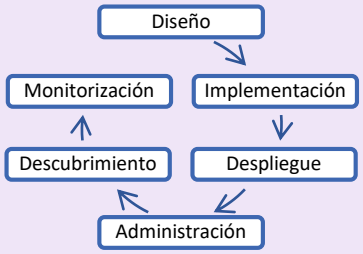
datos.gob.es
reutiliza la información pública

APIs
para Datos Abiertos

Conceptos generales

- Las APIs proporcionan una forma de acceso a los Datos Abiertos **compatible con la descarga** de archivos. Permiten **automatizar** el consumo de datos, aportan una mayor **flexibilidad** admitiendo funcionalidades de filtrado, ordenación y paginación según especificaciones de usuario y son el medio idóneo para la publicación de datos **dinámicos** y en **tiempo real**.
- Un modelo adecuado para implementar APIs sobre la Web es **API REST sobre HTTP** (*RESTful APIs*).
- En una API, la interacción entre servidor y clientes se realiza mediante el intercambio de **peticiones** y **respuestas** sobre el protocolo HTTP en un contexto seguro. Las peticiones usan **métodos** que son contestados con **códigos de respuesta** estándar.

Planifica el ciclo de vida API



Métodos habituales de petición

GET	**Recupera una representación de un recurso de datos.**
HEAD	**Recupera la cabecera de una respuesta.**
POST	**Crea un nuevo recurso de datos.**
PUT	**Actualiza un recurso existente o lo crea si no existe previamente.**
PATCH	**Realiza una actualización parcial de un recurso.**
DELETE	**Elimina un recurso de datos existente.**
OPTIONS	**Recupera opciones de comunicación para el recurso solicitado.**

El método **GET** es el más habitual para el acceso y descarga de Datos Abiertos.

Códigos de respuesta, categoría e interpretación

1xx	**Informativo**	Solicitud recibida, el proceso de respuesta está en marcha.
2xx	**Éxito**	La petición del cliente se ha recibido, entendido y aceptado con éxito.
3xx	**Redirección**	Se deben tomar medidas adicionales para completar la solicitud .
4xx	**Error** del cliente	La solicitud contiene una sintaxis incorrecta o no se puede cumplir.
5xx	**Error** del servidor	El servidor no pudo resolver una solicitud aparentemente válida.

Es recomendable definir interpretaciones comprensibles de códigos de estado, incluyendo mensajes de error legibles por las personas, informativos y útiles, así como enlaces a detalles complementarios.

Usa estándares abiertos

OpenAPI (OAS) es una especificación estándar independiente del lenguaje de programación recomendable para implementar API REST

Usa cabeceras HTTP para negociar el contenido

Ajusta cabeceras de petición y respuesta de servidor y cliente para especificar **opciones de representación y transmisión** de recursos.

Usa URIs para identificar recursos

https://datos.ejemplo.com/v1/licitaciones/contratos?estado=finalizado

esquema | autoridad | versión | contexto | recurso | consulta

Incorpora documentación completa

Es recomendable utilizar **herramientas de edición** para el siguiente contenido mínimo requerido:

- Método de **autenticación** requerido
- **Listado completo de peticiones** que la API puede manejar, incluyendo propósito, parámetros y salida.
- **Ejemplos de uso** de cada una de las peticiones posibles en diferentes lenguajes
- Relación de **versiones** de la API y características de cada una.
- Información de contacto y **feedback** con los promotores de la API.
- Mecanismo para **testar peticiones** y comprobar la respuesta de la API.

Aplica una estrategia de nombrado de recursos

- Distingue **Documentos, Colecciones, Almacenes y controladores**.
- Usa **convenciones** de lenguaje del tipo:
  - Usa términos sencillos, intuitivos y coherentes.
  - Los términos deben ser suficientemente auto-explicativos.
  - Evita la ambigüedad para la denominación de recursos en las URIs.
  - Usa términos que no requieran un conocimiento especifico del contexto
  - Evita nombres que puedan entrar en conflicto con palabras clave.
  - Sé coherente con el uso del plural o singular: el uso del plural es habitual
- Usa '/' para indicar relaciones de jerarquía
- No incluir el separador '/' como carácter final de la URI
- Usa el carácter "-" para mejorar la legibilidad de la URI
- No usar el carácter "_"
- Usa minúsculas para expresar los términos de la URI
- No incluir extensiones de archivo
- No usar nombres de funciones CRUD en las URIs

Aplica directrices de Seguridad

Es recomendable prevenir riesgos de seguridad y mitigar vulnerabilidades aplicando directrices API Security-OWASP

- Recomendaciones sobre el **cifrado de los intercambios de información:**
  - Publica servicios usando solo HTTPS (HTTP sobre TLS).
  - Usa las últimas versiones estables de librerías de componentes software
  - Redirige automáticamente de la versión HTTP de un servicio a la versión HTTPS.
  - Habilita una política de seguridad HTTP Strict Transport Security (HSTS)
  - Adquiere los certificados de servidor en autoridades de certificación confiables.
  - Aplica prevención sobre vulnerabilidades del tipo JSON injection usando sanitizer.

- **Autenticación** de usuarios. El objetivo debe ser mantener un **equilibrio adecuado entre seguridad y maximización del uso** de la API. Según el nivel de seguridad requerido, son enfoques aplicables:
  - Sin autenticación: solo el uso del método GET para el acceso a los datos.
  - Autenticación por medio de claves API (API Key)
  - Autenticación utilizando OAuth (Open Authorization)

- Recomendaciones sobre limitación de uso para **preservar la integridad y saturación** de la API:
  - Ajusta **la tasa de peticiones y cuota de uso** por clave
  - Optimiza el rendimiento **estructurando grandes volúmenes** de datos

Es recomendable **desacoplar accesos a datos públicos y privados** para evitar el impacto del consumo externo sobre operaciones internas en los mismos datos.

Evita la degradación del rendimiento del servidor

- Ajusta la **latencia** de la API según el tipo y frecuencia del dato que se este entregando.
- Usa técnicas de **caching** con recursos que se consumen habitualmente y no cambian con mucha frecuencia.

Guía práctica para la publicación de Datos Abiertos usando APIs

Iniciativa aporta
Gobierno de España · Vicepresidencia Tercera del Gobierno · Ministerio de Asuntos Económicos y Transformación Digital
red.es