

e. Especificación de servicios, esquemas y metadatos:

I. Definición de un servicio de interoperabilidad: El Catálogo de Servicios publicado en la Plataforma, utiliza la especificación OAS para definir cada uno de los servicios de interoperabilidad. Por tanto, al registrar un nuevo servicio, se solicitarán datos que describan la funcionalidad de cada una de las posibles operaciones del servicio.
Este es un ejemplo de la definición de un servicio.

II. Datos entregados por un servicio de interoperabilidad: Se definen dos tipos de esquemas:

1. Esquemas basales: Esquemas de bajo nivel utilizados para regular las nomenclaturas y estructuras básicas de información.
A modo de ejemplo, aquí se puede ver el esquema de Fecha y Hora.

2. Esquemas documentales: Esquemas de documentos que son de uso común para las instituciones del Estado, como por ejemplo Oficios y Resoluciones. Los esquemas documentales pueden contener esquemas basales de ser requerido.

f. Codificación de los datos:

La codificación de los mensajes y los documentos con representación autocontenida (ver punto 2.a) que se intercambien a través de los servicios de interoperabilidad será UTF-8[5].

g. Fechas y marcas de tiempo:

Las fechas y marcas de tiempo deben ser expresadas utilizando el estándar de tiempo ISO-8601 y la zona horaria UTC.

h. Encabezados de los mensajes:

Los consumidores de los servicios deberán incluir en el encabezado de las solicitudes las siguientes cabeceras:

```
Authorization: Bearer <JWT, validado con el provider secret>
Requester-Organization-Id: <Código de la Institución>
```

[5] RFC 3629

```
Transaction-Id: <OID>
Accept: <Application/JSON, Application/XML> // Opcional
```

Los proveedores de los servicios deberán incluir en el encabezado de las solicitudes las siguientes cabeceras:

```
Provider-Id: AF001: <Código de la Institución>
Transaction-Id: <OID>
```

3. Construcción de servicios

a. Definición de un servicio en el catálogo

La descripción de servicios será realizada utilizando el estándar OpenAPI Specification (OAS) v 2.0.

Los usuarios autorizados de cada institución deben ingresar a la Plataforma de Interoperabilidad y autenticarse por medio de claveúnica. Allí podrán realizar las siguientes acciones:

I. **Registrar un nuevo servicio de interoperabilidad:** Se debe subir a la plataforma un archivo OAS con los datos para describir el servicio y sus operaciones. La definición del servicio debe utilizar los esquemas basales o documentales en los casos en que se entregue información que esté definida en el catálogo de esquemas y metadatos. La solicitud del nuevo servicio será revisado por SEGPRES para su posterior publicación.

II. **Actualizar un servicio publicado**: En la Plataforma dentro de la descripción del servicio que se modificará, se debe generar una solicitud para una nueva versión, para lo cual se debe subir un archivo OAS bajo los mismo lineamientos del punto anterior. En los casos en que la nueva versión sea validada por SEGPRES, se habilitará la nueva versión en el catálogo de servicios. El servicio con la versión anterior deberá mantenerse operativo por 6 meses mientras que las instituciones que consuman el servicio realicen la migración a la nueva versión.

III. **Eliminar un servicio publicado**: En la Plataforma dentro de la descripción del servicio que se eliminará, se debe generar una solicitud informando el motivo de la eliminación. En los casos en que la eliminación sea validada por SEGPRES, se marcará el servicio como “En proceso de eliminación” en el Catálogo de Servicios. El servicio deberá mantenerse operativo por 6 meses. Después del mes 6, el servicio no estará disponible en el Catálogo de Servicios.

b. Definición de URI de recursos

Seguir las siguientes prácticas (expresadas con ejemplos):[6]

I. Usar sustantivos para describir los recursos:

GET /usuarios/ lista todos los usuarios
GET /usuarios/12 muestra detalle del usuario con id 12
POST /usuarios crea usuario
PUT /usuarios/12 actualiza usuario con id 12
PATCH /usuarios/12 actualiza parcialmente usuario con id 12
DELETE /usuarios/12 borra usuario con id 12

II. Usar "/" para indicar la relación de jerarquía:

GET /usuarios/12/mensajes muestra mensajes del usuario con id 12
GET /usuarios/12/mensajes/93 obtiene mensaje con id 93 del usuario con id 12
POST /usuarios/12/mensajes crea mensaje para usuario con id 12
DELETE /usuarios/12/mensajes/3 elimina mensaje 3 de usuario con id 12

III. Uso de parámetros de consulta: Para realizar operaciones de filtro, ordenamiento, paginación o búsqueda se pueden pasar parámetros de consulta.

GET /instituciones/buscar?nombre=ministerio* busca organizaciones cuyo nombre comienza con "ministerio"
GET /instituciones?ordenar=fecha_creacion,desc lista y ordena organizaciones por fecha de creación, descendente
GET /instituciones/2/usuarios?estado=inactivo lista todos los usuarios de la institución 2 con estado inactivo

IV. Formato de las URIs
   A. No usar "/" al final de la URI.
   B. No usar mayúsculas.

~~**/Instituciones/2/Usuarios/**~~

   C. Usar guión "-" (y no guión bajo "_") para facilitar la comprensión de la URI en los casos en que sea más de una palabra para el servicio.

/oficinas/134/como-llegar

[6] Fuente: http://restfulapi.net/resource-naming/

~~/oficinas/134/como_llegar~~

D. Usar sólo letras en minúscula dentro de la URI. No utilizar caracteres especiales ni acentos.
E. Versionamiento de los servicios de interoperabilidad. Cada servicio de interoperabilidad contará con un número de versión en la URI, el cual deberá ser incrementado en la medida en que el servicio se vea modificado en su naturaleza o los datos que entrega (ver artículo 11 de la norma técnica).

```
URI Versión 1 (antigua):
      GET /v1/oficinas/134/como-llegar
URI Version 2 actual:
      GET /v2/oficinas/134/como-llegar
```

V. No usar extensiones de archivos.

~~/instituciones/2/descripcion.txt~~

4. Procesos y procedimientos para interoperar

a. Proveedor.

Las instituciones proveedoras de información, posterior a la publicación del servicio definido en el punto 3, pondrán a disposición el servicio de interoperabilidad en ambiente de producción, el que será consumido de forma oficial y para realizar operaciones reales, luego de firmar el respectivo convenio entre las instituciones.

La institución proveedora puede utilizar el código fuente que automáticamente genera la Plataforma a partir de la definición del servicio en lenguajes PHP, Java y C#.

Se debe considerar dentro de la construcción del servicio:

- Seguridad del endpoint utilizando certificado SSL.
- Trazabilidad.
- Monitoreo.

b. Consumidor

En la Plataforma se encontrará la información de cada uno de los servicios de interoperabilidad publicados en el catálogo de servicios.

La institución consumidora puede utilizar el código fuente que automáticamente genera la Plataforma a partir de la definición del servicio en lenguajes PHP, Java y C#.

En la Plataforma se podrán hacer pruebas al servicio de interoperabilidad sin requerir la firma del convenio.

La Plataforma entregará a la parte consumidora las credenciales para hacer las consultas en el ambiente de producción, una vez que se haya firmado el convenio con la parte proveedora, lo cual se realizará dentro de la misma Plataforma.

5. Trazabilidad

Cada institución, tanto proveedora como consumidora, debe implementar una operación para que la Plataforma pueda obtener y almacenar los archivos de registro de cada servicio de interoperabilidad que ofrezca o consuma. La obtención por parte de la Plataforma se realizará diariamente, obteniendo todos los archivos de registro que se generen durante un día.

Cada registro equivale a una transacción y debe tener la siguiente información:

- Fecha y Hora según el estándar definido en el punto 2.
- IP del Host remoto
- Servicio de interoperabilidad consultado.
- Método HTTP y URI solicitada.
- Código HTTP de respuesta según lo establecido en el RFC 2616, sección 10:
    - **2XX** es exitoso
    - **3XX** es una redirección
    - **4XX** es un error del cliente
    - **5XX** es un error del servidor
- Código de la Institución consumidora de la información según el listado expuesto en el siguiente recurso: https://apis.digital.gob.cl/misc/instituciones/_search?size=900&pretty=true.
- Identificador de la transacción en formato OID. Para generar el identificador de cada transacción se debe consultar la Guía de Manejo de OIDs.

Para efectos de disponer del código de institución, tanto en el registro de peticiones como en el de respuestas, se debe añadir el código de la institución en el header de la petición o respuesta según corresponda.

    - Para el caso de una petición de un servicio

```
Requester-Organization-Id: BB001
```

- Para el caso de la respuesta a una petición:

```
Provider-Id: AF001
```

Dado todo lo anterior, el contenido de un archivo de registro debe tener el detalle por cada una de las transacciones, una transacción por línea, en el siguiente formato:

```
[<fecha y hora>] <host remoto> <servicio consultado> "<metodo HTTP y URI solicitada>"
<codigo respuesta HTTP> "<codigo institucion consumidora> <ID transaccion>"

Ejemplo:
[2016-10-25T08:15:30-04:00] 20.30.40.50 apis.digital.gob.cl "GET
/misc/instituciones/AD015" 200 "BB001 e0592e56-76c3-11e7-814c-0401beb96201"
```

La rotación de archivos de registro debe realizarse por fecha y tamaño, en los siguientes casos:

- Han transcurrido 24 horas desde su creación.
- Su tamaño supere los 10MB

En el nombre del archivo de registro se indicará la fecha de creación de éste, utilizando el formato de fecha definido en el punto 2:

`instituciones-2018-11-02T00:34:02,781.log`

En cuanto al registro histórico, la Plataforma registrará por un plazo de 5 años todos los registros de trazabilidad obtenidos. No obstante, cada institución deberá mantener sus propios archivos por a lo menos por 30 días.

En cuanto al servicio que se deberá poner a disposición para ser consumido por la Plataforma, este deberá ser parte del mismo servicio de interoperabilidad ofrecido. La respuesta de este servicio será un único archivo que contenga todos los archivos de log creados en el periodo solicitado. El empaquetado y compresión se debe realizar en formato .tar.gz, es decir combinando los formatos tar y gzip.

Ejemplo de petición por omisión que entrega logs de las últimas 24 horas comprimidos en un único archivo:

+ **Encabezado de Respuesta (*Response Headers*)**

```
Request URL: https://api.ejemplo.com/instituciones/logs
Request Method: GET
Status Code: 200 OK
Content-Length: 5242880
Content-Type: application/gzip
...
```

Ejemplo de petición por omisión que entrega logs comprimidos en un rango de tiempo, indicado por los parámetros “inicio” y “fin” y comprimidos en un único archivo:

+ **Encabezado de Respuesta (*Response Headers*)**

```
Request URL:
https://api.ejemplo.com/instituciones/logs?inicio=2017-01-10&fin=2017-01-12
Request Method: GET
Content-Length: 8942880
Content-Type: application/gzip
...
```

6. Monitoreo

Se debe generar una operación dentro del servicio a publicar que permita consultar el estado del servicio. La operación debe ser a través del método HTTP GET y el recurso a consultar será llamado “*monitor*”. Si el servicio está operando correctamente, la respuesta debe ser el código de estado HTTP “200”, para el resto de los casos, la respuesta será catalogada de acuerdo al código de estado HTTP que se responda de acuerdo al RFC 2616, punto *“10 Status Code Definitions”* (https://tools.ietf.org/html/rfc2616).

Ejemplo de respuesta para el recurso “dispositivo”:

+ **Encabezados de Respuesta (*Response Headers*)**

```
Request URL: https://api.ejemplo.com/dispositivo/monitor
Request Method: GET
Status Code: 503 Service Unavailable
```

+ **Cuerpo de Respuesta (*Response Body*)**

```
{
       “codigo_estado”: 503
       “msj_estado”: “Service Unavailable”
       “desc_personalizada_estado”: “Servicio en mantenimiento”
}
```

La Plataforma realizará un monitoreo cada minuto de todos los servicios de interoperabilidad publicados y enviará notificaciones a la Institución proveedora y consumidoras, en los casos en que se detecte que el servicio no es capaz de dar respuesta a la solicitudes durante un periodo superior a los 5 minutos. Las peticiones al servicio de monitoreo también serán incluidas en el registro de trazabilidad.

7. Uso del idioma

El uso del idioma a nivel de programación y de definición de esquemas para la construcción del servicio interoperable, tal como se puede ver en los ejemplos usados a través de esta guía, será de la siguiente forma:

- **Español**
    - Variables y contenido
    - Construcción de las URIs
    - Documentación
    - Metadatos
- **Inglés**
    - Operaciones
    - Encabezados